
Amazon Elastic Compute Cloud

User Guide for Linux Instances



Amazon Elastic Compute Cloud: User Guide for Linux Instances

Copyright © 2016 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon EC2?	1
Features of Amazon EC2	1
How to Get Started with Amazon EC2	2
Related Services	2
Accessing Amazon EC2	3
Pricing for Amazon EC2	3
PCI DSS Compliance	4
Instances and AMIs	4
Instances	5
AMIs	6
Regions and Availability Zones	7
Region and Availability Zone Concepts	7
Available Regions	8
Regions and Endpoints	9
Describing Your Regions and Availability Zones	9
Specifying the Region for a Resource	11
Launching Instances in an Availability Zone	12
Migrating an Instance to Another Availability Zone	13
Root Device Volume	14
Root Device Storage Concepts	14
Choosing an AMI by Root Device Type	16
Determining the Root Device Type of Your Instance	17
Changing the Root Device Volume to Persist	17
Setting Up	19
Sign Up for AWS	19
Create an IAM User	19
Create a Key Pair	21
Create a Virtual Private Cloud (VPC)	23
Create a Security Group	23
Getting Started	26
Overview	26
Prerequisites	27
Step 1: Launch an Instance	27
Step 2: Connect to Your Instance	28
Step 3: Clean Up Your Instance	29
Next Steps	30
Best Practices	31
Tutorials	33
Tutorial: Installing a LAMP Web Server on Amazon Linux	33
Related Topics	42
Tutorial: Hosting a WordPress Blog	42
Prerequisites	43
Install WordPress	43
Next Steps	50
Help! My Public DNS Name Changed and now my Blog is Broken	50
Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS	51
Prerequisites	52
Step 1: Enable SSL/TLS on the Server	52
Step 2: Obtain a CA-signed Certificate	54
Step 3: Test and Harden the Security Configuration	57
Troubleshooting	60
Tutorial: Increase the Availability of Your Application	60
Prerequisites	61
Scale and Load Balance Your Application	61
Test Your Load Balancer	63

Tutorial: Remotely Manage Your Instances	64
Launch a New Instance	64
Grant Your User Account Access to SSM	65
Install the SSM Agent	65
Send a Command Using the EC2 Console	67
Send a Command Using the AWS CLI	68
Amazon Machine Images	69
Using an AMI	69
Creating Your Own AMI	70
Buying, Sharing, and Selling AMIs	70
Deregistering Your AMI	70
Amazon Linux	70
AMI Types	71
Launch Permissions	71
Storage for the Root Device	71
Virtualization Types	74
Finding a Linux AMI	75
Finding a Linux AMI Using the Amazon EC2 Console	75
Finding an AMI Using the AWS CLI	76
Shared AMIs	76
Finding Shared AMIs	77
Making an AMI Public	78
Sharing an AMI with Specific AWS Accounts	80
Using Bookmarks	81
Guidelines for Shared Linux AMIs	81
Paid AMIs	85
Selling Your AMI	86
Finding a Paid AMI	86
Purchase a Paid AMI	87
Getting the Product Code for Your Instance	87
Using Paid Support	88
Bills for Paid and Supported AMIs	88
Managing Your AWS Marketplace Subscriptions	88
Creating an Amazon EBS-Backed Linux AMI	89
Overview of Creating Amazon EBS-Backed AMIs	89
Creating a Linux AMI from an Instance	90
Creating a Linux AMI from a Snapshot	91
Creating an Instance Store-Backed Linux AMI	92
Overview of the Creation Process for Instance Store-Backed AMIs	93
Prerequisites	93
Setting Up the AMI Tools	94
Creating an AMI from an Instance Store-Backed Instance	120
Converting to an Amazon EBS-Backed AMI	129
AMIs with Encrypted Snapshots	132
AMI Scenarios Involving Encrypted EBS Snapshots	133
Copying an AMI	135
Copying an AMI You Own	136
Copying an AMI Across AWS Accounts	136
Copying an AMI Across Regions	136
Copying to Encrypt	137
AMI Copying Scenarios	138
Copying an AMI Using the Console or Command Line	139
Stopping a Pending AMI Copy Operation	140
Deregistering Your AMI	140
Cleaning Up Your Amazon EBS-Backed AMI	141
Cleaning Up Your Instance Store-Backed AMI	142
Amazon Linux	142
Finding the Amazon Linux AMI	143

Launching and Connecting to an Amazon Linux Instance	143
Identifying Amazon Linux AMI Images	143
Included AWS Command Line Tools	144
cloud-init	145
Repository Configuration	146
Adding Packages	147
Accessing Source Packages for Reference	147
Developing Applications	148
Instance Store Access	148
Product Life Cycle	148
Security Updates	148
Support	149
User Provided Kernels	149
HVM AMIs (GRUB)	149
Paravirtual AMIs (PV-GRUB)	150
Instances	156
Instance Types	156
Available Instance Types	157
Hardware Specifications	158
Virtualization Types	158
Networking and Storage Features	158
Instance Limits	159
T2 Instances	160
C4 Instances	163
GPU Instances	166
I2 Instances	169
D2 Instances	170
H1 Instances	173
HS1 Instances	174
T1 Micro Instances	175
X1 Instances	187
Resizing Instances	189
Instance Purchasing Options	193
Determining the Instance Lifecycle	193
Reserved Instances	194
Scheduled Instances	215
Spot Instances	219
Dedicated Hosts	264
Instance Lifecycle	275
Instance Launch	275
Instance Stop and Start (Amazon EBS-backed instances only)	276
Instance Reboot	276
Instance Retirement	276
Instance Termination	276
Differences Between Reboot, Stop, and Terminate	277
Launch	278
Connect	289
Stop and Start	299
Reboot	302
Retire	303
Terminate	305
Recover	310
Configure Instances	311
Common Configuration Scenarios	311
Managing Software	312
Managing Users	320
Processor State Control	322
Setting the Time	326

Changing the Hostname	330
Setting Up Dynamic DNS	332
Running Commands at Launch	334
Instance Metadata and User Data	337
Remotely Manage Your Instances	352
Components and Concepts	352
Prerequisites	355
Installing the SSM Agent	357
Delegating Access	365
Setting Up Run Command On Managed Instances	371
Executing Commands	374
Viewing Command Output	377
Creating SSM Documents	378
Sharing SSM Documents	381
Walkthroughs	387
Cancelling a Command	394
Monitoring Commands	394
Troubleshooting Run Command	404
Importing and Exporting Virtual Machines	406
Monitoring	407
Automated and Manual Monitoring	409
Automated Monitoring Tools	409
Manual Monitoring Tools	410
Best Practices for Monitoring	410
Monitoring the Status of Your Instances	411
Instance Status Checks	411
Scheduled Events	415
Monitoring Your Instances Using CloudWatch	419
Enable Detailed Monitoring	419
List Available Metrics	421
Get Statistics for Metrics	426
Graph Metrics	441
Create a CloudWatch Alarm	444
Create Alarms That Stop, Terminate, Reboot, or Recover an Instance	450
Monitoring Memory and Disk Metrics	471
Prerequisites	472
Getting Started	474
mon-put-instance-data.pl	475
mon-get-instance-stats.pl	477
Viewing Your Custom Metrics in the Console	479
Network and Security	480
Key Pairs	481
Creating Your Key Pair Using Amazon EC2	482
Importing Your Own Key Pair to Amazon EC2	482
Retrieving the Public Key for Your Key Pair on Linux	484
Retrieving the Public Key for Your Key Pair on Windows	485
Verifying Your Key Pair's Fingerprint	485
Deleting Your Key Pair	486
Connecting to Your Linux Instance if You Lose Your Private Key	486
Security Groups	489
Security Groups for EC2-Classic	490
Security Groups for EC2-VPC	490
Security Group Rules	491
Default Security Groups	492
Custom Security Groups	493
Creating a Security Group	494
Describing Your Security Groups	495
Adding Rules to a Security Group	495

Deleting Rules from a Security Group	496
Deleting a Security Group	496
API and Command Overview	497
Controlling Access	498
Network Access to Your Instance	498
Amazon EC2 Permission Attributes	498
IAM and Amazon EC2	498
IAM Policies	500
IAM Roles	536
Network Access	541
Amazon VPC	543
Benefits of Using a VPC	543
Differences Between EC2-Classic and EC2-VPC	544
Sharing and Accessing Resources Between EC2-Classic and EC2-VPC	546
Instance Types Available Only in a VPC	548
Amazon VPC Documentation	548
Supported Platforms	549
ClassicLink	550
Migrating from EC2-Classic to a VPC	560
Instance IP Addressing	570
Private IP Addresses and Internal DNS Hostnames	571
Public IP Addresses and External DNS Hostnames	572
Elastic IP Addresses	573
Amazon DNS Server	573
IP Address Differences Between EC2-Classic and EC2-VPC	573
Determining Your Public, Private, and Elastic IP Addresses	574
Assigning a Public IP Address	575
Multiple Private IP Addresses	576
Elastic IP Addresses	581
Elastic IP Address Basics	581
Elastic IP Address Differences for EC2-Classic and EC2-VPC	582
Working with Elastic IP Addresses	584
Using Reverse DNS for Email Applications	587
Elastic IP Address Limit	587
Elastic Network Interfaces	588
Private IP Addresses Per Network Interface Per Instance Type	589
Public IP Addresses for Network Interfaces	591
Creating a Management Network	591
Use Network and Security Appliances in Your VPC	592
Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets	592
Create a Low Budget High Availability Solution	592
Monitoring IP Traffic on Your Network Interface	593
Best Practices for Configuring Elastic Network Interfaces	593
Configuring Your Network Interface Using ec2-net-utils	593
Creating an Elastic Network Interface	594
Deleting an Elastic Network Interface	595
Viewing Details about an Elastic Network Interface	595
Attaching an Elastic Network Interface When Launching an Instance	596
Attaching an Elastic Network Interface to a Stopped or Running Instance	597
Detaching an Elastic Network Interface from an Instance	598
Changing the Security Group of an Elastic Network Interface	598
Changing the Source/Destination Checking of an Elastic Network Interface	599
Associating an Elastic IP Address with an Elastic Network Interface	599
Disassociating an Elastic IP Address from an Elastic Network Interface	600
Changing Termination Behavior for an Elastic Network Interface	600
Adding or Editing a Description for an Elastic Network Interface	601
Adding or Editing Tags for an Elastic Network Interface	601
Placement Groups	602

Placement Group Limitations	602
Launching Instances into a Placement Group	603
Deleting a Placement Group	604
Network MTU	605
Jumbo Frames (9001 MTU)	605
Path MTU Discovery	606
Check the Path MTU Between Two Hosts	606
Check and Set the MTU on your Amazon EC2 Instance	607
Troubleshooting	608
Enhanced Networking	608
Enhanced Networking Types	608
Enabling Enhanced Networking on Your Instance	608
Enabling Enhanced Networking: Intel 82599 VF	609
Enabling Enhanced Networking: ENA	617
Troubleshooting ENA	626
Storage	633
Amazon EBS	634
Features of Amazon EBS	635
EBS Volumes	636
EBS Snapshots	685
EBS Optimization	692
EBS Encryption	695
EBS Performance	699
Instance Store	715
Instance Store Lifetime	716
Instance Store Volumes	717
Add Instance Store Volumes	719
SSD Instance Store Volumes	722
Instance Store Swap Volumes	723
Optimizing Disk Performance	726
Amazon S3	726
Amazon S3 and Amazon EC2	727
Instance Volume Limits	728
Linux-Specific Volume Limits	728
Windows-Specific Volume Limits	729
Bandwidth vs Capacity	729
Device Naming	729
Available Device Names	730
Device Name Considerations	730
Block Device Mapping	731
Block Device Mapping Concepts	731
AMI Block Device Mapping	734
Instance Block Device Mapping	736
Using Public Data Sets	740
Public Data Set Concepts	740
Finding Public Data Sets	740
Creating a Public Data Set Volume from a Snapshot	741
Attaching and Mounting the Public Data Set Volume	742
Resources and Tags	743
Resource Locations	743
Resource IDs	744
Working with Longer IDs	745
Controlling Access to Longer ID Settings	748
Listing and Filtering Your Resources	748
Advanced Search	748
Listing Resources Using the Console	749
Filtering Resources Using the Console	750
Listing and Filtering Using the CLI and API	751

Tagging Your Resources	751
Tag Basics	752
Tag Restrictions	753
Tagging Your Resources for Billing	754
Working with Tags Using the Console	755
Working with Tags Using the CLI or API	760
Service Limits	760
Viewing Your Current Limits	761
Requesting a Limit Increase	762
Usage Reports	762
Available Reports	762
Getting Set Up for Usage Reports	762
Granting IAM Users Access to the Amazon EC2 Usage Reports	764
Instance Usage	764
Reserved Instance Utilization	767
Troubleshooting	773
Launching Your Instance	773
Getting the Reason for Instance Termination	774
Connecting to Your Instance	774
Error connecting to your instance: Connection timed out	775
Error: User key not recognized by server	776
Error: Host key not found, Permission denied (publickey), or Authentication failed, permission denied	777
Error: Unprotected Private Key File	778
Error: Server refused our key or No supported authentication methods available	778
Error using MindTerm on Safari Browser	779
Error Using Mac OS X RDP Client	779
Cannot Ping Instance	779
Stopping Your Instance	779
Terminating Your Instance	781
Delayed Instance Termination	781
Terminated Instance Still Displayed	781
Automatically Launch or Terminate Instances	781
Instance Recovery Failures	781
Failed Status Checks	782
Initial Steps	782
Retrieving System Logs	783
Troubleshooting System Log Errors for Linux-Based Instances	783
Out of memory: kill process	784
ERROR: mmu_update failed (Memory management update failed)	785
I/O error (Block device failure)	786
IO ERROR: neither local nor remote disk (Broken distributed block device)	787
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	788
"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)	789
"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)	789
ERROR Invalid kernel (EC2 incompatible kernel)	791
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	792
fsck: No such file or directory while trying to open... (File system not found)	793
General error mounting filesystems (Failed mount)	794
VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)	796
Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)	797
XENBUS: Device with no driver...	798
... days without being checked, check forced (File system check required)	798
fsck died with exit status... (Missing device)	799

GRUB prompt (grubdom>)	800
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)	802
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)	803
XENBUS: Timeout connecting to devices (Xenbus timeout)	804
Instance Capacity	805
Error: InsufficientInstanceCapacity	805
Error: InstanceLimitExceeded	805
Getting Console Output and Rebooting Instances	805
Instance Reboot	806
Instance Console Output	806
Capture a Screenshot of an Unreachable Instance	806
Instance Recovery When a Host Computer Fails	807
My Instance is Booting from the Wrong Volume	807
Making API Requests	810
Document History	811
AWS Glossary	825

What Is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

For more information about cloud computing, see [What is Cloud Computing?](#)

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IP addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds (VPCs)*

For more information about the features of Amazon EC2, see the [Amazon EC2 product page](#).

For more information about running your website on AWS, see [Websites & Website Hosting](#).

How to Get Started with Amazon EC2

The first thing you need to do is get set up to use Amazon EC2. After you are set up, you are ready to complete the Getting Started tutorial for Amazon EC2. Whenever you need more information about a feature of Amazon EC2, you can read the technical documentation.

Get Up and Running

- [Setting Up with Amazon EC2 \(p. 19\)](#)
- [Getting Started with Amazon EC2 Linux Instances \(p. 26\)](#)

Basics

- [Instances and AMIs \(p. 4\)](#)
- [Regions and Availability Zones \(p. 7\)](#)
- [Instance Types \(p. 156\)](#)
- [Tags \(p. 751\)](#)

Networking and Security

- [Amazon EC2 Key Pairs \(p. 481\)](#)
- [Security Groups \(p. 489\)](#)
- [Elastic IP Addresses \(p. 581\)](#)
- [Amazon EC2 and Amazon VPC \(p. 543\)](#)

Storage

- [Amazon EBS \(p. 634\)](#)
- [Instance Store \(p. 715\)](#)

Working with Linux Instances

- [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#)
- [Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS \(p. 51\)](#)
- [Getting Started with AWS: Hosting a Web App for Linux](#)

If you have questions about whether AWS is right for you, [contact AWS Sales](#). If you have technical questions about Amazon EC2, use the [Amazon EC2 forum](#).

Related Services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information, see the following documentation:

- [Auto Scaling User Guide](#)

- [AWS CloudFormation User Guide](#)
- [AWS Elastic Beanstalk Developer Guide](#)
- [AWS OpsWorks User Guide](#)

To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see [Elastic Load Balancing User Guide](#).

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the [Amazon CloudWatch Developer Guide](#).

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#).

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see [Amazon Relational Database Service Developer Guide](#).

Accessing Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have the following options:

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see [AWS Command Line Interface User Guide](#). For more information about the commands for Amazon EC2, see `ec2` in the *AWS Command Line Interface Reference*.

AWS Tools for Windows PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the [AWS Tools for Windows PowerShell User Guide](#). For more information about the cmdlets for Amazon EC2, see the [AWS Tools for Windows PowerShell Reference](#).

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see [Actions](#) in the *Amazon EC2 API Reference*.

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it easier for you to get started. For more information, see [AWS SDKs and Tools](#).

Pricing for Amazon EC2

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#).

Amazon EC2 provides the following purchasing options for instances:

On-Demand instances

Pay for the instances that you use by the hour, with no long-term commitments or up-front payments.

Reserved Instances

Make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances.

Spot instances

Specify the maximum hourly price that you are willing to pay to run a particular instance type. The Spot price fluctuates based on supply and demand, but you never pay more than the maximum price you specified. If the Spot price moves higher than your maximum price, Amazon EC2 shuts down your Spot instances.

For a complete list of charges and specific prices for Amazon EC2, see [Amazon EC2 Pricing](#).

To calculate the cost of a sample provisioned environment, see [AWS Economics Center](#).

To see your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

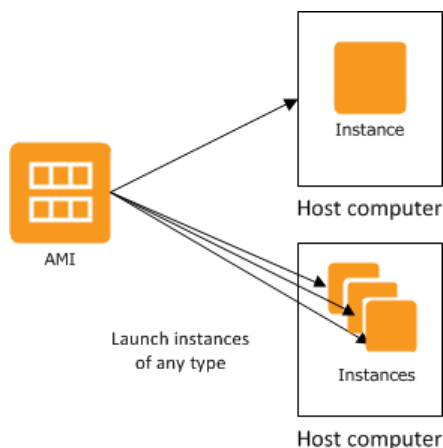
For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see [AWS Trusted Advisor](#).

PCI DSS Compliance

Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS Level 1](#).

Instances and AMIs

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the following figure.



Your instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Instances

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

After you launch an instance, it looks like a traditional host, and you can interact with it as you would any computer. You have complete control of your instances; you can use **sudo** to run commands that require root privileges.

Your AWS account has a limit on the number of instances that you can have running. For more information about this limit, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

Storage for Your Instance

The root device for your instance contains the image used to boot the instance. For more information, see [Amazon EC2 Root Device Volume \(p. 14\)](#).

Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping. For more information, see [Block Device Mapping \(p. 731\)](#). After these volumes have been added to and mapped on your instance, they are available for you to mount and use. If your instance fails, or if your instance is stopped or terminated, the data on these volumes is lost; therefore, these volumes are best used for temporary data. For important data, you should use a replication strategy across multiple instances in order to keep your data safe, or store your persistent data in Amazon S3 or Amazon EBS volumes. For more information, see [Storage \(p. 633\)](#).

Security Best Practices

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources, including your instances. You can create IAM users and groups under your AWS account, assign security credentials to each, and control the access that each has to resources and services in AWS. For more information, see [Controlling Access to Amazon EC2 Resources \(p. 498\)](#).
- Restrict access by only allowing trusted hosts or networks to access ports on your instance. For example, you can restrict SSH access by restricting incoming traffic on port 22. For more information, see [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#).
- Review the rules in your security groups regularly, and ensure that you apply the principle of *least privilege*—only open up permissions that you require. You can also create different security groups to deal with instances that have different security requirements. Consider creating a bastion security group that allows external logins, and keep the remainder of your instances in a group that does not allow external logins.
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk. For more information, see [Disable Password-Based Remote Logins for Root \(p. 82\)](#). For more information about sharing AMIs safely, see [Shared AMIs \(p. 76\)](#).

Stopping, Starting, and Terminating Instances

Stopping an instance

When an instance is stopped, the instance performs a normal shutdown, and then transitions to a `stopped` state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time.

You are not charged for additional instance hours while the instance is in a stopped state. A full instance hour will be charged for every transition from a stopped state to a running state, even if this happens multiple times within a single hour. If the instance type was changed while the instance was stopped, you will be charged the rate for the new instance type after the instance is started. All of the associated Amazon EBS usage of your instance, including root device usage, is billed using typical Amazon EBS prices.

When an instance is in a stopped state, you can attach or detach Amazon EBS volumes. You can also create an AMI from the instance, and you can change the kernel, RAM disk, and instance type.

Terminating an instance

When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume's `deleteOnTermination` attribute is set to `false`. The instance itself is also deleted, and you can't start the instance again at a later time.

To prevent accidental termination, you can disable instance termination. If you do so, ensure that the `disableApiTermination` attribute is set to `true` for the instance. To control the behavior of an instance shutdown, such as `shutdown -h` in Linux or `shutdown` in Windows, set the `instanceInitiatedShutdownBehavior` instance attribute to `stop` or `terminate` as desired. Instances with Amazon EBS volumes for the root device default to `stop`, and instances with instance-store root devices are always terminated as the result of an instance shutdown.

For more information, see [Instance Lifecycle \(p. 275\)](#).

AMIs

Amazon Web Services (AWS) publishes many Amazon Machine Images (AMIs) that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or a web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

All AMIs are categorized as either *backed by Amazon EBS*, which means that the root device for an instance launched from the AMI is an Amazon EBS volume, or *backed by instance store*, which means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

The description of an AMI indicates the type of root device (either `ebs` or `instance store`). This is important because there are significant differences in what you can do with each type of AMI. For more information about these differences, see [Storage for the Root Device \(p. 71\)](#).

Regions and Availability Zones

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each *region* is a separate geographic area. Each region has multiple, isolated locations known as *Availability Zones*. Amazon EC2 provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across regions unless you do so specifically.

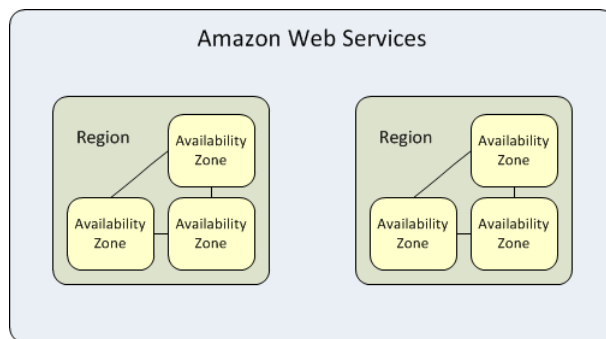
Amazon operates state-of-the-art, highly-available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all your instances in a single location that is affected by such a failure, none of your instances would be available.

Contents

- [Region and Availability Zone Concepts \(p. 7\)](#)
- [Available Regions \(p. 8\)](#)
- [Regions and Endpoints \(p. 9\)](#)
- [Describing Your Regions and Availability Zones \(p. 9\)](#)
- [Specifying the Region for a Resource \(p. 11\)](#)
- [Launching Instances in an Availability Zone \(p. 12\)](#)
- [Migrating an Instance to Another Availability Zone \(p. 13\)](#)

Region and Availability Zone Concepts

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.



Amazon EC2 resources are either global, tied to a region, or tied to an Availability Zone. For more information, see [Resource Locations \(p. 743\)](#).

Regions

Each Amazon EC2 region is designed to be completely isolated from the other Amazon EC2 regions. This achieves the greatest possible fault tolerance and stability.

When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically.

When you launch an instance, you must select an AMI that's in the same region. If the AMI is in another region, you can copy the AMI to the region you're using. For more information, see [Copying an AMI \(p. 135\)](#).

All communication between regions is across the public Internet. Therefore, you should use the appropriate encryption methods to protect your data. Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving instance. For more information, see [Amazon EC2 Pricing - Data Transfer](#).

Availability Zones

When you launch an instance, you can select an Availability Zone or let us choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone. For more information, see [Elastic IP Addresses \(p. 581\)](#).

An Availability Zone is represented by a region code followed by a letter identifier; for example, `us-east-1a`. To ensure that resources are distributed across the Availability Zones for a region, we independently map Availability Zones to identifiers for each account. For example, your Availability Zone `us-east-1a` might not be the same location as `us-east-1a` for another account. There's no way for you to coordinate Availability Zones between accounts.

As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new customers. Therefore, your account might have a different number of available Availability Zones in a region than another account.

You can list the Availability Zones that are available to your account. For more information, see [Describing Your Regions and Availability Zones \(p. 9\)](#).

Available Regions

Your account determines the regions that are available to you. For example:

- An AWS account provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.
- An AWS GovCloud (US) account provides access to the AWS GovCloud (US) region. For more information, see [AWS GovCloud \(US\) Region](#).
- An Amazon AWS account provides access to the China (Beijing) region.

The following table lists the regions provided by an AWS account. Note that you can't describe or access additional regions from an AWS account, such as AWS GovCloud (US) or China (Beijing).

Code	Name
<code>us-east-1</code>	US East (N. Virginia)
<code>us-west-2</code>	US West (Oregon)
<code>us-west-1</code>	US West (N. California)
<code>eu-west-1</code>	EU (Ireland)
<code>eu-central-1</code>	EU (Frankfurt)
<code>ap-southeast-1</code>	Asia Pacific (Singapore)

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Regions and Endpoints**

Code	Name
ap-northeast-1	Asia Pacific (Tokyo)
ap-southeast-2	Asia Pacific (Sydney)
ap-northeast-2	Asia Pacific (Seoul)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)

For more information, see [AWS Global Infrastructure](#).

The number and mapping of Availability Zones per region may vary between AWS accounts. To get a list of the Availability Zones that are available to your account, you can use the Amazon EC2 console or the command line interface. For more information, see [Describing Your Regions and Availability Zones \(p. 9\)](#).

Regions and Endpoints

When you work with an instance using the command line interface or API actions, you must specify its regional endpoint. For more information about the regions and endpoints for Amazon EC2, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

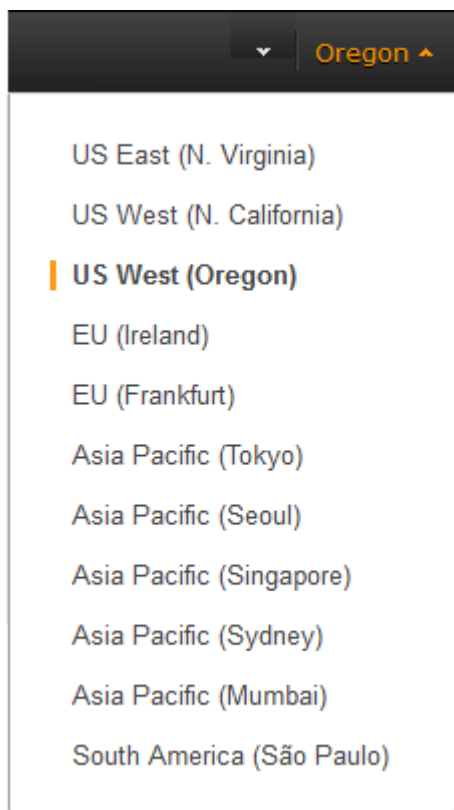
For more information about endpoints and protocols in AWS GovCloud (US), see [AWS GovCloud \(US\) Endpoints](#) in the *AWS GovCloud (US) User Guide*.

Describing Your Regions and Availability Zones

You can use the Amazon EC2 console or the command line interface to determine which regions and Availability Zones are available for your account. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To find your regions and Availability Zones using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, view the options in the region selector.



3. Your Availability Zones are listed on the dashboard under **Service Health**, under **Availability Zone Status**.

Service Health

Service Status:

- ✓ US East (N. Virginia):
This service is operating normally

Availability Zone Status:

- ✓ us-east-1b:
Availability zone is operating normally
- ✓ us-east-1c:
Availability zone is operating normally
- ✓ us-east-1d:
Availability zone is operating normally

To find your regions and Availability Zones using the command line

1. Use the [describe-regions](#) command as follows to describe your regions.

```
$ aws ec2 describe-regions  
{
```

```
"Regions": [
  {
    "Endpoint": "ec2.us-east-1.amazonaws.com",
    "RegionName": "us-east-1"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
    "RegionName": "ap-southeast-2"
  },
  ...
]
```

2. Use the [describe-availability-zones](#) command as follows to describe your Availability Zones within the `us-east-1` region.

```
$ aws ec2 describe-availability-zones --region us-east-1
{
  "AvailabilityZones": [
    {
      "State": "available",
      "RegionName": "us-east-1",
      "Messages": [],
      "ZoneName": "us-east-1b"
    },
    {
      "State": "available",
      "RegionName": "us-east-1",
      "Messages": [],
      "ZoneName": "us-east-1c"
    },
    {
      "State": "available",
      "RegionName": "us-east-1",
      "Messages": [],
      "ZoneName": "us-east-1d"
    }
  ]
}
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands: [Get-EC2Region](#) and [Get-EC2AvailabilityZone](#).

Specifying the Region for a Resource

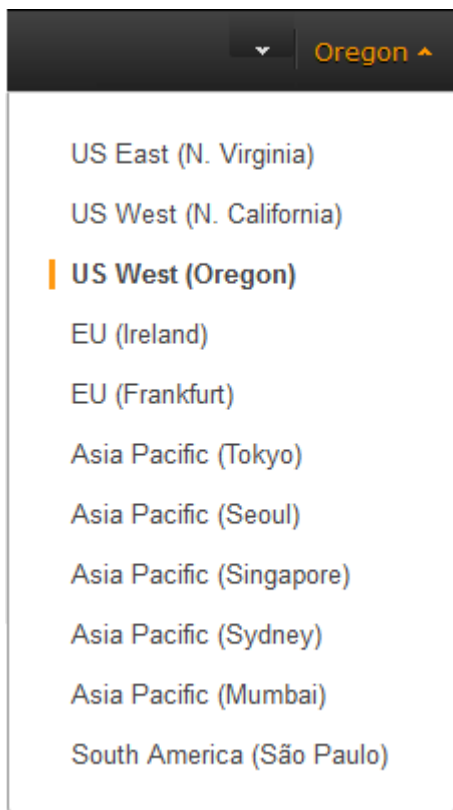
Every time you create an Amazon EC2 resource, you can specify the region for the resource. You can specify the region for a resource using the AWS Management Console or the command line.

Note

Some AWS resources might not be available in all regions and Availability Zones. Ensure that you can create the resources you need in the desired regions or Availability Zone before launching an instance in a specific Availability Zone.

To specify the region for a resource using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Use the region selector in the navigation bar.



To specify the default region using the command line

You can set the value of an environment variable to the desired regional endpoint (for example, `https://ec2.us-west-1.amazonaws.com`):

- `AWS_DEFAULT_REGION` (AWS CLI)
- `Set-AWSDefaultRegion` (AWS Tools for Windows PowerShell)

Alternatively, you can use the `--region` (AWS CLI) or `-Region` (AWS Tools for Windows PowerShell) command line option with each individual command. For example, `--region us-west-1`.

For more information about the endpoints for Amazon EC2, see [Amazon Elastic Compute Cloud Endpoints](#).

Launching Instances in an Availability Zone

When you launch an instance, select a region that puts your instances closer to specific customers, or meets the legal or other requirements you have. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location.

When you launch an instance, you can optionally specify an Availability Zone in the region that you are using. If you do not specify an Availability Zone, we select one for you. When you launch your

initial instances, we recommend that you accept the default Availability Zone, because this enables us to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, only specify an Availability Zone if your new instances must be close to, or separated from, your running instances.

To specify an Availability Zone for your instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. Follow the directions for the wizard. On the **Configure Instance Details** page, do the following:
 - [EC2-Classic] Select one of the Availability Zone options from the list, or select **No Preference** to enable us to select the best one for you.

Availability Zone ⓘ

- [EC2-VPC] Select one of the subnet options from the list, or select **No preference (default subnet in any Availability Zone)** to enable us to select the best one for you.

Subnet ⓘ

[Create new subnet](#)

To specify an Availability Zone for your instance using the AWS CLI

You can use the [run-instances](#) command with one of the following options:

- [EC2-Classic] `--placement`
- [EC2-VPC] `--subnet-id`

To specify an Availability Zone for your instance using the AWS Tools for Windows PowerShell

You can use the [New-EC2Instance](#) command with one of the following options:

- [EC2-Classic] `-AvailabilityZone`
- [EC2-VPC] `-SubnetId`

Migrating an Instance to Another Availability Zone

If you need to, you can migrate an instance from one Availability Zone to another. For example, if you are trying to modify the instance type of your instance and we can't launch an instance of the new instance type in the current Availability Zone, you could migrate the instance to an Availability Zone where we can launch an instance of that instance type.

The migration process involves creating an AMI from the original instance, launching an instance in the new Availability Zone, and updating the configuration of the new instance, as shown in the following procedure.

To migrate an instance to another Availability Zone

1. Create an AMI from the instance. The procedure depends on the operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:

- [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#)
 - [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#)
 - [Creating an Amazon EBS-Backed Windows AMI](#)
 - [Creating an Instance Store-Backed Windows AMI](#)
2. [EC2-VPC] If you need to preserve the private IP address of the instance, you must delete the subnet in the current Availability Zone and then create a subnet in the new Availability Zone with the same IP address range as the original subnet. Note that you must terminate all instances in a subnet before you can delete it. Therefore, you should move all instances in the current subnet to the new subnet.
 3. Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet. You can use the same instance type as the original instance, or select a new instance type. For more information, see [Launching Instances in an Availability Zone \(p. 12\)](#).
 4. If the original instance has an associated Elastic IP address, associate it with the new instance. For more information, see [Disassociating an Elastic IP Address and Reassociating it with a Different Instance \(p. 585\)](#).
 5. If the original instance is a Reserved Instance, change the Availability Zone for your reservation. (If you also changed the instance type, you can also change the instance type for your reservation.) For more information, see [Submitting Modification Requests \(p. 213\)](#).
 6. (Optional) Terminate the original instance. For more information, see [Terminating an Instance \(p. 306\)](#).

Amazon EC2 Root Device Volume

When you launch an instance, the *root device volume* contains the image used to boot the instance. When we introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. After we introduced Amazon EBS, we introduced AMIs that are backed by Amazon EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. You can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. We recommend that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

Topics

- [Root Device Storage Concepts \(p. 14\)](#)
- [Choosing an AMI by Root Device Type \(p. 16\)](#)
- [Determining the Root Device Type of Your Instance \(p. 17\)](#)
- [Changing the Root Device Volume to Persist \(p. 17\)](#)

Root Device Storage Concepts

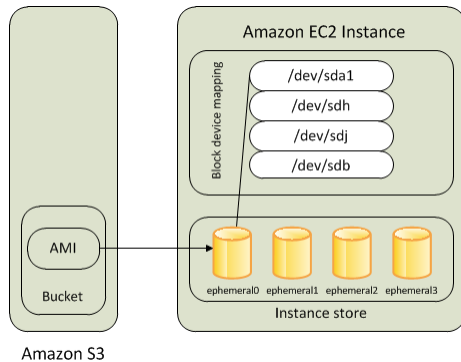
You can launch an instance from one of two types of AMIs: an instance store-backed AMI or an Amazon EBS-backed AMI. The description of an AMI includes which type of AMI it is; you'll see the root device referred to in some places as either *ebs* (for Amazon EBS-backed) or *instance store* (for instance store-backed). This is important because there are significant differences between what you can do with each type of AMI. For more information about these differences, see [Storage for the Root Device \(p. 71\)](#).

Instance Store-backed Instances

Instances that use instance stores for the root device automatically have instance store volumes available, with one serving as the root device volume. When an instance is launched, the image that

Amazon Elastic Compute Cloud User Guide for Linux Instances Root Device Storage Concepts

is used to boot the instance is copied to the root volume (typically `sda1`). Any data on the instance store volumes persists as long as the instance is running, but this data is deleted when the instance is terminated (instance store-backed instances do not support the **Stop** action) or if it fails (such as if an underlying drive has issues).

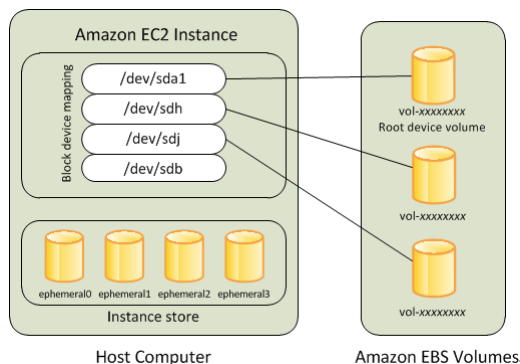


After an instance store-backed instance fails or terminates, it cannot be restored. If you plan to use Amazon EC2 instance store-backed instances, we highly recommend that you distribute the data on your instance stores across multiple Availability Zones. You should also back up the data on your instance store volumes to persistent storage on a regular basis.

For more information, see [Amazon EC2 Instance Store \(p. 715\)](#).

Amazon EBS-backed Instances

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each Amazon EBS snapshot referenced by the AMI you use. You can optionally use other Amazon EBS volumes or instance store volumes.



An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes. There are various instance- and volume-related tasks you can do when an Amazon EBS-backed instance is in a stopped state. For example, you can modify the properties of the instance, you can change the size of your instance or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose.

By default, the root device volume and the other Amazon EBS volumes attached when you launch an Amazon EBS-backed instance are automatically deleted when the instance terminates. For information about how to change this behavior when you launch an instance, see [Changing the Root Device Volume to Persist \(p. 17\)](#).

By default, any Amazon EBS volumes that you attach to a running instance are detached with their data intact when the instance terminates. You can attach a detached volume to any running instance.

If an Amazon EBS-backed instance fails, you can restore your session by following one of these methods:

- Stop and then start again (try this method first).
- Automatically snapshot all relevant volumes and create a new AMI. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#).
- Attach the volume to the new instance by following these steps:
 1. Create a snapshot of the root volume.
 2. Register a new AMI using the snapshot.
 3. Launch a new instance from the new AMI.
 4. Detach the remaining Amazon EBS volumes from the old instance.
 5. Reattach the Amazon EBS volumes to the new instance.

Choosing an AMI by Root Device Type

The AMI that you specify when you launch your instance determines the type of root device volume that your instance has.

To choose an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar click **Platform** to select the operating system (such as **Amazon Linux**), and **Root Device Type** to select **EBS images**.
4. (Optional) To get additional information to help you make your choice, click the **Show/Hide Columns** icon, update the columns to display, and click **Close**.
5. Choose an AMI and write down its AMI ID.

To choose an instance store-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar, click **Platform** to select the operating system (such as **Amazon Linux**), and **Root Device Type** to select **Instance store**.
4. (Optional) To get additional information to help you make your choice, click the **Show/Hide Columns** icon, update the columns to display, and click **Close**.
5. Choose an AMI and write down its AMI ID.

To verify the type of the root device volume of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-images](#) (AWS CLI)
- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

Determining the Root Device Type of Your Instance

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select the instance.
3. Check the value of **Root device type** in the **Description** tab as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed instance.
 - If the value is `instance store`, this is an instance store-backed instance.

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-instances` (AWS CLI)
- `Get-EC2Instance` (AWS Tools for Windows PowerShell)

Changing the Root Device Volume to Persist

By default, the root device volume for an AMI backed by Amazon EBS is deleted when the instance terminates. To change the default behavior, set the `DeleteOnTermination` attribute to `false` using a block device mapping.

Changing the Root Volume to Persist Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root device volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume of an Instance to Persist Using the AWS CLI

Using the AWS CLI, you can change the `DeleteOnTermination` attribute when you launch an instance or while the instance is running. The root device is typically `/dev/sda1` (Linux) or `xvda` (Windows).

Example at Launch

Use the `run-instances` command to preserve the root volume by including a block device mapping that sets its `DeleteOnTermination` attribute for to `false`.

```
$ aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings
' [{"DeviceName": "/dev/sda1", "Ebs": {"DeleteOnTermination": false}}]' other
parameters...
```

You can confirm that `DeleteOnTermination` is `false` by using the `describe-instances` command and looking for the `BlockDeviceMappings` entry for `/dev/sda1` in the command output, as shown here.

```
...
"BlockDeviceMappings": [
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "Status": "attached",
      "DeleteOnTermination": false,
      "VolumeId": "vol-1234567890abcdef0",
      "AttachTime": "2013-07-19T02:42:39.000Z"
    }
  }
]
...
```

Example While the Instance is Running

Use the `modify-instance-attribute` command to preserve the root volume by including a block device mapping that sets its `DeleteOnTermination` attribute to `false`.

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0
--block-device-mappings '[ {"DeviceName": "/dev/sda1", "Ebs":
{"DeleteOnTermination": false}} ]'
```

Setting Up with Amazon EC2

If you've already signed up for Amazon Web Services (AWS), you can start using Amazon EC2 immediately. You can open the Amazon EC2 console, click **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. [Sign Up for AWS \(p. 19\)](#)
2. [Create an IAM User \(p. 19\)](#)
3. [Create a Key Pair \(p. 21\)](#)
4. [Create a Virtual Private Cloud \(VPC\) \(p. 23\)](#)
5. [Create a Security Group \(p. 23\)](#)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see [AWS Free Tier](#).

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open <http://aws.amazon.com/>, and then choose **Create an AWS Account**.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console

requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or and grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see [Working with the AWS Management Console](#) for an overview.

To create a group for administrators

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups**, and then choose **Create New Group**.
3. For **Group Name**, type a name for your group, such as **Administrators**, and then choose **Next Step**.
4. In the list of policies, select the check box next to the **AdministratorAccess** policy. You can use the **Filter** menu and the **Search** box to filter the list of policies.
5. Choose **Next Step**, and then choose **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user for yourself, add the user to the administrators group, and create a password for the user

1. In the navigation pane, choose **Users**, and then choose **Create New Users**.
2. In box **1**, type a user name.
3. Clear the check box next to **Generate an access key for each user**.
4. Choose **Create**.
5. In the list of users, choose the name (not the check box) of the user you just created. You can use the **Search** box to search for the user name.
6. Choose the **Groups** tab and then choose **Add User to Groups**.
7. Select the check box next to the administrators group. Then choose **Add to Groups**.
8. Choose the **Security Credentials** tab. Under **Sign-In Credentials**, choose **Manage Password**.
9. Select **Assign a custom password**. Then type a password in the **Password** and **Confirm Password** boxes. When you are finished, choose **Apply**.

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name (not your email address) and password that you just created. When you're signed in, the navigation bar displays "*your_user_name @ your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM console, click **Dashboard** in the navigation pane. From the dashboard, click **Customize** and enter an alias such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see [IAM and Amazon EC2 \(p. 498\)](#).

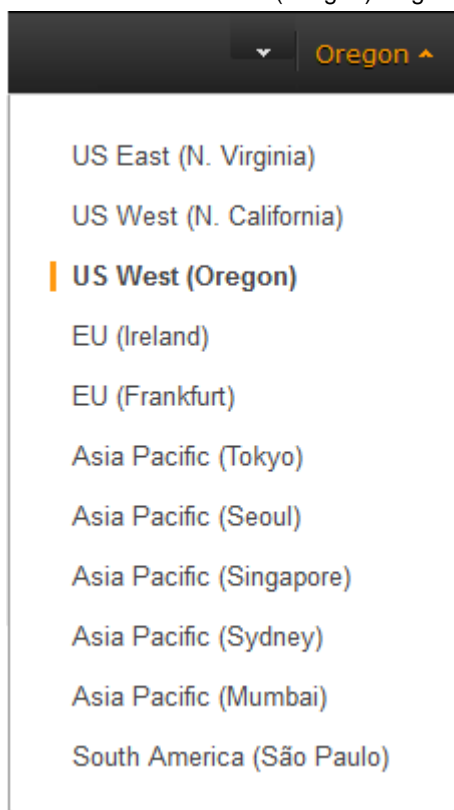
Create a Key Pair

AWS uses public-key cryptography to secure the login information for your instance. A Linux instance has no password; you use a key pair to log in to your instance securely. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region. For more information about regions, see [Regions and Availability Zones \(p. 7\)](#).

To create a key pair

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US West (Oregon) Region, you must create a key pair for the instance in the US West (Oregon) Region.



4. In the navigation pane, under **NETWORK & SECURITY**, click **Key Pairs**.

Tip

The navigation pane is on the left side of the console. If you do not see the pane, it might be minimized; click the arrow to expand the pane. You may have to scroll down to see the **Key Pairs** link.



5. Click **Create Key Pair**.
6. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**. Choose a name that is easy for you to remember, such as your IAM user name, followed by `-key-pair`, plus the region name. For example, `me-key-pair-uswest2`.
7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

8. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

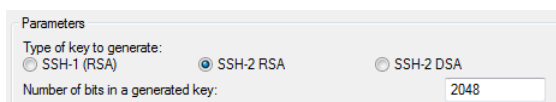
For more information, see [Amazon EC2 Key Pairs \(p. 481\)](#).

To connect to your instance using your key pair

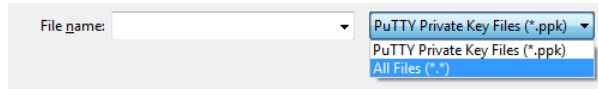
To connect to your Linux instance from a computer running Mac or Linux, you'll specify the `.pem` file to your SSH client with the `-i` option and the path to your private key. To connect to your Linux instance from a computer running Windows, you can use either MindTerm or PuTTY. If you plan to use PuTTY, you'll need to install it and use the following procedure to convert the `.pem` file to a `.ppk` file.

(Optional) To prepare to connect to a Linux instance from Windows using PuTTY

1. Download and install PuTTY from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Be sure to install the entire suite.
2. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
3. Under **Type of key to generate**, select **SSH-2 RSA**.



4. Click **Load**. By default, PuTTYgen displays only files with the extension `.ppk`. To locate your `.pem` file, select the option to display files of all types.



5. Select the private key file that you created in the previous procedure and click **Open**. Click **OK** to dismiss the confirmation dialog box.
6. Click **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.
7. Specify the same name for the key that you used for the key pair. PuTTY automatically adds the `.ppk` file extension.

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, [Create a Security Group \(p. 23\)](#). To determine whether you have a default VPC, see [Supported Platforms in the Amazon EC2 Console \(p. 549\)](#). Otherwise, you can create a nondefault VPC in your account using the steps below.

Important

If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

To create a nondefault VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, click **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and click **Create VPC**. On the confirmation page, click **OK**.

For more information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

Create a Security Group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region. For more information about regions, see [Regions and Availability Zones \(p. 7\)](#).

Prerequisites

You'll need the public IP address of your local computer, which you can get using a service. For example, we provide the following service: <http://checkip.amazonaws.com/>. To locate another service

that provides your IP address, use the search phrase "what is my IP address." If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

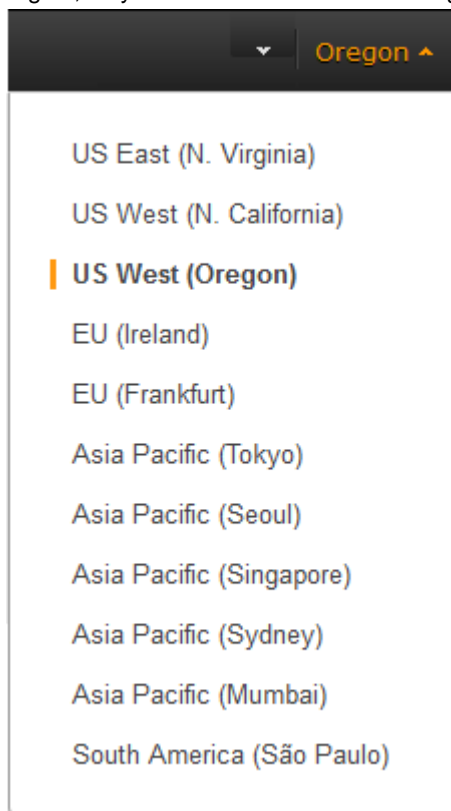
To create a security group with least privilege

1. Open the Amazon EC2 console.

Tip

Alternatively, you can use the Amazon VPC console to create a security group. However, the instructions in this procedure don't match the Amazon VPC console. Therefore, if you switched to the Amazon VPC console in the previous section, either switch back to the Amazon EC2 console and use these instructions, or use the instructions in [Set Up a Security Group for Your VPC](#) in the *Amazon VPC Getting Started Guide*.

2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.



3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description. Choose a name that is easy for you to remember, such as your IAM user name, followed by `_SG_`, plus the region name. For example, `me_SG_uswest2`.
6. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (*).

Note

If your account supports EC2-Classic, select the VPC that you created in the previous task.

7. On the **Inbound** tab, create the following rules (click **Add Rule** for each new rule), and then click **Create**:

- Select **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
- Select **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
- Select **SSH** from the **Type** list. In the **Source** box, ensure **Custom** is selected, and specify the public IP address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing prefix /32. For example, if your IP address is 203.0.113.25, specify 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

Caution

For security reasons, we don't recommend that you allow SSH access from all IP addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

For more information, see [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#).

Getting Started with Amazon EC2 Linux Instances

Let's get started with Amazon Elastic Compute Cloud (Amazon EC2) by launching, connecting to, and using a Linux instance. An *instance* is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

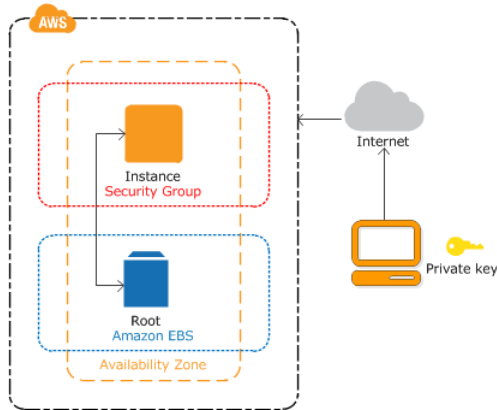
When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it will not cost you anything to complete this tutorial, because we help you select options that are within the free tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance (which is the final task of this tutorial), even if it remains idle.

Contents

- [Overview \(p. 26\)](#)
- [Prerequisites \(p. 27\)](#)
- [Step 1: Launch an Instance \(p. 27\)](#)
- [Step 2: Connect to Your Instance \(p. 28\)](#)
- [Step 3: Clean Up Your Instance \(p. 29\)](#)
- [Next Steps \(p. 30\)](#)

Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.



Tasks

To complete this tutorial, perform the following tasks:

1. [Launch an Instance](#) (p. 27)
2. [Connect to Your Instance](#) (p. 28)
3. [Clean Up Your Instance](#) (p. 29)

Related Tutorials

- If you'd prefer to launch a Windows instance, see this tutorial in the *Amazon EC2 User Guide for Windows Instances*: [Getting Started with Amazon EC2 Windows Instances](#).
- If you'd prefer to use the command line, see this tutorial in the *AWS Command Line Interface User Guide*: [Using Amazon EC2 through the AWS CLI](#).

Prerequisites

Before you begin, be sure that you've completed the steps in [Setting Up with Amazon EC2](#) (p. 19).

Step 1: Launch an Instance

You can launch a Linux instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see [Launching an Instance](#).

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Select the HVM edition of the Amazon Linux AMI. Notice that this AMI is marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the `t2.micro` type, which is selected by default. Notice that this instance type is eligible for the free tier.

Note

T2 instances, such as `t2.micro`, must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have a VPC in the selected region, the launch

wizard creates a VPC for you and you can continue to the next step. Otherwise, the **Review and Launch** button is disabled and you must choose **Next: Configure Instance Details** and follow the directions to select a subnet.

5. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
 - a. Choose **Edit security groups**.
 - b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
 - c. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Caution

Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
10. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. (If the **Public DNS** column is hidden, choose the Show/Hide icon in the top right corner of the page and then select **Public DNS**.)
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

Step 2: Connect to Your Instance

There are several ways to connect to a Linux instance. In this procedure, you'll connect using your browser. Alternatively, you can connect using PuTTY or an SSH client. It's also assumed that you followed the steps earlier and launched an instance from an Amazon Linux AMI, which has a specific user name. Other Linux distributions may use a different user name. For more information, see [Connecting to Your Linux Instance from Windows Using PuTTY \(p. 293\)](#) or [Connecting to Your Linux Instance Using SSH \(p. 289\)](#).

Important

You can't connect to your instance unless you launched it with a key pair for which you have the `.pem` file and you launched it with a security group that allows SSH access. If you can't connect to your instance, see [Troubleshooting Connecting to Your Instance \(p. 774\)](#) for assistance.

To connect to your Linux instance using a web browser

1. You must have Java installed and enabled in the browser. If you don't have Java already, you can contact your system administrator to get it installed, or follow the steps outlined in the following pages: [Install Java](#) and [Enable Java in your web browser](#).
2. From the Amazon EC2 console, choose **Instances** in the navigation pane.
3. Select the instance, and then choose **Connect**.
4. Choose **A Java SSH client directly from my browser (Java required)**.
5. Amazon EC2 automatically detects the public DNS name of your instance and populates **Public DNS** for you. It also detects the key pair that you specified when you launched the instance. Complete the following, and then choose **Launch SSH Client**.
 - a. In **User name**, enter `ec2-user`.
 - b. In **Private key path**, enter the fully qualified path to your private key (`.pem`) file, including the key pair name.
 - c. (Optional) Choose **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.
6. If necessary, choose **Yes** to trust the certificate, and choose **Run** to run the MindTerm client.
7. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, confirm setup for your home directory, and confirm setup of the known hosts directory. Confirm these settings.
8. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, choose **No**.
9. A window opens and you are connected to your instance.

Note

If you chose **No** in the previous step, you'll see the following message, which is expected:

```
Verification of server key disabled in this session.
```

Step 3: Clean Up Your Instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see [Next Steps](#) (p. 30).

Important

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the [AWS Free Tier](#), you'll stop incurring charges for that instance as soon as the instance status changes to `shutting down` or `terminated`. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see [Stopping Instances](#).

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Actions**, then **Instance State**, and then choose **Terminate**.
3. Choose **Yes, Terminate** when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.

Next Steps

After you start your instance, you might want to try some of the following exercises:

- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see [Create a Billing Alarm](#) in the *AWS Billing and Cost Management User Guide*.
- Add an EBS volume. For more information, see [Creating an Amazon EBS Volume \(p. 650\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).
- Install the LAMP stack. For more information, see [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#).

Best Practices for Amazon EC2

This checklist is intended to help you get the maximum benefit from and satisfaction with Amazon EC2.

Security and Network

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see [IAM Best Practices](#) in the *IAM User Guide*.
- Implement the least permissive rules for your security group. For more information, see [Security Group Rules](#) (p. 491).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux, see [Managing Software on Your Linux Instance](#). For more information about updating your Windows instance, see [Updating Your Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.
- Launch your instances into a VPC instead of EC2-Classic. Note that if you created your AWS account after 2013-12-04, we automatically launch your instances into a VPC. For more information about the benefits, see [Amazon EC2 and Amazon Virtual Private Cloud](#) (p. 543).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see [Storage for the Root Device](#) (p. 71).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination. For more information, see [Preserving Amazon EBS Volumes on Instance Termination](#) (p. 308).
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.

Resource Management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see [Instance Metadata and User Data](#) (p. 337) and [Tagging Your Amazon EC2 Resources](#) (p. 751).
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see [Amazon EC2 Service Limits](#) (p. 760).

Backup and Recovery

- Regularly back up your instance using [Amazon EBS snapshots \(p. 685\)](#) or a backup tool.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see [Amazon EC2 Instance IP Addressing \(p. 570\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 407\)](#).
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see [Elastic Network Interfaces \(ENI\) \(p. 588\)](#). For an automated solution, you can use Auto Scaling. For more information, see the [Auto Scaling User Guide](#).
- Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Tutorials for Amazon EC2 Instances Running Linux

The following tutorials show you how to perform common tasks using EC2 instances running Linux.

Tutorials

- [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#)
- [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 42\)](#)
- [Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS \(p. 51\)](#)
- [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 60\)](#)
- [Tutorial: Remotely Manage Your Amazon EC2 Instances \(p. 64\)](#)

Tutorial: Installing a LAMP Web Server on Amazon Linux

The following procedures help you install the Apache web server with PHP and MySQL support on your Amazon Linux instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Prerequisites

This tutorial assumes that you have already launched an instance with a public DNS name that is reachable from the Internet. For more information, see [Step 1: Launch an Instance \(p. 27\)](#). You must also have configured your security group to allow `SSH` (port 22), `HTTP` (port 80), and `HTTPS` (port 443) connections. For more information about these prerequisites, see [Setting Up with Amazon EC2 \(p. 19\)](#).

Important

If you are trying to set up a LAMP web server on an Ubuntu instance, this tutorial will not work for you. These procedures are intended for use with Amazon Linux. For more information

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Installing a LAMP
Web Server on Amazon Linux**

about other distributions, see their specific documentation. For information about LAMP web servers on Ubuntu, see the Ubuntu community documentation [ApacheMySQLPHP](#) topic.

To install and start the LAMP web server on Amazon Linux

1. [Connect to your instance \(p. 28\)](#).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure you have the latest security updates and bug fixes.

Note

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Now that your instance is current, you can install the Apache web server, MySQL, and PHP software packages. Use the `yum install` command to install multiple software packages and all related dependencies at the same time.

```
[ec2-user ~]$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

4. Start the Apache web server.

```
[ec2-user ~]$ sudo service httpd start
Starting httpd: [ OK ]
```

5. Use the `chkconfig` command to configure the Apache web server to start at each system boot.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

Tip

The `chkconfig` command does not provide any confirmation message when you successfully enable a service. You can verify that `httpd` is on by running the following command.

```
[ec2-user ~]$ chkconfig --list httpd
httpd          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

Here, `httpd` is on in runlevels 2, 3, 4, and 5 (which is what you want to see).

6. Test your web server. In a web browser, enter the public DNS address (or the public IP address) of your instance; you should see the Apache test page. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, choose **Show/Hide** and select **Public DNS**).

Tip

If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding Rules to a Security Group \(p. 495\)](#).

Important

If you are not using Amazon Linux, you may also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.

Amazon Linux AMI Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to "webmaster@example.com".

The [Amazon Linux AMI](#) is a supported and maintained Linux image provided by [Amazon Web Services](#) for use on [Amazon Elastic Compute Cloud \(Amazon EC2\)](#). It is designed to provide a stable, secure, and high performance execution environment for applications running on [Amazon EC2](#). It also includes packages that enable easy integration with [AWS](#), including launch configuration tools and many popular AWS libraries and tools. [Amazon Web Services](#) provides ongoing security and maintenance updates to all instances running the [Amazon Linux AMI](#). The [Amazon Linux AMI](#) is provided at no additional charge to [Amazon EC2 users](#).

Note

This test page appears only when there is no content in `/var/www/html`. When you add content to the document root, your content appears at the public DNS address of your instance instead of this test page.

Apache `httpd` serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which is owned by `root` by default.

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug 7 00:02 error
drwxr-xr-x 2 root root 4096 Jan 6 2012 html
drwxr-xr-x 3 root root 4096 Aug 7 00:02 icons
```

To allow `ec2-user` to manipulate files in this directory, you need to modify the ownership and permissions of the directory. There are many ways to accomplish this task; in this tutorial, you add a `www` group to your instance, and you give that group ownership of the `/var/www` directory and add write permissions for the group. Any members of that group will then be able to add, delete, and modify files for the web server.

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTTP servers. Thanks for using Apache and the Amazon Linux AMI!



To set file permissions

1. Add the `www` group to your instance.

```
[ec2-user ~]$ sudo groupadd www
```

2. Add your user (in this case, `ec2-user`) to the `www` group.

```
[ec2-user ~]$ sudo usermod -a -G www ec2-user
```

Important

You need to log out and log back in to pick up the new group. You can use the `exit` command, or close the terminal window.

3. Log out and then log back in again, and verify your membership in the `www` group.
 - a. Log out.

```
[ec2-user ~]$ exit
```

- b. Reconnect to your instance, and then run the following command to verify your membership in the `www` group.

```
[ec2-user ~]$ groups  
ec2-user wheel www
```

4. Change the group ownership of `/var/www` and its contents to the `www` group.

```
[ec2-user ~]$ sudo chown -R root:www /var/www
```

5. Change the directory permissions of `/var/www` and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

```
[ec2-user ~]$ sudo chmod 2775 /var/www  
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Recursively change the file permissions of `/var/www` and its subdirectories to add group write permissions.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now `ec2-user` (and any future members of the `www` group) can add, delete, and edit files in the Apache document root. Now you are ready to add content, such as a static website or a PHP application.

(Optional) Secure your web server

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you enter, the content of web pages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

For information about enabling HTTPS on your server, see [Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS](#).

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Installing a LAMP
Web Server on Amazon Linux**

To test your LAMP web server

If your server is installed and running, and your file permissions are set correctly, your `ec2-user` account should be able to create a simple PHP file in the `/var/www/html` directory that will be available from the Internet.

1. Create a simple PHP file in the Apache document root.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Tip

If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in [To set file permissions \(p. 36\)](#).

2. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page:

PHP Version 5.6.6

System	Linux ip-172-31-7-35 3.14.35-28.38.amzn1.x86_64 #1 SMP Wed Mar 11 22:50:37 UTC 2015 x86_64
Build Date	Mar 5 2015 23:26:53
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php-5.6.d
Additional .ini files parsed	/etc/php-5.6.d/20-bz2.ini, /etc/php-5.6.d/20-calendar.ini, /etc/php-5.6.d/20-ctype.ini, /etc/php-5.6.d/20-curl.ini, /etc/php-5.6.d/20-dom.ini, /etc/php-5.6.d/20-exif.ini, /etc/php-5.6.d/20-fileinfo.ini, /etc/php-5.6.d/20-ftp.ini, /etc/php-5.6.d/20-gettext.ini, /etc/php-5.6.d/20-iconv.ini, /etc/php-5.6.d/20-mysqld.ini, /etc/php-5.6.d/20-pdo.ini, /etc/php-5.6.d/20-phar.ini, /etc/php-5.6.d/20-posix.ini, /etc/php-5.6.d/20-shmop.ini, /etc/php-5.6.d/20-simplexml.ini, /etc/php-5.6.d/20-sockets.ini, /etc/php-5.6.d/20-sqlite3.ini, /etc/php-5.6.d/20-sysvmsg.ini, /etc/php-5.6.d/20-sysvshm.ini, /etc/php-5.6.d/20-sysvsem.ini, /etc/php-5.6.d/20-tokenizer.ini, /etc/php-5.6.d/20-xml.ini, /etc/php-5.6.d/20-xmlwriter.ini, /etc/php-5.6.d/20-xsl.ini, /etc/php-5.6.d/20-zip.ini, /etc/php-5.6.d/30-mysql.ini, /etc/php-5.6.d/30-mysqli.ini, /etc/php-5.6.d/30-pdo_mysql.ini, /etc/php-5.6.d/30-pdo_sqlite.ini, /etc/php-5.6.d/30-wddx.ini, /etc/php-5.6.d/30-xmlreader.ini, /etc/php-5.6.d/40-json.ini, /etc/php-5.6.d/php.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS

Note

If you do not see this page, verify that the `/var/www/html/phpinfo.php` file was created properly in the previous step. You can also verify that all of the required packages were installed with the following command (the package versions in the second column do not need to match this example output):

```
[ec2-user ~]$ sudo yum list installed httpd24 php56 mysql55-server  
php56-mysqld  
Loaded plugins: priorities, update-motd, upgrade-helper  
959 packages excluded due to repository priority protections  
Installed Packages
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Installing a LAMP
Web Server on Amazon Linux**

```
httpd24.x86_64 @amzn-main 2.4.16-1.62.amzn1
mysql55-server.x86_64 @amzn-main 5.5.45-1.9.amzn1
php56.x86_64 @amzn-main 5.6.13-1.118.amzn1
php56-mysqlnd.x86_64 @amzn-main 5.6.13-1.118.amzn1
```

If any of the required packages are not listed in your output, install them with the **sudo yum install *package*** command.

3. Delete the `phpinfo.php` file. Although this can be useful information to you, it should not be broadcast to the Internet for security reasons.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

To secure the MySQL server

The default installation of the MySQL server has several features that are great for testing and development, but they should be disabled or removed for production servers. The **mysql_secure_installation** command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MySQL server, performing this procedure is a good idea.

1. Start the MySQL server.

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database: Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...

Starting mysqld: [ OK ]
```

2. Run **mysql_secure_installation**.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. When prompted, enter a password for the `root` account.
 - i. Enter the current `root` password. By default, the `root` account does not have a password set, so press **Enter**.
 - ii. Type **Y** to set a password, and enter a secure password twice. For more information about creating a secure password, see <http://www.pctools.com/guides/password/>. Make sure to store this password in a safe place.

Note

Setting a root password for MySQL is only the most basic measure for securing your database. When you build or install a database-driven application, you

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Installing a LAMP
Web Server on Amazon Linux**

typically create a database service user for that application and avoid using the root account for anything but database administration.

- b. Type **Y** to remove the anonymous user accounts.
 - c. Type **Y** to disable remote `root` login.
 - d. Type **Y** to remove the test database.
 - e. Type **Y** to reload the privilege tables and save your changes.
3. (Optional) Stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

```
[ec2-user ~]$ sudo service mysqld stop
Stopping mysqld: [ OK ]
```

4. (Optional) If you want the MySQL server to start at every boot, enter the following command.

```
[ec2-user ~]$ sudo chkconfig mysqld on
```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

(Optional) Install phpMyAdmin

[phpMyAdmin](#) is a web-based database management tool that you can use to view and edit the MySQL databases on your EC2 instance. Follow the steps below to install and configure phpMyAdmin on your Amazon Linux instance.

Important

We do not recommend using phpMyAdmin to access a LAMP server unless you have enabled SSL/TLS in Apache; otherwise, your database administrator password and other data will be transmitted insecurely across the Internet. For information about configuring a secure web server on an EC2 instance, see [Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS](#).

1. Enable the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project on your instance.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

2. Install the phpMyAdmin package.

```
[ec2-user ~]$ sudo yum install -y phpMyAdmin
```

Note

Answer **y** to import the GPG key for the EPEL repository when prompted.

3. Configure your phpMyAdmin installation to allow access from your local machine. By default, phpMyAdmin only allows access from the server that it is running on, which is not very useful because Amazon Linux does not include a web browser.
 - a. Find your local IP address by visiting a service such as [whatismyip.com](#).
 - b. Edit the `/etc/httpd/conf.d/phpMyAdmin.conf` file and replace the server IP address (127.0.0.1) with your local IP address with the following command, replacing `your_ip_address` with the local IP address that you identified in the previous step.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Installing a LAMP
Web Server on Amazon Linux**

```
[ec2-user ~]$ sudo sed -i -e 's/127.0.0.1/your_ip_address/g' /etc/  
httpd/conf.d/phpMyAdmin.conf
```

- Restart the Apache web server to pick up the new configuration.

```
[ec2-user ~]$ sudo service httpd restart  
Stopping httpd: [ OK ]  
Starting httpd: [ OK ]
```

- Restart the MySQL server to pick up the new configuration.

```
[ec2-user ~]$ sudo service mysqld restart  
Stopping mysqld: [ OK ]  
Starting mysqld: [ OK ]
```

- In a web browser, enter the URL of your phpMyAdmin installation. This URL is the public DNS address of your instance followed by a forward slash and `phpmyadmin`. For example:

```
http://my.public.dns.amazonaws.com/phpmyadmin
```

You should see the phpMyAdmin login page:



Welcome to phpMyAdmin

Language

English

Log in ⓘ

Username:

Password:

Go

Note

If you get a 403 Forbidden error, verify that you have set the correct IP address in the `/etc/httpd/conf.d/phpMyAdmin.conf` file. You can see what IP address the Apache server is actually getting your requests from by viewing the Apache access log with the following command:

```
[ec2-user ~]$ sudo tail -n 1 /var/log/httpd/access_log | awk  
'{ print $1 }'
```



```
205.251.233.48
```

Repeat [Step 3.b \(p. 39\)](#), replacing the incorrect address that you previously entered with the address returned here; for example:

```
[ec2-user ~]$ sudo sed -i -e 's/previous_ip_address/205.251.233.48/  
g' /etc/httpd/conf.d/phpMyAdmin.conf
```

After you've replaced the IP address, restart the `httpd` service with [Step 4 \(p. 40\)](#).

7. Log into your `phpMyAdmin` installation with the `root` user name and the MySQL root password you created earlier. For more information about using `phpMyAdmin`, see the [phpMyAdmin User Guide](#).

Related Topics

For more information on transferring files to your instance or installing a WordPress blog on your web server, see the following topics:

- [Transferring Files to Your Linux Instance Using WinSCP \(p. 296\)](#)
- [Transferring Files to Linux Instances from Linux Using SCP \(p. 291\)](#)
- [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 42\)](#)

For more information about the commands and software used in this topic, see the following web pages:

- Apache web server: <http://httpd.apache.org/>
- MySQL database server: <http://www.mysql.com/>
- PHP programming language: <http://php.net/>
- The `chmod` command: <https://en.wikipedia.org/wiki/Chmod>
- The `chown` command: <https://en.wikipedia.org/wiki/Chown>

If you are interested in registering a domain name for your web server, or transferring an existing domain name to this host, see [Creating and Migrating Domains and Subdomains to Amazon Route 53](#) in the *Amazon Route 53 Developer Guide*.

Tutorial: Hosting a WordPress Blog with Amazon Linux

The following procedures will help you install, configure, and secure a WordPress blog on your Amazon Linux instance.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation. Many steps in this tutorial do not work on Ubuntu instances. For help installing WordPress on an Ubuntu instance, see [WordPress](#) in the Ubuntu documentation.

Prerequisites

This tutorial assumes that you have launched an Amazon Linux instance with a functional web server with PHP and MySQL support by following all of the steps in [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#). This tutorial also has steps for configuring a security group to allow HTTP and HTTPS traffic, as well as several steps to ensure that file permissions are set properly for your web server. If you have not already done so, see [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#) to meet these prerequisites and then return to this tutorial to install WordPress. For information about adding rules to your security group, see [Adding Rules to a Security Group \(p. 495\)](#).

We strongly recommend that you associate an Elastic IP address (EIP) to the instance you are using to host a WordPress blog. This prevents the public DNS address for your instance from changing and breaking your installation. If you own a domain name and you want to use it for your blog, you can update the DNS record for the domain name to point to your EIP address (for help with this, contact your domain name registrar). You can have one EIP address associated with a running instance at no charge. For more information, see [Elastic IP Addresses \(p. 581\)](#).

If you don't already have a domain name for your blog, you can register a domain name with Amazon Route 53 and associate your instance's EIP address with your domain name. For more information, see [Registering Domain Names Using Amazon Route 53](#) in the *Amazon Route 53 Developer Guide*.

Install WordPress

This tutorial is a good introduction to using Amazon EC2 in that you have full control over a web server that hosts your WordPress blog, which is not typical with a traditional hosting service. Of course, that means that you are responsible for updating the software packages and maintaining security patches for your server as well. For a more automated WordPress installation that does not require direct interaction with the web server configuration, the AWS CloudFormation service provides a WordPress template that can also get you started quickly. For more information, see [Getting Started](#) in the *AWS CloudFormation User Guide*. If you'd prefer to host your WordPress blog on a Windows instance, see [Deploying a WordPress Blog on Your Amazon EC2 Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

To download and unzip the WordPress installation package

1. Download the latest WordPress installation package with the **wget** command. The following command should always download the latest release.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
--2013-08-09 17:19:01-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 66.155.40.249, 66.155.40.250
Connecting to wordpress.org (wordpress.org)|66.155.40.249|:443...
  connected.
HTTP request sent, awaiting response... 200 OK
Length: 4028740 (3.8M) [application/x-gzip]
Saving to: latest.tar.gz

100%[=====>] 4,028,740    20.1MB/s   in
0.2s

2013-08-09 17:19:02 (20.1 MB/s) - latest.tar.gz saved [4028740/4028740]
```

2. Unzip and unarchive the installation package. The installation folder is unzipped to a folder called `wordpress`.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
[ec2-user ~]$ ls
```

```
latest.tar.gz wordpress
```

To create a MySQL user and database for your WordPress installation

Your WordPress installation needs to store information, such as blog post entries and user comments, in a database. This procedure helps you create a database for your blog and a user that is authorized to read and save information to that database.

1. Start the MySQL server.

```
[ec2-user ~]$ sudo service mysqld start
```

2. Log in to the MySQL server as the `root` user. Enter your MySQL `root` password when prompted; this may be different than your `root` system password, or it may even be empty if you have not secured your MySQL server.

Important

If you have not secured your MySQL server yet, it is very important that you do so. For more information, see [To secure the MySQL server \(p. 38\)](#).

```
[ec2-user ~]$ mysql -u root -p
Enter password:
```

3. Create a user and password for your MySQL database. Your WordPress installation uses these values to communicate with your MySQL database. Enter the following command, substituting a unique user name and password.

```
mysql> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY
'your_strong_password';
Query OK, 0 rows affected (0.00 sec)
```

Make sure that you create a strong password for your user. Do not use the single quote character (') in your password, because this will break the preceding command. For more information about creating a secure password, go to <http://www.pctools.com/guides/password/>. Do not reuse an existing password, and make sure to store this password in a safe place.

4. Create your database. Give your database a descriptive, meaningful name, such as `wordpress-db`.

Note

The punctuation marks surrounding the database name in the command below are called backticks. The backtick (`) key is usually located above the **Tab** key on a standard keyboard. Backticks are not always required, but they allow you to use otherwise illegal characters, such as hyphens, in database names.

```
mysql> CREATE DATABASE `wordpress-db`;
Query OK, 1 row affected (0.01 sec)
```

5. Grant full privileges for your database to the WordPress user that you created earlier.

```
mysql> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-
user"@"localhost";
Query OK, 0 rows affected (0.00 sec)
```

6. Flush the MySQL privileges to pick up all of your changes.

```
mysql> FLUSH PRIVILEGES;
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Install WordPress

```
Query OK, 0 rows affected (0.01 sec)
```

7. Exit the `mysql` client.

```
mysql> exit  
Bye
```

To create and edit the `wp-config.php` file

The WordPress installation folder contains a sample configuration file called `wp-config-sample.php`. In this procedure, you copy this file and edit it to fit your specific configuration.

1. Copy the `wp-config-sample.php` file to a file called `wp-config.php`. This creates a new configuration file and keeps the original sample file intact as a backup.

```
[ec2-user ~]$ cd wordpress/  
[ec2-user wordpress]$ cp wp-config-sample.php wp-config.php
```

2. Edit the `wp-config.php` file with your favorite text editor (such as **nano** or **vim**) and enter values for your installation. If you do not have a favorite text editor, `nano` is much easier for beginners to use.

```
[ec2-user wordpress]$ nano wp-config.php
```

- a. Find the line that defines `DB_NAME` and change `database_name_here` to the database name that you created in [Step 4 \(p. 44\) of To create a MySQL user and database for your WordPress installation \(p. 44\)](#).

```
define('DB_NAME', 'wordpress-db');
```

- b. Find the line that defines `DB_USER` and change `username_here` to the database user that you created in [Step 3 \(p. 44\) of To create a MySQL user and database for your WordPress installation \(p. 44\)](#).

```
define('DB_USER', 'wordpress-user');
```

- c. Find the line that defines `DB_PASSWORD` and change `password_here` to the strong password that you created in [Step 3 \(p. 44\) of To create a MySQL user and database for your WordPress installation \(p. 44\)](#).

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. Find the section called Authentication Unique Keys and Salts. These `KEY` and `SALT` values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure. Visit <https://api.wordpress.org/secret-key/1.1/salt/> to randomly generate a set of key values that you can copy and paste into your `wp-config.php` file. To paste text into a PuTTY terminal, place the cursor where you want to paste the text and right-click your mouse inside the PuTTY terminal.

For more information about security keys, go to http://codex.wordpress.org/Editing_wp-config.php#Security_Keys.

Note

The values below are for example purposes only; do not use these values for your installation.

Amazon Elastic Compute Cloud User Guide for Linux Instances Install WordPress

```
define('AUTH_KEY', ' #U$${RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/  
Aj[wTwSiZ<Qb[mghEXcRh-');  
define('SECURE_AUTH_KEY', ' Zsz._P=l/  
y.Lq)Xj1kwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?6OP$eJT@;+(ndLg');  
define('LOGGED_IN_KEY', ' ju}qwre3V*+8f_zOWF?{L1GsQ]Ye@2Jh^,8x>)Y |;  
(^[Iw]Pi+LG#A4R?7N`YB3');  
define('NONCE_KEY', ' P(g62HeZxEes|LnI^i=H,[Xwk9I&[2s|:?0N}VJM  
%?;v2v]v+;+^9eXUahg@::Cj');  
define('AUTH_SALT', ' C$DpB4Hj[JK:#{ql`sRVa:{:7yShy(9A@5wg  
+`JJVb1fk%-Bx*M4(qc[Qg%JT!h');  
define('SECURE_AUTH_SALT', ' d!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&  
%~gL>U>NV<zpD-@2-Es7Q10-bp28EKv');  
define('LOGGED_IN_SALT', ' ;j{00P*owZf)kVD+FVLn~>.|Y  
%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');  
define('NONCE_SALT', ' -97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|  
_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

- e. Save the file and exit your text editor.

To move your WordPress installation to the Apache document root

Now that you've unzipped the installation folder, created a MySQL database and user, and customized the WordPress configuration file, you are ready to move your installation files to your web server document root so you can run the installation script that completes your installation. The location of these files depends on whether you want your WordPress blog to be available at the root of your web server (for example, my.public.dns.amazonaws.com) or in a subdirectory or folder (for example, my.public.dns.amazonaws.com/blog).

- Choose the location where you want your blog to be available and only run the **mv** associated with that location.

Important

If you run both sets of commands below, you will get an error message on the second **mv** command because the files you are trying to move are no longer there.

- To make your blog available at my.public.dns.amazonaws.com, move the files in the `wordpress` folder (but not the folder itself) to the Apache document root (`/var/www/html` on Amazon Linux instances).

```
[ec2-user wordpress]$ mv * /var/www/html/
```

- OR, to make your blog available at my.public.dns.amazonaws.com/blog instead, create a new folder called `blog` inside the Apache document root and move the files in the `wordpress` folder (but not the folder itself) to the new `blog` folder.

```
[ec2-user wordpress]$ mkdir /var/www/html/blog  
[ec2-user wordpress]$ mv * /var/www/html/blog
```

Important

For security purposes, if you are not moving on to the next procedure immediately, stop the Apache web server (`httpd`) now. After you move your installation to the Apache document root, the WordPress installation script is unprotected and an attacker could gain access to your blog if the Apache web server were running. To stop the Apache web server, enter the command **sudo service httpd stop**. If you are moving on to the next procedure, you do not need to stop the Apache web server.

To allow WordPress to use permalinks

WordPress permalinks need to use Apache `.htaccess` files to work properly, but this is not enabled by default on Amazon Linux. Use this procedure to allow all overrides in the Apache document root.

1. Open the `httpd.conf` file with your favorite text editor (such as **nano** or **vim**). If you do not have a favorite text editor, nano is much easier for beginners to use.

```
[ec2-user wordpress]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. Find the section that starts with `<Directory "/var/www/html">`.

```
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI
MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess
files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

3. Change the `AllowOverride None` line in the above section to read `AllowOverride All`.

Note

There are multiple `AllowOverride` lines in this file; be sure you change the line in the `<Directory "/var/www/html">` section.

```
AllowOverride All
```

4. Save the file and exit your text editor.

To fix file permissions for the Apache web server

Some of the available features in WordPress require write access to the Apache document root (such as uploading media through the Administration screens). The web server runs as the `apache` user, so you need to add that user to the `www` group that was created in the [LAMP web server tutorial \(p. 33\)](#).

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Install WordPress**

1. Add the `apache` user to the `www` group.

```
[ec2-user wordpress]$ sudo usermod -a -G www apache
```

2. Change the file ownership of `/var/www` and its contents to the `apache` user.

```
[ec2-user wordpress]$ sudo chown -R apache /var/www
```

3. Change the group ownership of `/var/www` and its contents to the `www` group.

```
[ec2-user wordpress]$ sudo chgrp -R www /var/www
```

4. Change the directory permissions of `/var/www` and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

```
[ec2-user wordpress]$ sudo chmod 2775 /var/www  
[ec2-user wordpress]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. Recursively change the file permissions of `/var/www` and its subdirectories to add group write permissions.

```
[ec2-user wordpress]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

6. Restart the Apache web server to pick up the new group and permissions.

```
[ec2-user wordpress]$ sudo service httpd restart  
Stopping httpd: [ OK ]  
Starting httpd: [ OK ]
```

To run the WordPress installation script

1. Use the `chkconfig` command to ensure that the `httpd` and `mysqld` services start at every system boot.

```
[ec2-user wordpress]$ sudo chkconfig httpd on  
[ec2-user wordpress]$ sudo chkconfig mysqld on
```

2. Verify that the MySQL server (`mysqld`) is running.

```
[ec2-user wordpress]$ sudo service mysqld status  
mysqld (pid 4746) is running...
```

If the `mysqld` service is not running, start it.

```
[ec2-user wordpress]$ sudo service mysqld start  
Starting mysqld: [ OK ]
```

3. Verify that your Apache web server (`httpd`) is running.

```
[ec2-user wordpress]$ sudo service httpd status  
httpd (pid 502) is running...
```

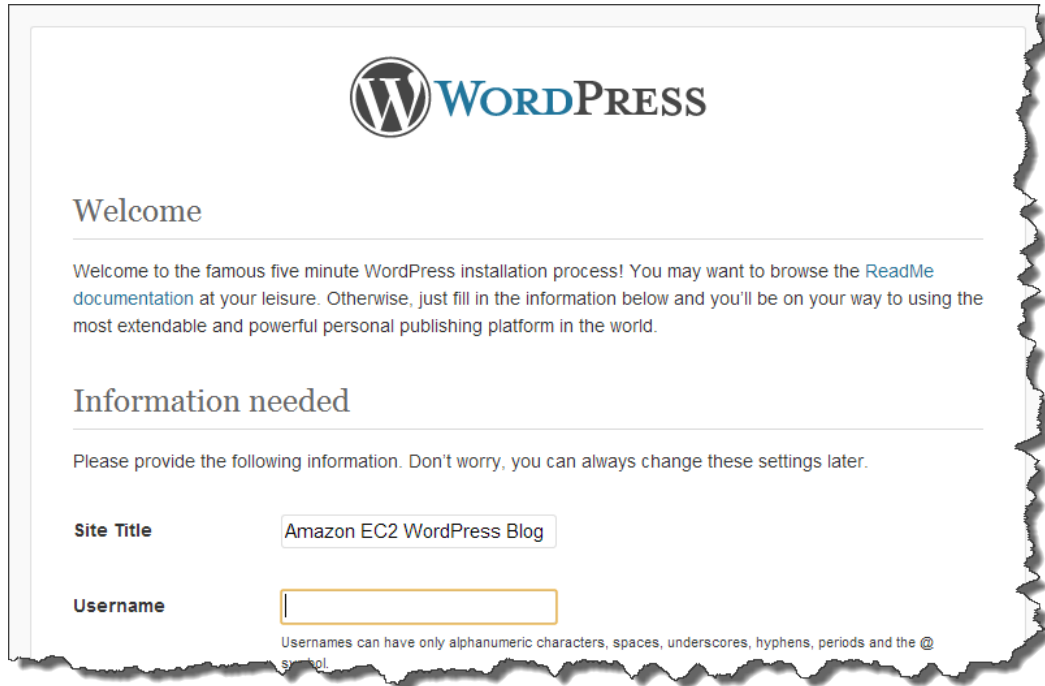
If the `httpd` service is not running, start it.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Install WordPress**

```
[ec2-user wordpress]$ sudo service httpd start  
Starting httpd: [ OK ]
```

- In a web browser, enter the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the `blog` folder). You should see the WordPress installation screen.

```
http://my.public.dns.amazonaws.com
```



- Enter the remaining installation information into the WordPress installation wizard.

Field	Value
Site Title	Enter a name for your WordPress site.
Username	Enter a name for your WordPress administrator. For security purposes, you should choose a unique name for this user, because it will be more difficult to exploit than the default user name, <code>admin</code> .
Password	Enter a strong password, and then enter it again to confirm. Do not reuse an existing password, and make sure to store this password in a safe place.
Your E-mail	Enter the email address you want to use for notifications.

- Click **Install WordPress** to complete the installation.

Congratulations, you should now be able to log into your WordPress blog and start posting entries.

Next Steps

After you have tested your initial WordPress blog, consider updating its configuration.

Use a Custom Domain Name

If you have a domain name associated with your EC2 instance's EIP address, you can configure your blog to use that name instead of the EC2 public DNS address. For more information, see http://codex.wordpress.org/Changing_The_Site_URL.

Configure Your Blog

You can configure your blog to use different [themes](#) and [plugins](#) to offer a more personalized experience for your readers. However, sometimes the installation process can backfire, causing you to lose your entire blog. We strongly recommend that you create a backup Amazon Machine Image (AMI) of your instance before attempting to install any themes or plugins so you can restore your blog if anything goes wrong during installation. For more information, see [Creating Your Own AMI \(p. 70\)](#).

Increase Capacity

If your WordPress blog becomes popular and you need more compute power or storage, consider the following steps:

- Expand the storage space on your instance. For more information, see [Expanding the Storage Space of an EBS Volume on Linux \(p. 671\)](#).
- Move your MySQL database to [Amazon RDS](#) to take advantage of the service's ability to scale automatically.
- Migrate to a larger instance type. For more information, see [Resizing Your Instance \(p. 189\)](#).
- Add additional instances. For more information, see [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 60\)](#).

Learn More about WordPress

For information about WordPress, see the WordPress Codex help documentation at <http://codex.wordpress.org/>. For more information about troubleshooting your installation, go to http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, go to http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, go to http://codex.wordpress.org/Updating_WordPress.

Help! My Public DNS Name Changed and now my Blog is Broken

Your WordPress installation is automatically configured using the public DNS address for your EC2 instance. If you stop and restart the instance, the public DNS address changes (unless it is associated with an Elastic IP address) and your blog will not work anymore because it references resources at an address that no longer exists (or is assigned to another EC2 instance). A more detailed description of the problem and several possible solutions are outlined in http://codex.wordpress.org/Changing_The_Site_URL.

If this has happened to your WordPress installation, you may be able to recover your blog with the procedure below, which uses the **wp-cli** command line interface for WordPress.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tutorial: Configure Apache Web Server
on Amazon Linux to use SSL/TLS**

To change your WordPress site URL with the wp-cli

1. Connect to your EC2 instance with SSH.
2. Note the old site URL and the new site URL for your instance. The old site URL is likely the public DNS name for your EC2 instance when you installed WordPress. The new site URL is the current public DNS name for your EC2 instance. If you are not sure of your old site URL, you can use **curl** to find it with the following command.

```
[ec2-user ~]$ curl localhost | grep wp-content
```

You should see references to your old public DNS name in the output, which will look like this (old site URL in red):

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

3. Download the **wp-cli** with the following command.

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. Search and replace the old site URL in your WordPress installation with the following command. Substitute the old and new site URLs for your EC2 instance and the path to your WordPress installation (usually `/var/www/html` or `/var/www/html/blog`).

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/path/to/wordpress/installation --skip-columns=guid
```

5. In a web browser, enter the new site URL of your WordPress blog to verify that the site is working properly again. If it is not, see http://codex.wordpress.org/Changing_The_Site_URL and http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems for more information.

Tutorial: Configure Apache Web Server on Amazon Linux to use SSL/TLS

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on a single instance of Amazon Linux running Apache web server. The [AWS Certificate Manager](#), not discussed here, is a good option as well, especially if you need to manage several domains or will be offering commercial-grade services.

Note

For historical reasons, web encryption is often referred to as simply SSL. While web browsers still support SSL, its successor protocol TLS is considered to be less vulnerable to attack. Amazon Linux disables SSL version 2 by default, and this tutorial recommends disabling SSL version 3 as well, as described below. For more information about the proposed updated encryption standard, go to [RFC7568](#).

Important

These procedures are intended for use with Amazon Linux. If you are trying to set up a LAMP web server on an instance of a different distribution, this tutorial will not work for you. For information about LAMP web servers on Ubuntu, go to the Ubuntu community documentation [ApacheMySQLPHP](#) topic. For information about Red Hat Enterprise Linux, go to the Customer Portal topic [Web Servers](#).

Topics

- [Prerequisites \(p. 52\)](#)
- [Step 1: Enable SSL/TLS on the Server \(p. 52\)](#)
- [Step 2: Obtain a CA-signed Certificate \(p. 54\)](#)
- [Step 3: Test and Harden the Security Configuration \(p. 57\)](#)
- [Troubleshooting \(p. 60\)](#)

Prerequisites

Before you begin this tutorial, complete the following steps:

- Launch an Amazon Linux instance. For more information, see [Step 1: Launch an Instance \(p. 27\)](#).
- Configure your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information, see [Setting Up with Amazon EC2 \(p. 19\)](#).
- Install Apache web server. For step-by-step instructions, see [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#). Only the httpd package and its dependencies are needed; you can ignore the instructions involving PHP and MySQL.
- The SSL/TLS public key infrastructure (PKI) relies on the Domain Name System (DNS) to identify and authenticate web sites. If you plan to use your EC2 instance to host a public web site, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host. Numerous third-party domain registration and DNS hosting services are available for this, or you may use [Amazon Route 53](#).

Step 1: Enable SSL/TLS on the Server

This procedure takes you through the process of setting up SSL/TLS on Amazon Linux with a self-signed, digital certificate.

To enable SSL/TLS on a server

1. [Connect to your instance \(p. 28\)](#) and confirm that Apache is running.

```
[ec2-user ~]$ sudo service httpd status
```

If necessary, start Apache.

```
[ec2-user ~]$ sudo service httpd start
```

2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure you have the latest security updates and bug fixes.

Note

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Now that your instance is current, add SSL/TLS support by installing the Apache module `mod_ssl`:

```
[ec2-user ~]$ sudo yum install -y mod24_ssl
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 1: Enable SSL/TLS on the Server**

Later in this tutorial you will work with three important files that have been installed:

- **/etc/httpd/conf.d/ssl.conf**

The configuration file for `mod_ssl`. It contains "directives" telling Apache where to find encryption keys and certificates, which SSL/TLS protocols to allow, and what encryption algorithms to use.

- **/etc/pki/tls/private/localhost.key**

An automatically generated, 2048-bit RSA private key for your Amazon EC2 host. During installation, OpenSSL used this key to generate a self-signed host certificate, and you can also use it later to generate a certificate signing request (CSR) to submit to a certificate authority (CA).

- **/etc/pki/tls/certs/localhost.crt**

An automatically generated, self-signed X.509 certificate for your server host. This certificate is useful for testing that Apache is properly set up to use SSL/TLS.

The `.key` and `.crt` files are both in PEM format, with consists of Base64-encoded ASCII characters framed by "BEGIN" and "END" lines, as in this abbreviated example of a certificate:

```
-----BEGIN CERTIFICATE-----  
  
MIIIEazCCA10gAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbExCzAJBgNVBAYTAi0t  
  
MRIwEAYDVQQIDAlTb211U3RhZGUxETAPBgNVBACMCFNvbWVDaXR5MRkwFwYDVQQK  
  
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYWxV  
  
bml0MRkwFwYDVQQDDDBpcC0xNzItMzEtMjAtMjMMSQwIgyYJKoZIhvcNAQkBFhVy  
  
...  
z5rRUE/XzxRLBZOoWZpNWTXJkQ3uFYH6s/  
sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
WanXWehT6FiSZvB4sTEXXJN2jdw8g  
+sHGnz8zCOsclknYhHrCVD2vnBLZJKSZvak  
3ZazhBxtQSukFMonWPP2a0DMMFGYUHOd0BQE8sBJxg==  
-----END CERTIFICATE-----
```

The file names and extensions are a convenience and have no effect on function; you can call a certificate either `cert.crt` or `cert.pem` or `certificate.pem`, so long as the related directive in the `ssl.conf` file uses the same name.

Note

When you replace the default SSL files with your own customized files, be sure that they are in PEM format.

4. Restart Apache.

```
[ec2-user ~]$ sudo service httpd restart
```

5. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by typing the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix `https://`. Because you are connecting to a site with a self-signed, untrusted certificate, your browser may display a series of warnings.

Override these and proceed to the site. If the default Apache welcome page opens, it means that you have successfully configured SSL/TLS on your server. All data passing between the browser and server is now safely encrypted, as signalled by the lock icon in the browser's URL bar.

To prevent site visitors from encountering warning screens, you need to obtain a certificate that not only encrypts, but also publicly authenticates you as the owner of the site.

Step 2: Obtain a CA-signed Certificate

This section describes the process of generating a certificate signing request (CSR) from a private key, submitting the CSR to a certificate authority (CA), obtaining a signed certificate, and configuring Apache to use it.

A self-signed SSL/TLS X.509 certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical; a CA promises to validate, at a minimum, a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to your private server key, and a signature by the CA that is cryptographically tied to the public key. When a browser connects to a web server over HTTPS, the server presents a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a chain of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A list of well-known CAs can be found at dmoz.org. A few CAs, such as StartCom, offer basic-level ("Class 1") certificates free of charge.

Underlying the certificate is the key. As of 2013, [government](#) and [industry](#) groups recommend using a minimum key (modulus) size of 2048 bits for RSA keys. The default modulus size generated by OpenSSL in Amazon Linux is 2048 bits, which means that the existing auto-generated key is suitable for use in a CA-signed certificate. An alternative procedure is described below for those who desire a customized key, for instance, one with a larger modulus or using a different encryption algorithm.

To obtain a CA-signed certificate

1. [Connect to your instance \(p. 28\)](#) and navigate to `/etc/pki/tls/private/`. This is the directory where the server's private key for SSL/TLS is stored. If you prefer to use your existing host key to generate the CSR, skip to Step 3.
2. (Optional) Generate a new private key. Here are some sample key configurations. Any of the resulting keys will work with your web server, but they vary in how (and how much) security they implement.

1. As a starting point, here is the command to create an RSA key resembling the default host key on your instance:

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 2048
```

The resulting file, `custom.key`, is a 2048-bit RSA private key.

2. To create a stronger RSA key with a bigger modulus, use the following command:

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

The resulting file, `custom.key`, is a 4096-bit RSA private key.

3. To create a 4096-bit encrypted RSA key with password protection, use the following command:

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out  
custom.key 4096
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 2: Obtain a CA-signed Certificate**

This results in a 4096-bit RSA private key that has been encrypted with the AES-128 cipher.

Important

Encryption provides greater security, but because an encrypted key requires a password, services depending on it cannot be auto-started. Each time you use this key, you need to supply the password "abcde12345" over an SSH connection.

4. RSA cryptography can be relatively slow, because its security relies on the difficulty of factoring the product of two large two prime numbers. However, it is possible to create keys for SSL/TLS that use non-RSA ciphers. Keys based on the mathematics of elliptic curves are smaller and faster when delivering an equivalent level of security. Here is an example:

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

The output in this case is a 256-bit elliptic curve private key using prime256v1, a "named curve" that OpenSSL supports. Its cryptographic strength is slightly greater than a 2048-bit RSA key, [according to NIST](#).

Note

Not all CAs provide the same level of support for elliptic-curve-based keys as for RSA keys.

Make sure that the new private key has highly restrictive permissions (owner root, group root, read/write for owner only). The commands would be as follows:

```
[ec2-user ~]$ sudo chown root.root custom.key  
[ec2-user ~]$ sudo chmod 600 custom.key  
[ec2-user ~]$ ls -al custom.key
```

The commands above should yield the following result:

```
-rw----- root root custom.key
```

After you have created and configured a satisfactory key, you can create a CSR.

3. Create a CSR using your preferred key; the example below uses `private.key`:

```
[ec2-user ~]$ sudo openssl req -new -key private.key -out csr.pem
```

OpenSSL opens a dialog and prompts you for information shown in the table below. All of the fields except **Common Name** are optional for a basic, domain-validated certificate.

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 2: Obtain a CA-signed Certificate**

Name	Description	Example
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corp
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to type into a browser. Usually, this means a domain name with a prefixed host name or alias in the form www.example.com . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the host name alone. CAs also offer more expensive certificates that accept wild-card names such as *.example.com .	www.example.com
Email Address	The server administrator's email address.	someone@example.com

Finally, OpenSSL prompts you for an optional challenge password. This password applies only to the CSR and to transactions between you and your CA, so follow the CA's recommendations about this and the other optional field, optional company name. The CSR challenge password has no effect on server operation.

The resulting file `csr.pem` contains your public key, your digital signature of your public key, and the metadata that you entered.

4. Submit the CSR to a CA. This usually consists of opening your CSR file in a text editor and copying the contents into a web form. At this time, you may be asked to supply one or more subject alternate names (SANs) to be placed on the certificate. If `www.example.com` is the common name, then `example.com` would be a good SAN, and vice versa. A user typing in either of the listed names would see an error-free connection. If your CA web form allows it, include the common name in the list of SANs. (Some CAs include it automatically.)

After your request has been approved, you will receive a new host certificate signed by the CA. You may also be instructed to download an *intermediate certificate* file that contains additional certificates needed to complete the CA's chain of trust.

5. Remove the old self-signed host certificate `localhost.crt` from the `/etc/pki/tls/certs` directory and place the new CA-signed certificate there (along with any intermediate certificates).

Note

There are several ways to upload your new certificate to your EC2 instance, but the most straightforward and informative way is to open a text editor (vi, nano, notepad, etc.) on both your local computer and your instance, and then copy and paste the file contents between them. This way, you will see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

From inside the `/etc/pki/tls/certs` directory, check that the file ownership, group, and permission settings match the highly restrictive Amazon Linux defaults (owner root, group root, read/write for owner only). The commands would be as follows:

```
[ec2-user certs]$ sudo chown root.root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 3: Test and Harden the Security Configuration**

The commands above should yield the following result:

```
-rw----- root root custom.crt
```

The permissions for the intermediate certificate file are less stringent (owner root, group root, owner can write, world can read). The commands would be:

```
[ec2-user certs]$ sudo chown root.root intermediate.crt  
[ec2-user certs]$ sudo chmod 644 intermediate.crt  
[ec2-user certs]$ ls -al intermediate.crt
```

The commands above should yield the following result:

```
-rw-r--r-- root root intermediate.crt
```

6. The file name of the new CA-signed certificate (`custom.crt` in this example) probably differs from the old certificate. Edit `/etc/httpd/conf.d/ssl.conf` and provide the correct path and file name using Apache's `SSLCertificateFile` directive.

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

If you received an intermediate certificate file (`intermediate.crt` in this example), provide its path and file name using Apache's `SSLCACertificateFile` directive.

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

7. Save `/etc/httpd/conf.d/ssl.conf` and restart Apache.

```
[ec2-user ~]$ sudo service httpd restart
```

Step 3: Test and Harden the Security Configuration

After your SSL/TLS is operational and exposed to the public, you should test how secure it really is. This is easy to do using online services such as [Qualys SSL Labs](#), which performs a free and thorough analysis of your security setup. Based on the results, you may decide to harden the default security configuration by controlling which protocols you accept, which ciphers you prefer, and which you exclude. For more information, see [how Qualys formulates its scores](#).

Important

Real-world testing is crucial to the security of your server. Small configuration errors may lead to serious security breaches and loss of data. Because recommended security practices change constantly in response to research and emerging threats, periodic security audits are essential to good server administration.

On the [Qualys SSL Labs](#) site, type the fully qualified domain name of your server, in the form `www.example.com`. After about two minutes, you will receive a grade (from A to F) for your site and a detailed breakdown of the findings. The table below summarizes the report for a domain with settings identical to the default Apache configuration on Amazon Linux:

Overall rating	C
----------------	---

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 3: Test and Harden the Security Configuration**

Certificate	100%
Protocol support	90%
Key exchange	90%
Cipher strength	90%

The report shows that the configuration is mostly sound, with acceptable ratings for certificate, protocol support, key exchange, and cipher strength issues. However, the report also flags three vulnerabilities that are responsible for lowering the overall grade and should be addressed:

- **POODLE vulnerability:** The [POODLE attack](#), discovered in 2014, exploits a weakness in SSL version 3 that allows an attacker to impersonate a web site. The fix is straightforward: Disable SSL version 3 support on the server. In the configuration file `/etc/httpd/conf.d/ssl.conf`, comment out the following by typing `"#"` at the beginning of the line:

```
SSLProtocol all -SSLv2
```

Then, add the following directive:

```
SSLProtocol -SSLv2 -SSLv3 +TLSv1 +TLSv1.1 +TLSv1.2
```

In addition to explicitly disabling SSL version 2, this command also disables SSL version 3 (the one flagged by the security audit) and explicitly allows all currently existing versions of TLS. The server will now refuse to accept encrypted connections with clients using anything except TLS. The verbose wording in the directive communicates more clearly, to a human reader, what the server is configured to do.

- **RC4 cipher support:** A cipher is the mathematical core of an encryption algorithm. RC4, a fast cipher used to encrypt SSL/TLS data-streams, is known to have several [serious weaknesses](#). The fix is to disable RC4 support in `ssl.conf`, which we'll do as part of the fix in the next case.
- **Missing support for forward secrecy:** [Forward secrecy](#) is a feature of protocols that encrypt using temporary (ephemeral) session keys derived from the private key. This means in practice that attackers cannot decrypt HTTPS data even if they possess a web server's long-term private key. The web browsers that make up Qualys's list of "reference browsers" all support forward secrecy.

The fix for both RC4 and forward secrecy issues is to customize Apache's list of permitted and forbidden ciphers, and to enforce a preference for strong ciphers over weaker ones. This requires two configuration changes.

In the configuration file `/etc/httpd/conf.d/ssl.conf`, find the section with commented-out examples for configuring `SSLCipherSuite`, comment out (but keep) the current list, and add the following directive:

Note

Though shown here on several lines for readability, the entire directive must be on a single line without spaces between the cipher names.

```
SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:  
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Step 3: Test and Harden the Security Configuration**

```
AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES:!
aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:
!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

These ciphers are a subset of the much longer list of supported ciphers in OpenSSL. They were selected and ordered according to the following criteria:

1. Support for forward secrecy
2. Strength
3. Speed
4. Specific ciphers before cipher families
5. Allowed ciphers before denied ciphers

Note that the high-ranking ciphers have *ECDHE* in their names, for *Elliptic Curve Diffie-Hellman Ephemeral*; the *ephemeral* indicates forward secrecy. Also, RC4 is now among the forbidden ciphers near the end.

We recommend that you use an explicit list of ciphers instead relying on defaults or terse directives whose content isn't visible.

Important

The cipher list shown here is just one of many possible lists; for instance, you might want to optimize a list for speed rather than forward secrecy.

If you anticipate a need to support older clients, you can allow the DES-CBC3-SHA cipher suite.

Finally, each update to OpenSSL introduces new ciphers and deprecates old ones. Keep your EC2 Amazon Linux instance up to date, watch for security announcements from [OpenSSL](#), and be alert to reports of new security exploits in the technical press. For more information, see [Predefined SSL Security Policies for Elastic Load Balancing](#) in the *Elastic Load Balancing User Guide*.

Uncomment the following line by removing the "#":

```
#SSLHonorCipherOrder on
```

This command forces the server to prefer high-ranking ciphers, including (in this case) those that support forward secrecy. With this directive turned on, the server tries to establish a strongly secure connection before falling back to allowed ciphers with lesser security.

Restart Apache after saving the edited configuration file.

If you test the domain again on [Qualys SSL Labs](#), you should see that the vulnerabilities are gone and the summary looks something like this:

Overall rating	A
Certificate	100%
Protocol support	95%
Key exchange	90%
Cipher strength	90%

Troubleshooting

- **My Apache webserver won't start unless I supply a password.**

This is expected behavior if you installed an encrypted, password-protected, private server key.

You can strip the key of its encryption and password. Assuming you have a private encrypted RSA key called `custom.key` in the default directory, and that the passphrase on it is `abcde12345`, run the following commands on your EC2 instance to generate an unencrypted version of this key:

```
[ec2-user ~]$ cd /etc/pki/tls/private/  
[ec2-user private]$ sudo cp custom.key custom.key.bak  
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345  
-out custom.key.nocrypt  
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key  
[ec2-user private]$ sudo chown root.root custom.key  
[ec2-user private]$ sudo chmod 600 custom.key  
[ec2-user private]$ sudo service httpd restart
```

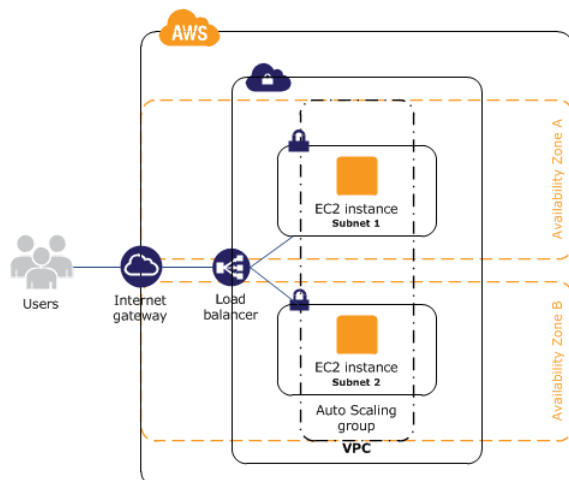
Apache should now start without prompting you for a password.

Tutorial: Increase the Availability of Your Application on Amazon EC2

Suppose that you start out running your app or website on a single EC2 instance, and over time, traffic increases to the point that you require more than one instance to meet the demand. You can launch multiple EC2 instances from your AMI and then use Elastic Load Balancing to distribute incoming traffic for your application across these EC2 instances. This increases the availability of your application. Placing your instances in multiple Availability Zones also improves the fault tolerance in your application. If one Availability Zone experiences an outage, traffic is routed to the other Availability Zone.

You can use Auto Scaling to maintain a minimum number of running instances for your application at all times. Auto Scaling can detect when your instance or application is unhealthy and replace it automatically to maintain the availability of your application. You can also use Auto Scaling to scale your Amazon EC2 capacity up or down automatically based on demand, using criteria that you specify.

In this tutorial, we use Auto Scaling with Elastic Load Balancing to ensure that you maintain a specified number of healthy EC2 instances behind your load balancer. Note that these instances do not need public IP addresses, because traffic goes to the load balancer and is then routed to the instances. For more information, see [Auto Scaling](#) and [Elastic Load Balancing](#).



Contents

- [Prerequisites](#) (p. 61)
- [Scale and Load Balance Your Application](#) (p. 61)
- [Test Your Load Balancer](#) (p. 63)

Prerequisites

This tutorial assumes that you have already done the following:

1. If you don't have a default virtual private cloud (VPC), create a VPC with one public subnet in two or more Availability Zones. For more information, see [Create a Virtual Private Cloud \(VPC\)](#) (p. 23).
2. Launch an instance in the VPC.
3. Connect to the instance and customize it. For example, you can install software and applications, copy data, and attach additional EBS volumes. For information about setting up a web server on your instance, see [Tutorial: Installing a LAMP Web Server on Amazon Linux](#) (p. 33).
4. Test your application on your instance to ensure that your instance is configured correctly.
5. Create a custom Amazon Machine Image (AMI) from your instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89) or [Creating an Instance Store-Backed Linux AMI](#) (p. 92).
6. (Optional) Terminate the instance if you no longer need it.
7. Create an IAM role that grants your application the access to AWS that it needs. For more information, see [Creating an IAM Role Using the Console](#) (p. 538).

Scale and Load Balance Your Application

Use the following procedure to create a load balancer, create a launch configuration for your instances, create an Auto Scaling group with two or more instances, and associate the load balancer with the Auto Scaling group.

To scale and load-balance your application

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **LOAD BALANCING**, click **Load Balancers**.
3. Click **Create Load Balancer**.
4. On the **Define Load Balancer** page, do the following:

- a. Enter a name for the load balancer in **Load Balancer name**. For example, `my-1b`.
- b. Select your VPC from **Create LB Inside**.
- c. Leave the default for **Listener Configuration**, which is an HTTP listener.
- d. If you selected a nondefault VPC, you are prompted to select subnets under **Select Subnets**. Click the icon in the **Action** column for each of your two public subnets.
- e. Click **Next: Assign Security Groups**.
5. On the **Assign Security Groups** page, do the following:
 - a. Click **Create a new security group**.
 - b. Enter a name and description for your security group. This new security group contains a rule that allows traffic to the port for your HTTP listener.
 - c. Click **Next: Configure Security Settings**.
6. On the **Configure Security Settings** page, click **Next: Configure Health Check** to continue to the next page, as we are not creating an HTTPS load balancer.
7. On the **Configure Health Check** page, do the following:
 - a. Leave **Ping Protocol** and **Ping Port** at their default values.
 - b. Set **Ping Path** to `/`. This sends queries to your default page, whether it is named `index.html` or something else.
 - c. Click **Next: Add EC2 Instances**.
8. On the **Add EC2 Instances** page, click **Next: Add Tags** to continue to the next page, as we'll use Auto Scaling to add EC2 instances to the load balancer.
9. On the **Add Tags** page, add any tags that you need, and then click **Review and Create**.
10. On the **Review** page, review your settings and then click **Create**. After the load balancer is created, click **Close**.
11. In the navigation pane, under **AUTO SCALING**, click **Launch Configurations**.
 - If you are new to Auto Scaling, you see a welcome page. Click **Create Auto Scaling group** to start the Create Auto Scaling Group wizard, and then click **Create launch configuration**.
 - Otherwise, click **Create launch configuration**.
12. On the **Choose AMI** page, select the **My AMIs** tab, and then select the AMI that you created in [Prerequisites \(p. 61\)](#).
13. On the **Choose Instance Type** page, select an instance type, and then click **Next: Configure details**.
14. On the **Configure details** page, do the following:
 - a. In **Name**, enter a name for your launch configuration (for example, `my-launch-config`).
 - b. From **IAM role**, select the IAM role that you created in [Prerequisites \(p. 61\)](#).
 - c. (Optional) If you need to run a startup script, expand **Advanced Details** and enter the script in **User data**.
 - d. Click **Skip to review**.
15. On the **Review** page, click **Edit security groups**. You can select an existing security group or create a new one. This security group must allow HTTP traffic and health checks from the load balancer. If your instances will have public IP addresses, you can optionally allow SSH traffic if you need to connect to the instances. When you are finished, click **Review**.
16. On the **Review** page, click **Create launch configuration**.
17. When prompted, select an existing key pair, create a new key pair, or proceed without a key pair. Click the acknowledgment check box, and then click **Create launch configuration**.
18. After the launch configuration is created, you must create an Auto Scaling group.
 - If you are new to Auto Scaling and you are using the Create Auto Scaling group wizard, you are taken to the next step automatically.

- Otherwise, click **Create an Auto Scaling group using this launch configuration**.
19. On the **Configure Auto Scaling group details** page, do the following:
 - a. Enter a name for the Auto Scaling group. For example, `my-asg`.
 - b. In **Group size**, enter the number of instances in the text box (for example, 2). Note that we recommend that you maintain approximately the same number of instances in each Availability Zone.
 - c. Select your VPC from **Network** and your two public subnets from **Subnet**.
 - d. Expand **Advanced Details**. Select **Receive traffic from Elastic Load Balancer(s)**. Select your load balancer from the text field.
 - e. Click **Next: Configure scaling policies**.
 20. On the **Configure scaling policies** page, click **Review**, as we will simply let Auto Scaling maintain the group at the specified size. Note that later on, you can manually scale this Auto Scaling group, configure the group to scale on a schedule, or configure the group to scale based on demand.
 21. On the **Review** page, click **Create Auto Scaling group**.
 22. After the group is created, click **Close**.

Test Your Load Balancer

First, verify that Auto Scaling has launched your instances and that they are ready. From the **Auto Scaling Groups** page, select your Auto Scaling group and then select the **Instances** tab.

Details	Activity History	Scaling Policies	Instances	Notifications	Tags	Scheduled Actions
Actions ▾						
Filter: Any Health Status ▾ Any Lifecycle State ▾ <input type="text" value="Filter instances..."/>						
<input type="checkbox"/>	Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status	
	i-07ffe74c7330ebf53	Pending	my-launch-config	us-west-2a	Healthy	
	i-0b263919b6498b123	Pending	my-launch-config	us-west-2c	Healthy	

Initially, your instances are in the `Pending` state. When their states are `InService`, they are ready for use.

Next, verify that your instances are registered with the load balancer and ready to receive traffic from the load balancer. From the **Load Balancers** page, select your load balancer and then select the **Instances** tab.

Load balancer: **my-lb**

Description **Instances** Health Check Monitoring Security Listeners Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-07fe74c7330ebf53		us-west-2a	OutOfService ⓘ	Remove from Load Balancing
i-0b263919b6498b123		us-west-2c	OutOfService ⓘ	Remove from Load Balancing

If the state of your instances is `OutOfService`, it's possible that they are still registering. When their states are `InService`, they are ready for use. After your instances are ready, you can test your load balancer as follows.

To test your load balancer

1. From the **Load Balancers** page, select your load balancer.
2. On the **Description** tab, locate the DNS name. This name has the following form:

```
my-lb-xxxxxxxxxx.us-west-2.elb.amazonaws.com
```

3. In a web browser, paste the DNS name for the load balancer into the address bar and press Enter. You'll see your website displayed.

Tutorial: Remotely Manage Your Amazon EC2 Instances

This tutorial shows you how to remotely manage an Amazon EC2 instance using Amazon Elastic Compute Cloud (Amazon EC2) Run Command from your local machine. In this tutorial, you will learn how to do the following tasks:

- Launch a new instance that is configured for Run Command.
- Configure your user account for Run Command.
- Use Run Command to send a command from your local machine and retrieve a list of services running on the instance.

This tutorial includes procedures for executing commands using either the Amazon EC2 console or AWS Command Line Interface.

Note

With Run Command, you can also manage your servers and virtual machines (VMs) in your on-premises environment or in an environment provided by other cloud providers. For more information, see [Setting Up Run Command On Managed Instances \(p. 371\)](#).

Launch a New Instance

Instances require an AWS Identity and Access Management (IAM) role that enables the instance to communicate with Amazon EC2 Simple Systems Manager (SSM). You must assign the IAM role when

you create the new instance. You can't assign a role to an instance that is already running. For existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Linux Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.
6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Run Command role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
8. From **IAM role**, choose the role you just created.
9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this tutorial.

Grant Your User Account Access to SSM

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed IAM policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type `AmazonSSMFullAccess` and press Enter.
4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

Install the SSM Agent

The SSM agent processes Run Command requests and configures the instances that are specified in the request. You must manually install the agent using the procedure for your version of Linux. The following procedure describes how to install the agent on Red Hat Enterprise Linux (RHEL). For information about how to install the agent on Ubuntu, Amazon Linux or CentOS, see [Manually Install the SSM Agent on EC2 Instances \(p. 359\)](#).

To install the SSM agent on Red Hat Enterprise Linux

1. Connect to your RHEL instance and create a temporary directory on the instance.

Amazon Elastic Compute Cloud User Guide for Linux Instances Install the SSM Agent

```
mkdir /tmp/ssm
```

2. Use one of the following commands to download the SSM installer to the temporary directory. Replace *region* with one of the AWS Regions where SSM is available. To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance.

64-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

32-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_386/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

3. Run the SSM installer.

```
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
```

4. Run one of the following commands to determine if the SSM agent is running. The command should return "amazon-ssm-agent is running."

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x

```
sudo status amazon-ssm-agent
```

5. Execute the following commands if the previous command returned "amazon-ssm-agent is stopped."

- a. Start the service.

RHEL 7.x

```
sudo systemctl start amazon-ssm-agent
```

RHEL 6.x

```
sudo start amazon-ssm-agent
```

- b. Check the status of the agent.

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x

```
sudo status amazon-ssm-agent
```

Send a Command Using the EC2 Console

Use the following procedure to list all services running on the instance by using Run Command from the Amazon EC2 console.

1. In the [Amazon EC2 console](#) choose **Command History** in the navigation pane, and then choose **Run a command**.
2. In the **Command document** list, choose **AWS-RunShellScript**.
3. Choose **Select instances**, and then choose the instance you just created. If you do not see the instance, verify that you are currently in the same region as the instance you created. Also verify that you configured the IAM role and trust policies correctly as described earlier in this topic.
4. In the **Commands** field, type `service --status-all`. You can specify a **Working Directory** and **Execution Timeout**, if you want. The **Execution Timeout** is the number of seconds the SSM agent will attempt to run the command before it is considered to have failed. We recommend entering a comment in the **Comments** field. A comment will help you identify the command in the list of pending commands and make it easier to view the output.
5. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails.
6. Choose **Run** to execute the command simultaneously on the selected instances. Run Command displays a status screen.
7. Choose **View result**.
8. Choose the command invocation for the command you just ran.
9. Choose the **Output** tab, and then choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
a0f0eb16-5073-482...	i-35da6583	AWS-RunShellScript	Success	December 8, ...	Checking ser
a0f0eb16-5073-482...	i-36da6580	AWS-RunShellScript	Success	December 8, ...	Checking ser
a0f0eb16-5073-482...	i-37da6581	AWS-RunShellScript	Success	December 8, ...	Checking ser

Command ID: a0f0eb16-5073-482b-b875-0c8ae9038cbf Instance ID: i-35da6583

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time
aws:runShellS...	Success	0	December 8, 2015 at 5:26:40 P...	December 8, 2015 at 5:26:40 P...

10. The system displays the output in your browser. If the output is longer than 2500 characters, only the first 2500 characters are shown and the rest is truncated. You could view the full output by specifying an Amazon S3 bucket before executing the command.

Send a Command Using the AWS CLI

Use the following procedure to list all services running on the instance by using Run Command in the AWS CLI.

To execute a command

1. On your local computer, download the latest version of the [AWS Command Line Interface](#) (AWS CLI).
2. Open the AWS CLI on your local computer and execute the following command to specify your credentials and the region.

```
aws configure
```

3. The system prompts you to specify the following.

```
AWS Access Key ID [None]: key  
AWS Secret Access Key [None]: key  
Default region name [None]: region, for example us-east-1  
Default output format [None]: ENTER
```

4. Execute the following command to retrieve the services running on the instance.

```
aws ssm send-command --document-name "AWS-RunShellScript" --comment  
"listing services" --instance-ids "Instance-ID" --parameters  
commands="service --status-all" --region us-west-2 --output text
```

The command returns a command ID, which you will use to view the results.

5. The following command returns the output of the original Send-SSMCommand. The output is truncated after 2500 characters. To view the full list of services, you would need to specify an Amazon S3 bucket in the command using the `--output-s3-bucket-name bucket_name` parameter.

```
aws ssm list-command-invocations --command-id "command ID" --details
```

For more examples of how to execute commands using Run Command with the AWS CLI and the AWS Management Console, see [Amazon EC2 Run Command Walkthroughs](#) (p. 387). For more information about Run Command, see [Remotely Manage Your Instances](#) (p. 352).

For videos, see [AWS Instructional Videos and Labs](#).

Amazon Machine Images (AMI)

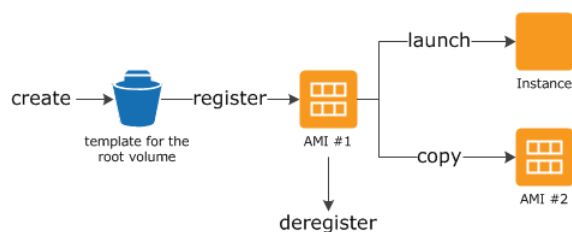
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI to the same region or to different regions. When you are finished launching instance from an AMI, you can deregister the AMI.



You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see [AMI Types \(p. 71\)](#) and [Finding a Linux AMI \(p. 75\)](#).

When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see [Amazon EC2 Instances](#) (p. 156).

Creating Your Own AMI

You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume. For information, see [Amazon EC2 Root Device Volume](#) (p. 14).

To create an Amazon EBS-backed AMI, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89).
To create an instance store-backed AMI, see [Creating an Instance Store-Backed Linux AMI](#) (p. 92).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging Your Amazon EC2 Resources](#) (p. 751).

Buying, Sharing, and Selling AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see [Shared AMIs](#) (p. 76).

You can purchase an AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see [Paid AMIs](#) (p. 85).

Deregistering Your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, you can't use it to launch new instances. For more information, see [Deregistering Your AMI](#) (p. 140).

Amazon Linux

The Amazon Linux AMI is a supported and maintained Linux image provided by AWS. The following are some of the features of Amazon Linux:

- A stable, secure, and high-performance execution environment for applications running on Amazon EC2.
- Provided at no additional charge to Amazon EC2 users.
- Repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, Tomcat, and many more common packages.
- Updated on a regular basis to include the latest components, and these updates are also made available in the **yum** repositories for installation on running instances.

- Includes packages that enable easy integration with AWS services, such as the AWS CLI, Amazon EC2 API and AMI tools, the Boto library for Python, and the Elastic Load Balancing tools.

For more information, see [Amazon Linux \(p. 142\)](#).

AMI Types

You can select an AMI to use based on the following characteristics:

- Region (see [Regions and Availability Zones \(p. 7\)](#))
- Operating system
- Architecture (32-bit or 64-bit)
- [Launch Permissions \(p. 71\)](#)
- [Storage for the Root Device \(p. 71\)](#)

Launch Permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see [Shared AMIs \(p. 76\)](#). Developers can charge for their AMIs. For more information, see [Paid AMIs \(p. 85\)](#).

Storage for the Root Device

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. For more information, see [Amazon EC2 Root Device Volume \(p. 14\)](#).

This section summarizes the important differences between the two types of AMIs. The following table provides a quick summary of these differences.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Storage for the Root Device**

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.	Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

* By default, Amazon EBS-backed instance root volumes have the `DeleteOnTermination` flag set to `true`. For information about how to change this flag so that the volume persists after termination, see [Changing the Root Device Volume to Persist](#) (p. 17).

Determining the Root Device Type of Your AMI

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed AMI.
 - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- `describe-images` (AWS CLI)
- `Get-EC2Image` (AWS Tools for Windows PowerShell)

Stopped State

You can stop an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from `running` to `stopping`

to `stopped`). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance store-backed AMIs can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see [Stop and Start Your Instance](#) (p. 299).

Default Data Storage and Persistence

Instances that use an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). Any data on an instance store volume is deleted when the instance fails or terminates (except for data on the root device). You can add persistent storage to your instance by attaching one or more Amazon EBS volumes.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. The instances don't use any available instance store volumes by default. You can add instance storage or additional Amazon EBS volumes using a block device mapping. For more information, see [Block Device Mapping](#) (p. 731). For information about what happens to the instance store volumes when you stop an instance, see [Stop and Start Your Instance](#) (p. 299).

Boot Times

Amazon EBS-backed AMIs launch faster than Amazon EC2 instance store-backed AMIs. When you launch an Amazon EC2 instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available. However, the performance of an instance that uses an Amazon EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume. When you stop and restart the instance, it launches quickly, because the state is stored in an Amazon EBS volume.

AMI Creation

To create Linux AMIs backed by instance store, you must create an AMI from your instance on the instance itself using the Amazon EC2 AMI tools.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89).

How You're Charged

With AMIs backed by instance store, you're charged for AMI storage and instance usage. With AMIs backed by Amazon EBS, you're charged for volume storage and usage in addition to the AMI and instance usage charges.

With Amazon EC2 instance store-backed AMIs, each time you customize an AMI and create a new one, all of the parts are stored in Amazon S3 for each AMI. So, the storage footprint for each customized AMI is the full size of the AMI. For Amazon EBS-backed AMIs, each time you customize an AMI and create a new one, only the changes are stored. So the storage footprint for subsequent AMIs you customize after the first is much smaller, resulting in lower AMI storage charges.

When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage. We charge a full instance hour for every transition from a stopped state to a running state, even if you transition the instance multiple times within a single hour. For example, let's say the hourly instance charge for your instance is \$0.10. If you were to run that instance for one hour without stopping it, you would be charged \$0.10. If you stopped and restarted

that instance twice during that hour, you would be charged \$0.30 for that hour of usage (the initial \$0.10, plus 2 x \$0.10 for each restart).

Linux AMI Virtualization Types

Linux Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main difference between PV and HVM AMIs is the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch your instances. For more information about current generation instance types, see the [Amazon EC2 Instances](#) detail page. If you are using previous generation instance types and would like to upgrade, see [Upgrade Paths](#).

For information about the types of the Amazon Linux AMI recommended for each instance type, see the [Amazon Linux AMI Instance Types](#) detail page.

HVM AMIs

HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.

Unlike PV guests, HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. For more information on CPU virtualization extensions available in Amazon EC2, see [Intel Virtualization Technology](#) on the Intel website. HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access. For more information, see [Enhanced Networking \(p. 608\)](#) and [Linux GPU Instances \(p. 166\)](#).

All current generation instance types support HVM AMIs. The CC2, CR1, HI1, and HS1 previous generation instance types support HVM AMIs.

To find an HVM AMI, verify that the virtualization type of the AMI is set to `hvm`, using the console or the [describe-images](#) command.

PV AMIs

PV AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the `menu.lst` file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization, but they cannot take advantage of special hardware extensions such as enhanced networking or GPU processing. Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true. For more information about PV-GRUB and its use in Amazon EC2, see [User Provided Kernels \(p. 149\)](#).

The C3 and M3 current generation instance types support PV AMIs. The C1, HI1, HS1, M1, M2, and T1 previous generation instance types support PV AMIs.

To find a PV AMI, verify that the virtualization type of the AMI is set to `paravirtual`, using the console or the [describe-images](#) command.

PV on HVM

Paravirtual guests traditionally performed better with storage and network operations than HVM guests because they could leverage special drivers for I/O that avoided the overhead of emulating network

and disk hardware, whereas HVM guests had to translate these instructions to emulated hardware. Now these PV drivers are available for HVM guests, so operating systems that cannot be ported to run in a paravirtualized environment (such as Windows) can still see performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same, or better, performance than paravirtual guests.

Finding a Linux AMI

Before you can launch an instance, you must select an AMI to use. As you select an AMI, consider the following requirements you might have for the instances that you'll launch:

- The region
- The operating system
- The architecture: 32-bit (i386) or 64-bit (x86_64)
- The root device type: Amazon EBS or instance store
- The provider: Amazon Web Services, Oracle, IBM, Microsoft, or the community

If you need to find a Windows AMI, see [Finding a Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [Finding a Linux AMI Using the Amazon EC2 Console](#) (p. 75)
- [Finding an AMI Using the AWS CLI](#) (p. 76)

Finding a Linux AMI Using the Amazon EC2 Console

You can find Linux AMIs using the Amazon EC2 console. You can search through all available AMIs using the **Images** page, or select from commonly used AMIs on the **Quick Launch** tab when you use the console to launch an instance.

To find a Linux AMI using the Images page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region. You can select any region that's available to you, regardless of your location. This is the region in which you'll launch your instance.
3. In the navigation pane, choose **AMIs**.
4. (Optional) Use the **Filter** options to scope the list of displayed AMIs to see only the AMIs that interest you. For example, to list all Linux AMIs provided by AWS, select **Public images**. Choose the Search bar and select **Owner** from the menu, then select **Amazon images**. Choose the Search bar again to select **Platform** and then the operating system from the list provided.
5. (Optional) Choose the **Show/Hide Columns** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties in the **Details** tab.
6. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see [Storage for the Root Device](#) (p. 71).
7. To launch an instance from this AMI, select it and then choose **Launch**. For more information about launching an instance using the console, see [Launching Your Instance from an AMI](#) (p. 281). If you're not ready to launch the instance now, write down the AMI ID (ami-xxxxxxx) for later.

To find a Linux AMI when you launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, on the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you need, select the **AWS Marketplace** or **Community AMIs** tab to find additional AMIs.

Finding an AMI Using the AWS CLI

You can use command line parameters to list only the types of AMIs that interest you. For example, you can use the `describe-images` command as follows to find public AMIs owned by you or Amazon.

```
$ aws ec2 describe-images --owners self amazon
```

Add the following filter to the previous command to display only AMIs backed by Amazon EBS:

```
--filters "Name=root-device-type,Values=ebs"
```

After locating an AMI that meets your needs, write down its ID (ami-xxxxxxx). You can use this AMI to launch instances. For more information, see [Launching an Instance Using the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Shared AMIs

A *shared AMI* is an AMI that a developer created and made available for other developers to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content. You can also create your own AMIs and share them with others.

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence. We recommend that you get an AMI from a trusted source. If you have questions or observations about a shared AMI, use the [AWS forums](#).

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

For information about creating an AMI, see [Creating an Instance Store-Backed Linux AMI](#) or [Creating an Amazon EBS-Backed Linux AMI](#). For more information about building, delivering, and maintaining your applications on the AWS Marketplace, see the [AWS Marketplace User Guide](#) and [AWS Marketplace Seller Guide](#).

Contents

- [Finding Shared AMIs \(p. 77\)](#)
- [Making an AMI Public \(p. 78\)](#)
- [Sharing an AMI with Specific AWS Accounts \(p. 80\)](#)
- [Using Bookmarks \(p. 81\)](#)
- [Guidelines for Shared Linux AMIs \(p. 81\)](#)

Finding Shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

Finding a Shared AMI (Console)

To find a shared private AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Private images**. All AMIs that have been shared with you are listed. To granulate your search, choose the Search bar and use the filter options provided in the menu.

To find a shared public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Public images**. To granulate your search, choose the Search bar and use the filter options provided in the menu.
4. Use filters to list only the types of AMIs that interest you. For example, choose **Owner :** and then choose **Amazon images** to display only Amazon's public images.

Finding a Shared AMI (Command Line)

To find a shared public AMI using the command line tools

Use the `describe-images` command (AWS CLI) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the `--executable-users` option. This list includes any public AMIs that you own.

```
$ aws ec2 describe-images --executable-users all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
$ aws ec2 describe-images --executable-users self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
$ aws ec2 describe-images --owners amazon
```

The following command lists the AMIs owned by the specified AWS account.

```
$ aws ec2 describe-images --owners 123456789012
```

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

```
--filters "Name=root-device-type,Values=efs"
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Get-EC2Image](#).

Using Shared AMIs

Before you use a shared AMI, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party. Check the documentation for the Linux distribution used by the AMI for information about improving the security of the system.

To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.

1. Identify and disable any unauthorized public SSH keys. The only key in the file should be the key you used to launch the AMI. The following command locates `authorized_keys` files:

```
$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. Disable password-based authentication for the root user. Open the `ssh_config` file and edit the `PermitRootLogin` line as follows:

```
PermitRootLogin without-password
```

Alternatively, you can disable the ability to log into the instance as root:

```
PermitRootLogin No
```

Restart the `sshd` service.

3. Check whether there are any other user accounts that are able to log in to your instance. Accounts with superuser privileges are particularly dangerous. Remove or lock the password of any unknown accounts.
4. Check for open ports that you aren't using and running network services listening for incoming connections.
5. To prevent preconfigured remote logging, you should delete the existing configuration file and restart the `rsyslog` service. For example:

```
$ sudo rm /etc/  
rsyslog.config  
$ sudo service rsyslog restart
```

6. Verify that all cron jobs are legitimate.

If you discover a public AMI that you feel presents a security risk, contact the AWS security team. For more information, see the [AWS Security Center](#).

Making an AMI Public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the

AMI (see [Sharing an AMI with Specific AWS Accounts \(p. 80\)](#)). You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it. For more information, see [Copying an AMI \(p. 135\)](#).

To avoid exposing sensitive data when you share an AMI, read the security considerations in [Guidelines for Shared Linux AMIs \(p. 81\)](#) and follow the recommended actions.

Note

If an AMI has a product code, you can't make it public. You must share the AMI with only specific AWS accounts.

Sharing an AMI with all AWS Accounts (Console)

After you make an AMI public, it is available in **Community AMIs** when you launch an instance in the same region using the console. Note that it can take a short while for an AMI to appear in **Community AMIs** after you make it public. It can also take a short while for an AMI to be removed from **Community AMIs** after you make it private again.

To share a public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI from the list, and then choose **Actions, Modify Image Permissions**.
4. Choose **Public** and choose **Save**.

Sharing an AMI with all AWS Accounts (Command Line)

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts) or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the `modify-image-attribute` command (AWS CLI) as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission
  {"Add\":[{"Group\":"all\"}]}"
```

To verify the launch permissions of the AMI, use the following `describe-image-attribute` command.

```
$ aws ec2 describe-image-attribute --image-id ami-12345678 --attribute
  launchPermission
```

(Optional) To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission  
"{\"Remove\": [{\"Group\": \"all\"}]}"
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands: [Edit-EC2ImageAttribute](#) and [Get-EC2ImageAttribute](#).

Sharing an AMI with Specific AWS Accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need are the AWS account IDs.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it. For more information, see [Copying an AMI](#) (p. 135).

Sharing an AMI (Console)

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, Modify Image Permissions**.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then choose **Add Permission**.

To share this AMI with multiple users, repeat the above step until you have added all the required users.

5. To allow create volume permissions for snapshots, select **Add "create volume" permissions to the following associated snapshots when creating permissions**.

Note

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch.

6. Choose **Save** when you are done.

Sharing an AMI (Command Line)

Use the [modify-image-attribute](#) command (AWS CLI) to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission  
"{\"Add\": [{\"UserId\": \"123456789012\"}]}"
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
$ aws ec2 modify-image-attribute --image-id ami-12345678 --launch-permission  
"{\"Remove\": [{\"UserId\": \"123456789012\"}]}"
```


To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ aws ec2 reset-image-attribute --image-id ami-12345678 --attribute  
launchPermission
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Edit-EC2ImageAttribute](#).

Using Bookmarks

If you have created a public AMI, or shared an AMI with another AWS user, you can create a *bookmark* that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.

Note that your AMI must be public, or you must have shared it with the user to whom you want to send the bookmark.

To create a bookmark for your AMI

1. Type a URL with the following information, where *<region>* is the region in which your AMI resides, and *<ami_id>* is the ID of the AMI:

```
https://console.aws.amazon.com/ec2/v2/home?  
region=<region>#LaunchInstanceWizard:ami=<ami_id>
```

For example, this URL launches an instance from the ami-12345678 AMI in the us-east-1 region:

```
https://console.aws.amazon.com/ec2/v2/home?region=us-  
east-1#LaunchInstanceWizard:ami=ami-12345678
```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, choose the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Guidelines for Shared Linux AMIs

Use the following guidelines to reduce the attack surface and improve the reliability of the AMIs you create.

Note

No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.

Topics

- [Update the AMI Tools at Boot Time \(p. 82\)](#)
- [Disable Password-Based Remote Logins for Root \(p. 82\)](#)
- [Disable Local Root Access \(p. 83\)](#)
- [Remove SSH Host Key Pairs \(p. 83\)](#)
- [Install Public Key Credentials \(p. 83\)](#)

- [Disabling sshd DNS Checks \(Optional\) \(p. 84\)](#)
- [Identify Yourself \(p. 84\)](#)
- [Protect Yourself \(p. 85\)](#)

If you are building AMIs for AWS Marketplace, see [Building AMIs for AWS Marketplace](#) for guidelines, policies and best practices.

For additional information about sharing AMIs safely, see the following articles:

- [How To Share and Use Public AMIs in A Secure Manner](#)
- [Public AMI Publishing: Hardening and Clean-up Requirements](#)

Update the AMI Tools at Boot Time

For AMIs backed by instance store, we recommend that your AMIs download and upgrade the Amazon EC2 AMI creation tools during startup. This ensures that new AMIs based on your shared AMIs have the latest AMI tools.

For [Amazon Linux](#), add the following to `/etc/rc.local`:

```
# Update the Amazon EC2 AMI tools
echo " + Updating EC2 AMI tools"
yum update -y aws-amitools-ec2
echo " + Updated EC2 AMI tools"
```

Use this method to automatically update other software on your image.

Note

When deciding which software to automatically update, consider the amount of WAN traffic that the update will generate (your users will be charged for it) and the risk of the update breaking other software on the AMI.

For other distributions, make sure you have the latest AMI tools.

Disable Password-Based Remote Logins for Root

Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse.

To solve this problem, disable password-based remote logins for the root user.

To disable password-based remote logins for root

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#PermitRootLogin yes
```

2. Change the line to:

```
PermitRootLogin without-password
```

The location of this configuration file might differ for your distribution, or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Disable Local Root Access

When you work with shared AMIs, a best practice is to disable direct root logins. To do this, log into your running instance and issue the following command:

```
[ec2-user ~]$ sudo passwd -l root
```

Note

This command does not impact the use of `sudo`.

Remove SSH Host Key Pairs

If you plan to share an AMI derived from a public AMI, remove the existing SSH host key pairs located in `/etc/ssh`. This forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of "man-in-the-middle" attacks.

Remove all of the following key files that are present on your system.

- `ssh_host_dsa_key`
- `ssh_host_dsa_key.pub`
- `ssh_host_key`
- `ssh_host_key.pub`
- `ssh_host_rsa_key`
- `ssh_host_rsa_key.pub`
- `ssh_host_ecdsa_key`
- `ssh_host_ecdsa_key.pub`
- `ssh_host_ed25519_key`
- `ssh_host_ed25519_key.pub`

You can securely remove all of these files with the following command.

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

Warning

Secure deletion utilities such as `shred` may not remove all copies of a file from your storage media. Hidden copies of files may be created by journaling file systems (including Amazon Linux default ext4), snapshots, backups, RAID, and temporary caching. For more information see the [shred documentation](#).

Important

If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.

Install Public Key Credentials

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

Amazon EC2 allows users to specify a public-private key pair name when launching an instance. When a valid key pair name is provided to the `RunInstances` API call (or through the command line API tools), the public key (the portion of the key pair that Amazon EC2 retains on the server after a call to `CreateKeyPair` or `ImportKeyPair`) is made available to the instance through an HTTP query against the instance metadata.

To log in through SSH, your AMI must retrieve the key value at boot and append it to `/root/.ssh/authorized_keys` (or the equivalent for any other user account on the AMI). Users can launch instances of your AMI with a key pair and log in without requiring a root password.

Many distributions, including Amazon Linux and Ubuntu, use the `cloud-init` package to inject public key credentials for a configured user. If your distribution does not support `cloud-init`, you can add the following code to a system start-up script (such as `/etc/rc.local`) to pull in the public key you specified at launch for the `root` user.

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /
tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

This can be applied to any user account; you do not need to restrict it to `root`.

Note

Rebundling an instance based on this AMI includes the key with which it was launched. To prevent the key's inclusion, you must clear out (or delete) the `authorized_keys` file or exclude this file from rebundling.

Disabling sshd DNS Checks (Optional)

Disabling `sshd` DNS checks slightly weakens your `sshd` security. However, if DNS resolution fails, SSH logins still work. If you do not disable `sshd` checks, DNS resolution failures prevent all logins.

To disable sshd DNS checks

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#UseDNS yes
```

2. Change the line to:

```
UseDNS no
```

Note

The location of this configuration file can differ for your distribution or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Identify Yourself

Currently, there is no easy way to know who provided a shared AMI, because each AMI is represented by an account ID.

We recommend that you post a description of your AMI, and the AMI ID, in the [Amazon EC2 forum](#). This provides a convenient central location for users who are interested in trying new shared AMIs. You can also post the AMI to the [Amazon Machine Images \(AMIs\)](#) page.

Protect Yourself

The previous sections described how to make your shared AMIs safe, secure, and usable for the users who launch them. This section describes guidelines to protect yourself from the users of your AMI.

We recommend against storing sensitive data or software on any AMI that you share. Users who launch a shared AMI might be able to rebundle it and register it as their own. Follow these guidelines to help you to avoid some easily overlooked security risks:

- We recommend using the `--exclude directory` option on `ec2-bundle-vol` to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. In particular, exclude all user-owned SSH public/private key pairs and SSH `authorized_keys` files when bundling the image. The Amazon public AMIs store these in `/root/.ssh` for the `root` account, and `/home/user_name/.ssh/` for regular user accounts. For more information, see [ec2-bundle-vol](#) (p. 99).
- Always delete the shell history before bundling. If you attempt more than one bundle upload in the same AMI, the shell history contains your secret access key. The following example should be the last command executed before bundling from within the instance.

```
[ec2-user ~]$ shred -u ~/.*history
```

Warning

The limitations of `shred` described in the warning above apply here as well.

Be aware that `bash` writes the history of the current session to the disk on exit. If you log out of your instance after deleting `~/.bash_history`, and then log back in, you will find that `~/.bash_history` has been re-created and contains all of the commands executed during your previous session.

Other programs besides `bash` also write histories to disk. Use caution and remove or exclude unnecessary dot-files and dot-directories.

- Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (such as the instance store).

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS; including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the [AWS Marketplace](#) site.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services; for example, the hourly rate for running a `m1.small` instance type in Amazon EC2. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Important

Amazon DevPay is no longer accepting new sellers or products. AWS Marketplace is now the single, unified e-commerce platform for selling software and services through AWS. For information about how to deploy and sell software from AWS Marketplace, see [Selling on AWS Marketplace](#). AWS Marketplace supports AMIs backed by Amazon EBS.

Topics

- [Selling Your AMI](#) (p. 86)
- [Finding a Paid AMI](#) (p. 86)
- [Purchase a Paid AMI](#) (p. 87)
- [Getting the Product Code for Your Instance](#) (p. 87)
- [Using Paid Support](#) (p. 88)
- [Bills for Paid and Supported AMIs](#) (p. 88)
- [Managing Your AWS Marketplace Subscriptions](#) (p. 88)

Selling Your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot instances.

For information about how to sell your AMI on AWS Marketplace, see [Selling on AWS Marketplace](#).

Finding a Paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use [AWS Marketplace](#), the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a Paid AMI Using the Console

To find a paid AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **Public images** from the first **Filter** list. Click the Search bar and select **Product Code**, then **Marketplace**. Click the Search bar again, select **Platform** and then choose the operating system from the list.

Finding a Paid AMI Using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open [AWS Marketplace](#).
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either `AMI` or `Software as a Service`.

Finding a Paid AMI Using the Command Line

You can find a paid AMI using the `describe-images` command (AWS CLI) as follows.

```
$ aws ec2 describe-images --owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [  
  {  
    "ProductCodeId": "product_code",  
    "ProductCodeType": "marketplace"  
  }  
],
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Get-EC2Image](#).

Purchase a Paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a Paid AMI Using the Console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see [Launching an AWS Marketplace Instance \(p. 287\)](#).

Subscribing to a Product Using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website:** You can launch preconfigured software quickly with the 1-Click deployment feature.
- **Amazon EC2 launch wizard:** You can search for an AMI and launch an instance directly from the wizard. For more information, see [Launching an AWS Marketplace Instance \(p. 287\)](#).

Purchasing a Paid AMI From a Developer

The developer of a paid AMI can enable you to purchase a paid AMI that isn't listed in AWS Marketplace. The developer provides you with a link that enables you to purchase the product through Amazon. You can sign in with your Amazon.com credentials and select a credit card that's stored in your Amazon.com account to use when purchasing the AMI.

Getting the Product Code for Your Instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see [Instance Metadata and User Data \(p. 337\)](#).

To retrieve a product code, use the following query:

```
$ GET http://169.254.169.254/latest/meta-data/product-codes
```

If the instance has a product code, Amazon EC2 returns it. For example:

```
774F4FF8
```

Using Paid Support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

Important

You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- `modify-image-attribute` (AWS CLI)

```
$ aws ec2 modify-image-attribute --image-id ami_id --product-codes  
"product_code"
```

- `Edit-EC2ImageAttribute` (AWS Tools for Windows PowerShell)

```
C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

After you set the product code attribute, it cannot be changed or removed.

Bills for Paid and Supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see [Paying For AWS Marketplace Products](#).

Managing Your AWS Marketplace Subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the [AWS Marketplace](#).
2. Click **Your Account**.
3. Click **Manage Your Software Subscriptions**.
4. All your current subscriptions are listed. Click **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
 - a. Open the Amazon EC2 console.
 - b. In the navigation pane, click **Instances**.
 - c. Select the instance, click **Actions**, select **Instance State**, and select **Terminate**. When prompted, click **Yes, Terminate**.

2. Log in to the [AWS Marketplace](#), and click **Your Account**, then **Manage Your Software Subscriptions**.
3. Click **Cancel subscription**. You are prompted to confirm your cancellation.

Note

After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

Creating an Amazon EBS-Backed Linux AMI

To create an Amazon EBS-backed Linux AMI, start from an instance that you've launched from an existing Amazon EBS-backed Linux AMI. After you've customized the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations. For more information about creating an Amazon EBS-backed Windows AMI, see [Creating an Amazon EBS-Backed Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

The procedures described below work for Amazon EC2 instances backed by encrypted Amazon EBS volumes (including the root volume) as well as for unencrypted volumes.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the Root Device \(p. 71\)](#). For more information about creating an instance store-backed Linux AMI, see [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#).

Overview of Creating Amazon EBS-Backed AMIs

First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is configured correctly, ensure data integrity by stopping the instance before you create an AMI, then create the image. When you create an Amazon EBS-backed AMI, we automatically register it for you.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can tell Amazon EC2 not to power down and reboot the instance. Some file systems, such as XFS, can freeze and unfreeze activity, making it safe to create the image without rebooting the instance.

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption \(p. 695\)](#).

Depending on the size of the volumes, it can take several minutes for the AMI-creation process to complete (sometimes up to 24 hours). You may find it more efficient to create snapshots of your volumes prior to creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same). For more information, see [Creating an Amazon EBS Snapshot \(p. 686\)](#).

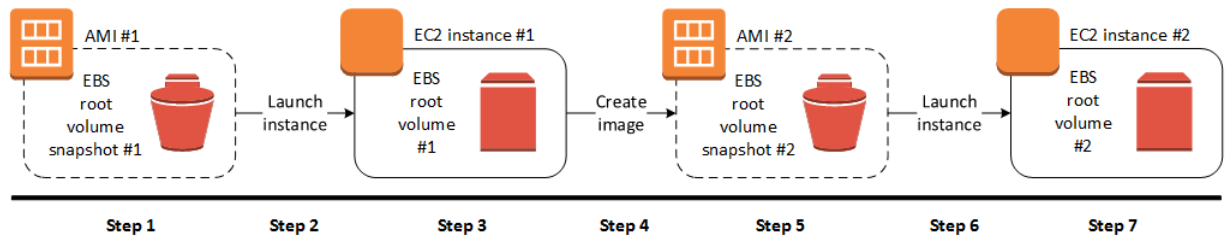
After the process completes, you have a new AMI and snapshot created from the root volume of the instance. When you launch an instance using the new AMI, we create a new EBS volume for its root volume using the snapshot. Both the AMI and the snapshot incur charges to your account until you delete them. For more information, see [Deregistering Your AMI \(p. 140\)](#).

If you add instance-store volumes or EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and

the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance-store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on EBS volumes persists. For more information, see [Block Device Mapping \(p. 731\)](#).

Creating a Linux AMI from an Instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an Amazon EBS-backed AMI from a running EC2 instance. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The steps in the following diagram match the steps in the procedure below.



To create an AMI from an instance using the console

1. Select an appropriate EBS-backed AMI to serve as a starting point for your new AMI, and configure it as needed prior to launch. For more information, see [Launching an Instance \(p. 279\)](#).
2. Choose **Launch** to launch an instance of the EBS-backed AMI that you've selected. Accept the default values as you step through the wizard. For more information, see [Launching an Instance \(p. 279\)](#).
3. While the instance is running, connect to it.

You can perform any of the following actions on your instance to customize it for your needs:

- Install software and applications
- Copy data
- Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space
- Attach additional Amazon EBS volumes

(Optional) Create snapshots of all the volumes attached to your instance. For more information about creating snapshots, see [Creating an Amazon EBS Snapshot \(p. 686\)](#).

In the navigation pane, choose **Instances** and select your instance. Choose **Actions, Image**, and **Create Image**.

Tip

If this option is disabled, your instance isn't an Amazon EBS-backed instance.

4. In the **Create Image** dialog box, specify values for the following fields, and then choose **Create Image**.

Name

A unique name for the image.

Description

(Optional) A description of the image, up to 255 characters.

By default, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Choose **No reboot** if you don't want your instance to be shut down.

Warning

If you choose **No reboot**, we can't guarantee the file system integrity of the created image.

You can modify the root volume, Amazon EBS volumes, and instance store volumes as follows:

- To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.
 - To suppress an Amazon EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the EBS volume in the list and choose **Delete**.
 - To add an Amazon EBS volume, choose **Add New Volume**, **Type**, and **EBS**, and fill in the fields. When you then launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume in the list and choose **Delete**.
 - To add an instance store volume, choose **Add New Volume**, **Type**, and **Instance Store**, and select a device name from the **Device** list. When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.
5. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially this will be `pending`. After a few minutes the status should change to `available`.

(Optional) Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

6. Launch an instance from your new AMI. For more information, see [Launching an Instance \(p. 279\)](#).
7. The new running instance contains all of the customizations you applied in previous steps.

To create an AMI from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-image` (AWS CLI)
- `New-EC2Image` (AWS Tools for Windows PowerShell)

Creating a Linux AMI from a Snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot using the AWS Management Console or the command line.

Important

Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the Amazon EC2 `billingProduct` code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot

does not maintain this billing code, and subsequent instances launched from such an AMI will not be able to connect to package update infrastructure.

Similarly, although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from the AMI.

In general, AWS advises against manually creating AMIs from snapshots.

For more information about creating Windows AMIs or AMIs for Linux operating systems that must retain AMI billing codes to work properly, see [Creating a Linux AMI from an Instance](#) (p. 90).

To create an AMI from a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots**.
3. Choose the snapshot and choose **Actions, Create Image**.
4. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, then choose **Create**. If you're re-creating a parent instance, then choose the same options as the parent instance.
 - **Architecture**: Choose **i386** for 32-bit or **x86_64** for 64-bit.
 - **Root device name**: Enter the appropriate name for the root volume. For more information, see [Device Naming on Linux Instances](#) (p. 729).
 - **Virtualization type**: Choose whether instances launched from this AMI use paravirtual (PV) or hardware virtual machine (HVM) virtualization. For more information, see [Linux AMI Virtualization Types](#) (p. 74).
 - (PV virtualization type only) **Kernel ID** and **RAM disk ID**: Choose the AKI and ARI from the lists. If you choose the default AKI or don't choose an AKI, you'll be required to specify an AKI every time you launch an instance using this AMI. In addition, your instance may fail the health checks if the default AKI is incompatible with the instance.
 - (Optional) **Block Device Mappings**: Add volumes or expand the default size of the root volume for the AMI. For more information about resizing the file system on your instance for a larger volume, see [Extending a Linux File System](#) (p. 673).

To create an AMI from a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [register-image](#) (AWS CLI)
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

Creating an Instance Store-Backed Linux AMI

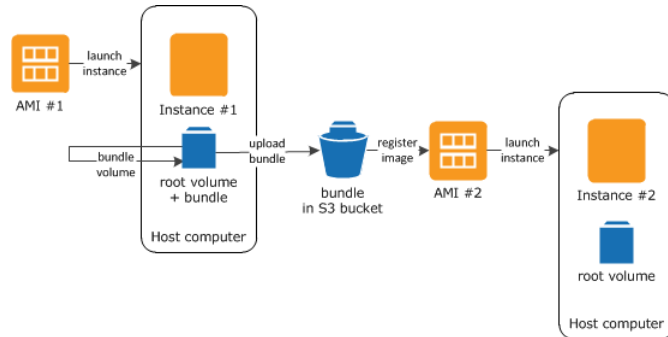
To create an instance store-backed Linux AMI, start from an instance that you've launched from an existing instance store-backed Linux AMI. After you've customized the instance to suit your needs, bundle the volume and register a new AMI, which you can use to launch new instances with these customizations.

If you need to create an instance store-backed Windows AMI, see [Creating an Instance Store-Backed Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the Root Device](#) (p. 71). If you need to create an Amazon EBS-backed Linux AMI, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89).

Overview of the Creation Process for Instance Store-Backed AMIs

The following diagram summarizes the process of creating an AMI from an instance store-backed instance.



First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, you can bundle it. It takes several minutes for the bundling process to complete. After the process completes, you have a bundle, which consists of an image manifest (`image.manifest.xml`) and files (`image.part.xx`) that contain a template for the root volume. Next you upload the bundle to your Amazon S3 bucket and then register your AMI.

When you launch an instance using the new AMI, we create the root volume for the instance using the bundle that you uploaded to Amazon S3. The storage space used by the bundle in Amazon S3 incurs charges to your account until you delete it. For more information, see [Deregistering Your AMI \(p. 140\)](#).

If you add instance store volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. For more information, see [Block Device Mapping \(p. 731\)](#).

Prerequisites

Before you can create an AMI, you must complete the following tasks:

- Install the AMI tools. For more information, see [Setting Up the AMI Tools \(p. 94\)](#).
- Install the AWS CLI. For more information, see [Getting Set Up with the AWS Command Line Interface](#).
- Ensure that you have an Amazon S3 bucket for the bundle. To create an Amazon S3 bucket, open the Amazon S3 console and click **Create Bucket**.

Note

You can also use the AWS CLI `mb` command to create a bucket.

- Ensure that you have the following credentials:
 - Your AWS account ID. To find your AWS account ID number in the AWS Management Console, click on **Support** in the navigation bar in the upper-right, and then click **Support Center**. Your currently signed in account ID appears below the **Support** menu.
 - An X.509 certificate and corresponding private key. If you need to create an X.509 certificate, see [Managing Signing Certificates \(p. 115\)](#). The X.509 certificate and private key are used to encrypt and decrypt your AMI.
 - Your AWS account access key ID and secret access key. For more information, see [Creating, Modifying, and Viewing Access Keys](#) in the *IAM User Guide*.

- Connect to your instance and customize it. For example, you can install software and applications, copy data, delete temporary files, and modify the Linux configuration.

Topics

- [Setting Up the AMI Tools \(p. 94\)](#)
- [Creating an AMI from an Instance Store-Backed Amazon Linux Instance \(p. 120\)](#)
- [Creating an AMI from an Instance Store-Backed Ubuntu Instance \(p. 124\)](#)
- [Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI \(p. 129\)](#)

Setting Up the AMI Tools

You can use the AMI tools to create and manage instance store-backed Linux AMIs. To use the tools, you must install them on your Linux instance. The AMI tools are available as both an RPM and as a .zip file for Linux distributions that don't support RPM. For more information, see [Amazon EC2 AMI Tools](#).

Note

The AMI tools are supported on instance store-backed Linux instances only. To create an Amazon EBS-backed AMI, use the [create-image](#) AWS CLI command instead. To create an instance store-backed Windows AMI, see [Creating an Instance Store-Backed Windows AMI](#).

To set up the AMI tools using the RPM

1. Install Ruby using the package manager for your Linux distribution, such as yum. For example:

```
$ sudo yum install -y ruby
```

2. Download the RPM file using a tool such as wget or curl. For example:

```
$ sudo wget http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. Install the RPM using the following command.

```
$ sudo yum install ec2-ami-tools.noarch.rpm
```

4. Verify your AMI tools installation with the following command.

```
$ ec2-ami-tools-version
```

Note

If you receive a load error such as `cannot load such file -- ec2/amitools/version (LoadError)`, complete the next step to add the location of your AMI tools installation to your `RUBYLIB` path.

5. (Optional) If you received an error in the previous step, add the location of your AMI tools installation to your `RUBYLIB` path.

- a. Run the following command to determine the paths to add.

```
$ rpm -qil ec2-ami-tools | grep ec2/amitools/version  
/usr/lib/ruby/site_ruby/ec2/amitools/version.rb  
/usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
```

In the above example, the missing file from the previous load error is located at `/usr/lib/ruby/site_ruby` and `/usr/lib64/ruby/site_ruby`.

- b. Add the locations from the previous step to your `RUBYLIB` path.

```
$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
```

- c. Verify your AMI tools installation with the following command.

```
$ ec2-ami-tools-version
```

To set up the AMI tools using the .zip file

1. Install Ruby and unzip using the package manager for your Linux distribution, such as **apt-get**. For example:

```
$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. Download the .zip file using a tool such as `wget` or `curl`. For example:

```
$ wget http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. Unzip the files into a suitable installation directory, such as `/usr/local/ec2`.

```
$ sudo mkdir -p /usr/local/ec2  
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

Notice that the .zip file contains a folder `ec2-ami-tools-x.x.x`, where `x.x.x` is the version number of the tools (for example, `ec2-ami-tools-1.5.7`).

4. Set the `EC2_AMITool_HOME` environment variable to the installation directory for the tools. For example:

```
$ export EC2_AMITool_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. Add the tools to your `PATH` environment variable. For example:

```
$ export PATH=$EC2_AMITool_HOME/bin:$PATH
```

6. You can verify your AMI tools installation with the following command.

```
$ ec2-ami-tools-version
```

AMI Tool Commands

You can use the following commands with the AMI tools to create and manage instance store-backed Linux AMIs. To set up the tools, see [Setting Up the AMI Tools \(p. 94\)](#).

Topics

- [ec2-ami-tools-version \(p. 96\)](#)
- [ec2-bundle-image \(p. 96\)](#)

- [ec2-bundle-vol](#) (p. 99)
- [ec2-delete-bundle](#) (p. 104)
- [ec2-download-bundle](#) (p. 106)
- [ec2-migrate-manifest](#) (p. 108)
- [ec2-unbundle](#) (p. 110)
- [ec2-upload-bundle](#) (p. 111)
- [Common Options for AMI Tools](#) (p. 115)

ec2-ami-tools-version

Description

Describes the version of the AMI tools.

Syntax

```
ec2-ami-tools-version
```

Options

This command has no parameters.

Output

The version information.

Example

This example command displays the version information for the AMI tools that you're using.

```
$ ec2-ami-tools-version
1.5.2 20071010
```

ec2-bundle-image

Description

Creates an instance store-backed Linux AMI from an operating system image created in a loopback file.

Syntax

```
ec2-bundle-image -c path -k path -u account -i path [-d path] [--ec2cert path] [-r architecture] [--productcodes code1,code2,...] [-B mapping] [-p prefix]
```

Options

Option	Description
<code>-c, --cert <i>path</i></code>	The user's PEM encoded RSA public key certificate file. Required: Yes Example: <code>-c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem</code>
<code>-k, --privatekey <i>path</i></code>	The path to a PEM-encoded RSA key file. You'll need to specify this key to unbundle this bundle, so keep it in a safe

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
	place. Note that the key doesn't have to be registered to your AWS account. Required: Yes Example: <code>-k pk-HKZYKTAIG2ECMXIIBH3HXV4ZBEXAMPLE.pem</code>
<code>-u, --user account</code>	The user's AWS account ID without dashes. Required: Yes Example: <code>-u 111122223333</code>
<code>-i, --image path</code>	The path to the image to bundle. Required: Yes Example: <code>-i /var/spool/my-image/version-2/debian.img</code>
<code>-d, --destination path</code>	The directory in which to create the bundle. Default: <code>/tmp</code> Required: No Example: <code>-d /media/ephemeral0</code>
<code>--ec2cert path</code>	The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest. The <code>us-gov-west-1</code> and <code>cn-north-1</code> regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at <code>/opt/aws/amitools/ec2/etc/ec2/amitools/</code> . If you installed the AMI tools from the RPM or ZIP file in Setting Up the AMI Tools (p. 94) , the certificates are located at <code>\$(EC2_AMITOOL_HOME)/etc/ec2/amitools/</code> . Default: varies, depending on the tools Required: Only for the <code>us-gov-west-1</code> and <code>cn-north-1</code> regions. Example: <code>--ec2cert \$(EC2_AMITOOL_HOME)/etc/ec2/amitools/cert-ec2.pem</code>
<code>-r, --arch architecture</code>	Image architecture. If you don't provide the architecture on the command line, you'll be prompted for it when bundling starts. Valid values: <code>i386 x86_64</code> Required: No Example: <code>-r x86_64</code>
<code>--productcodes code1,code2,...</code>	Product codes to attach to the image at registration time, separated by commas. Required: No Example: <code>--productcodes 1234abcd</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>-B, --block-device-mapping <i>mapping</i></code>	<p>Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.</p> <p>Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:</p> <ul style="list-style-type: none"> <code>ami</code>—The root file system device, as seen by the instance <code>root</code>—The root file system device, as seen by the kernel <code>swap</code>—The swap device, as seen by the instance <code>ephemeralN</code>—The Nth instance store volume <p>Required: No Example: <code>--block-device-mapping ami=sda1,root=/dev/sda1,ephemeral0=sda2,swap=sda3</code> Example: <code>--block-device-mapping ami=0,root=/dev/dsk/c0d0s0,ephemeral0=1</code></p>
<code>-p, --prefix <i>prefix</i></code>	<p>The filename prefix for bundled AMI files.</p> <p>Default: The name of the image file. For example, if the image path is <code>/var/spool/my-image/version-2/debian.img</code>, then the default prefix is <code>debian.img</code>.</p> <p>Required: No Example: <code>-p my-image-is-special</code></p>
<code>--kernel <i>kernel_id</i></code>	<p>Deprecated. Use register-image to set the kernel.</p> <p>Required: No Example: <code>--kernel aki-ba3adfd3</code></p>
<code>--ramdisk <i>ramdisk_id</i></code>	<p>Deprecated. Use register-image to set the RAM disk if required.</p> <p>Required: No Example: <code>--ramdisk ari-badbad00</code></p>
Common options	<p>For options common to most of the AMI Tools, see Common Options for AMI Tools (p. 115).</p>

Output

Status messages describing the stages and status of the bundling process.

Example

This example creates a bundled AMI from an operating system image that was created in a loopback file.

```
$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/
-r x86_64
Please specify a value for arch [i386]:
```

```
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

Description

Creates an instance store-backed Linux AMI by compressing, encrypting, and signing a copy of the root device volume for the instance.

Amazon EC2 attempts to inherit product codes, kernel settings, RAM disk settings, and block device mappings from the instance.

By default, the bundle process excludes files that might contain sensitive information. These files include *.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa* *.gpg, *.jks, */.ssh/authorized_keys, and */.bash_history. To include all of these files, use the --no-filter option. To include some of these files, use the --include option.

For more information, see [Creating an Instance Store-Backed Linux AMI](#).

Syntax

```
ec2-bundle-vol -c path -k path -u account [-d path] [--ec2cert path] [-r architecture] [--productcodes code1,code2,...] [-B mapping] [--all] [-e directory1,directory2,...] [-i file1,file2,...] [--no-filter] [-p prefix] [-s size] [--[no-]inherit] [-v volume] [-P type] [-S script] [--fstab path] [--generate-fstab] [--grub-config path]
```

Options

Option	Description
-c, --cert <i>path</i>	The user's PEM encoded RSA public key certificate file. Required: Yes Example: -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-k, --privatekey <i>path</i>	The path to the user's PEM-encoded RSA key file. Required: Yes

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
	Example: <code>-k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem</code>
<code>-u, --user account</code>	The user's AWS account ID without dashes. Required: Yes Example: <code>-u 111122223333</code>
<code>-d, --destination destination</code>	The directory in which to create the bundle. Default: <code>/tmp</code> Required: No Example: <code>-d /var/run/my-bundle</code>
<code>--ec2cert path</code>	The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest. The <code>us-gov-west-1</code> and <code>cn-north-1</code> regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at <code>/opt/aws/amitools/ec2/etc/ec2/amitools/</code> . If you installed the AMI tools from the RPM or ZIP file in Setting Up the AMI Tools (p. 94) , the certificates are located at <code>\$(EC2_AMITOOL_HOME)/etc/ec2/amitools/</code> . Default: varies, depending on the tools Required: Only for the <code>us-gov-west-1</code> and <code>cn-north-1</code> regions. Example: <code>--ec2cert \$(EC2_AMITOOL_HOME)/etc/ec2/amitools/cert-ec2.pem</code>
<code>-r, --arch architecture</code>	The image architecture. If you don't provide this on the command line, you'll be prompted to provide it when the bundling starts. Valid values: <code>i386 x86_64</code> Required: No Example: <code>-r x86_64</code>
<code>--productcodes code1,code2,...</code>	Product codes to attach to the image at registration time, separated by commas. Required: No Example: <code>--productcodes 1234abcd</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>-B, --block-device-mapping <i>mapping</i></code>	<p>Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.</p> <p>Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:</p> <ul style="list-style-type: none"> • <code>ami</code>—The root file system device, as seen by the instance • <code>root</code>—The root file system device, as seen by the kernel • <code>swap</code>—The swap device, as seen by the instance • <code>ephemeralN</code>—The Nth instance store volume <p>Required: No Example: <code>--block-device-mapping ami=sda1,root=/dev/sda1,ephemeral0=sda2,swap=sda3</code> Example: <code>--block-device-mapping ami=0,root=/dev/dsk/c0d0s0,ephemeral0=1</code></p>
<code>-a, --all</code>	<p>Bundle all directories, including those on remotely mounted file systems.</p> <p>Required: No Example: <code>-a</code></p>
<code>-e, --exclude <i>directory1,directory2,...</i></code>	<p>A list of absolute directory paths and files to exclude from the bundle operation. This parameter overrides the <code>--all</code> option. When <code>exclude</code> is specified, the directories and subdirectories listed with the parameter will not be bundled with the volume.</p> <p>Required: No Example: Assuming the mount point of the volume is <code>-v /foo</code>, and you want to exclude directories <code>/foo/bar</code> and <code>/foo/baz</code>, specify <code>-e /bar,/baz</code>.</p>
<code>-i, --include <i>file1,file2,...</i></code>	<p>A list of files to include in the bundle operation. The specified files would otherwise be excluded from the AMI because they might contain sensitive information.</p> <p>Required: No Example: If the volume mount point is <code>/mnt/myvol/</code> and you want to include the file <code>/mnt/myvol/foo/bar.pem</code>, specify <code>-i /foo/bar.pem</code>.</p>
<code>--no-filter</code>	<p>If specified, we won't exclude files from the AMI because they might contain sensitive information.</p> <p>Required: No Example: <code>--no-filter</code></p>
<code>-p, --prefix <i>prefix</i></code>	<p>The file name prefix for bundled AMI files.</p> <p>Default: <code>image</code> Required: No Example: <code>-p my-image-is-special</code></p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>-s, --size size</code>	The size, in MB (1024 * 1024 bytes), of the image file to create. The maximum size is 10240 MB. Default: 10240 Required: No Example: <code>-s 2048</code>
<code>--[no-]inherit</code>	Indicates whether the image should inherit the instance's metadata (the default is to inherit). Bundling fails if you enable <code>--inherit</code> but the instance metadata is not accessible. Required: No Example: <code>--inherit</code>
<code>-v, --volume volume</code>	The absolute path to the mounted volume from which to create the bundle. Default: The root directory (<code>/</code>) Required: No Example: <code>-v /mnt/my-customized-ami</code>
<code>-P, --partition type</code>	Indicates whether the disk image should use a partition table. If you don't specify a partition table type, the default is the type used on the parent block device of the volume, if applicable, otherwise the default is <code>gpt</code> . Valid values: <code>mbr gpt none</code> Required: No Example: <code>--partition gpt</code>
<code>-S, --script script</code>	A customization script to be run right before bundling. The script must expect a single argument, the mount point of the volume. Required: No
<code>--fstab path</code>	The path to the <code>fstab</code> to bundle into the image. If this is not specified, Amazon EC2 bundles <code>/etc/fstab</code> . Required: No Example: <code>--fstab /etc/fstab</code>
<code>--generate-fstab</code>	Bundles the volume using an Amazon EC2-provided <code>fstab</code> . Required: No Example: <code>--generate-fstab</code>
<code>--grub-config</code>	The path to an alternate <code>grub</code> configuration file to bundle into the image. By default, <code>ec2-bundle-vol</code> expects either <code>/boot/grub/menu.lst</code> or <code>/boot/grub/grub.conf</code> to exist on the cloned image. This option allows you to specify a path to an alternative <code>grub</code> configuration file, which will then be copied over the defaults (if present). Required: No Example: <code>--grub-config /path/to/grub.conf</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>--kernel <i>kernel_id</i></code>	Deprecated. Use register-image to set the kernel. Required: No Example: <code>--kernel aki-ba3adfd3</code>
<code>--ramdisk <i>ramdisk_id</i></code>	Deprecated. Use register-image to set the RAM disk if required. Required: No Example: <code>--ramdisk ari-badbad00</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Output

Status messages describing the stages and status of the bundling.

Example

This example creates a bundled AMI by compressing, encrypting and signing a snapshot of the local machine's root file system.

```
$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMYIBH3HXV4ZBEXAMPLE.pem -c cert-
HKZYKTAIG2ECMYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
  Copying / into the image file /mnt/image...
  Excluding:
    sys
    dev/shm
    proc
    dev/pts
    proc/sys/fs/binfmt_misc
    dev
    media
    mnt
    proc
    sys
    tmp/image
    mnt/img-mnt
  1+0 records in
  1+0 records out
  mke2fs 1.38 (30-Jun-2005)
  warning: 256 blocks unused.

  Splitting /mnt/image.gz.crypt...
  Created image.part.00
  Created image.part.01
  Created image.part.02
  Created image.part.03
  ...
  Created image.part.22
  Created image.part.23
  Generating digests for each part...
  Digests generated.
  Creating bundle manifest...
  Bundle Volume complete.
```

ec2-delete-bundle

Description

Deletes the specified bundle from Amazon S3 storage. After you delete a bundle, you can't launch instances from the corresponding AMI.

Syntax

```
ec2-delete-bundle -b bucket -a access_key_id -s secret_access_key [-t token]
[--url url] [--region region] [--sigv version] [-m path] [-p prefix] [--
clear] [--retry] [-y]
```

Options

Option	Description
-b, --bucket <i>bucket</i>	The name of the Amazon S3 bucket containing the bundled AMI, followed by an optional '/'-delimited path prefix Required: Yes Example: -b myawsbucket/ami-001
-a, --access-key <i>access_key_id</i>	The AWS access key ID. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Yes Example: -a AKIAIOSFODNN7EXAMPLE
-s, --secret-key <i>secret_access_key</i>	The AWS secret access key. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Yes Example: -s wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
-t, --delegation-token <i>token</i>	The delegation token to pass along to the AWS request. For more information, see Using Temporary Security Credentials . Required: Only when you are using temporary security credentials. Default: The value of the AWS_DELEGATION_TOKEN environment variable (if set). Example: -t AQoDYXdzEJr...<remainder of security token>
--region <i>region</i>	The region to use in the request signature. Default: us-east-1 Required: Conditional Condition: Required if using signature version 4 Example: --region eu-west-1
--sigv <i>version</i>	The signature version to use when signing the request. Valid values: 2 4 Default: 4 Required: No Example: --sigv 2

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>-m, --manifest path</code>	The path to the manifest file. Required: Conditional Condition: You must specify <code>--prefix</code> or <code>--manifest</code> . Example: <code>-m /var/spool/my-first-bundle/image.manifest.xml</code>
<code>-p, --prefix prefix</code>	The bundled AMI filename prefix. Provide the entire prefix. For example, if the prefix is <code>image.img</code> , use <code>-p image.img</code> and not <code>-p image</code> . Required: Conditional Condition: You must specify <code>--prefix</code> or <code>--manifest</code> . Example: <code>-p image.img</code>
<code>--clear</code>	Deletes the Amazon S3 bucket if it's empty after deleting the specified bundle. Required: No Example: <code>--clear</code>
<code>--retry</code>	Automatically retries on all Amazon S3 errors, up to five times per operation. Required: No Example: <code>--retry</code>
<code>-y, --yes</code>	Automatically assumes the answer to all prompts is <code>yes</code> . Required: No Example: <code>-y</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Output

Amazon EC2 displays status messages indicating the stages and status of the delete process.

Example

This example deletes a bundle from Amazon S3.

```
$ ec2-delete-bundle -b myawsbucket -a your_access_key_id -
s your_secret_access_key
Deleting files:
myawsbucket/image.manifest.xml
myawsbucket/image.part.00
myawsbucket/image.part.01
myawsbucket/image.part.02
myawsbucket/image.part.03
myawsbucket/image.part.04
myawsbucket/image.part.05
myawsbucket/image.part.06
Continue? [y/n]
y
Deleted myawsbucket/image.manifest.xml
Deleted myawsbucket/image.part.00
Deleted myawsbucket/image.part.01
```


**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

```
Deleted myawsbucket/image.part.02
Deleted myawsbucket/image.part.03
Deleted myawsbucket/image.part.04
Deleted myawsbucket/image.part.05
Deleted myawsbucket/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

Description

Downloads the specified instance store-backed Linux AMIs from Amazon S3 storage.

Syntax

```
ec2-download-bundle -b bucket -a access_key_id -s secret_access_key -k path
[--url url] [--region region] [--sigv version] [-m file] [-p prefix] [-d
directory] [--retry]
```

Options

Option	Description
-b, --bucket <i>bucket</i>	The name of the Amazon S3 bucket where the bundle is located, followed by an optional '/'-delimited path prefix. Required: Yes Example: -b myawsbucket/ami-001
-a, --access-key <i>access_key_id</i>	The AWS access key ID. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Yes Example: -a AKIAIOSFODNN7EXAMPLE
-s, --secret-key <i>secret_access_key</i>	The AWS secret access key. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Yes Example: -s wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
-k, --privatekey <i>path</i>	The private key used to decrypt the manifest. Required: Yes Example: -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
--url <i>url</i>	The Amazon S3 service URL. Default: https://s3.amazonaws.com Required: No Example: --url https://s3.example.com
--region <i>region</i>	The region to use in the request signature. Default: us-east-1 Required: Conditional Condition: Required if using signature version 4 Example: --region eu-west-1

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>--sigv version</code>	The signature version to use when signing the request. Valid values: 2 4 Default: 4 Required: No Example: <code>--sigv 2</code>
<code>-m, --manifest file</code>	The name of the manifest file (without the path). We recommend that you specify either the manifest (<code>-m</code>) or a prefix (<code>-p</code>). Required: No Example: <code>-m my-image.manifest.xml</code>
<code>-p, --prefix prefix</code>	The filename prefix for the bundled AMI files. Default: <code>image</code> Required: No Example: <code>-p my-image</code>
<code>-d, --directory directory</code>	The directory where the downloaded bundle is saved. The directory must exist. Default: The current working directory. Required: No Example: <code>-d /tmp/my-downloaded-bundle</code>
<code>--retry</code>	Automatically retries on all Amazon S3 errors, up to five times per operation. Required: No Example: <code>--retry</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Output

Status messages indicating the various stages of the download process are displayed.

Example

This example creates the `bundled` directory (using the Linux `mkdir` command) and downloads the bundle from the `myawsbucket` Amazon S3 bucket.

```
$ mkdir bundled
$ ec2-download-bundle -b myawsbucket/bundles/bundle_name -m
  image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from myawsbucket to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from myawsbucket/bundles/bundle_name to
mybundle/image.part.00 ...
Downloaded image.part.00 from myawsbucket
Downloading part image.part.01 from myawsbucket/bundles/bundle_name to
mybundle/image.part.01 ...
Downloaded image.part.01 from myawsbucket
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

```
Downloading part image.part.02 from myawsbucket/bundles/bundle_name to  
mybundle/image.part.02 ...  
Downloaded image.part.02 from myawsbucket  
Downloading part image.part.03 from myawsbucket/bundles/bundle_name to  
mybundle/image.part.03 ...  
Downloaded image.part.03 from myawsbucket  
Downloading part image.part.04 from myawsbucket/bundles/bundle_name to  
mybundle/image.part.04 ...  
Downloaded image.part.04 from myawsbucket  
Downloading part image.part.05 from myawsbucket/bundles/bundle_name to  
mybundle/image.part.05 ...  
Downloaded image.part.05 from myawsbucket  
Downloading part image.part.06 from myawsbucket/bundles/bundle_name to  
mybundle/image.part.06 ...  
Downloaded image.part.06 from myawsbucket
```

ec2-migrate-manifest

Description

Modifies an instance store-backed Linux AMI (for example, its certificate, kernel, and RAM disk) so that it supports a different region.

Syntax

```
ec2-migrate-manifest -c path -k path -m path {(-a access_key_id -  
s secret_access_key --region region) | (--no-mapping)} [--ec2cert  
ec2_cert_path] [--kernel kernel-id] [--ramdisk ramdisk_id]
```

Options

Option	Description
<code>-c, --cert <i>path</i></code>	The user's PEM encoded RSA public key certificate file. Required: Yes Example: <code>-c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem</code>
<code>-k, --privatekey <i>path</i></code>	The path to the user's PEM-encoded RSA key file. Required: Yes Example: <code>-k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem</code>
<code>--manifest <i>path</i></code>	The path to the manifest file. Required: Yes Example: <code>--manifest my-ami.manifest.xml</code>
<code>-a, --access-key <i>access_key_id</i></code>	The AWS access key ID. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Conditional Condition: Required if using automatic mapping. Example: <code>-a AKIAIOSFODNN7EXAMPLE</code>
<code>-s, --secret-key <i>secret_access_key</i></code>	The AWS secret access key. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys .

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
	<p>Required: Conditional</p> <p>Condition: Required if using automatic mapping.</p> <p>Example: <code>-s wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</code></p>
<code>--region region</code>	<p>The region to look up in the mapping file.</p> <p>Condition: Required if using automatic mapping.</p> <p>Required: Conditional</p> <p>Example: <code>--region eu-west-1</code></p>
<code>--no-mapping</code>	<p>Disables automatic mapping of kernels and RAM disks. During migration, Amazon EC2 replaces the kernel and RAM disk in the manifest file with a kernel and RAM disk designed for the destination region. Unless the <code>--no-mapping</code> parameter is given, <code>ec2-migrate-bundle</code> might use the <code>DescribeRegions</code> and <code>DescribeImages</code> operations to perform automated mappings.</p> <p>Required: Conditional</p> <p>Condition: Required if you're not providing the <code>-a</code>, <code>-s</code>, and <code>--region</code> options (which are used for automatic mapping).</p>
<code>--ec2cert path</code>	<p>The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.</p> <p>The <code>us-gov-west-1</code> and <code>cn-north-1</code> regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at <code>/opt/aws/amitools/ec2/etc/ec2/amitools/</code>. If you installed the AMI tools from the ZIP file in Setting Up the AMI Tools (p. 94), the certificates are located at <code>\$EC2_AMITOOL_HOME/etc/ec2/amitools/</code>.</p> <p>Default: varies, depending on the tools</p> <p>Required: Only for the <code>us-gov-west-1</code> and <code>cn-north-1</code> regions.</p> <p>Example: <code>--ec2cert \$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem</code></p>
<code>--kernel kernel_id</code>	<p>The ID of the kernel to select.</p> <p>Important</p> <p>We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see PV-GRUB.</p> <p>Required: No</p> <p>Example: <code>--kernel aki-ba3adfd3</code></p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>--ramdisk <i>ramdisk_id</i></code>	The ID of the RAM disk to select. Important We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see PV-GRUB . Required: No Example: <code>--ramdisk ari-badbad00</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Output

Status messages describing the stages and status of the bundling process.

Example

This example copies the AMI specified in the `my-ami.manifest.xml` manifest from the US to the EU.

```
$ ec2-migrate-manifest --manifest my-ami.manifest.xml --cert
cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-
HKZYKTAIG2ECMXIYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-
west-1.
```

ec2-unbundle

Description

Re-creates the bundle from an instance store-backed Linux AMI.

Syntax

```
ec2-unbundle -k path -m path [-s source_directory] [-d destination_directory]
```

Options

Option	Description
<code>-k, --privatekey <i>path</i></code>	The path to your PEM-encoded RSA key file. Required: Yes Example: <code>-k \$HOME/pk-234242example.pem</code>
<code>-m, --manifest <i>path</i></code>	The path to the manifest file. Required: Yes Example: <code>-m /var/spool/my-first-bundle/Manifest</code>
<code>-s, --source <i>source_directory</i></code>	The directory containing the bundle. Default: The current directory. Required: No Example: <code>-s /tmp/my-bundled-image</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>-d, --destination <i>destination_directory</i></code>	The directory in which to unbundle the AMI. The destination directory must exist. Default: The current directory. Required: No Example: <code>-d /tmp/my-image</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Example

This Linux and UNIX example unbundles the AMI specified in the `image.manifest.xml` file.

```
$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-
HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -s mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

Output

Status messages indicating the various stages of the unbundling process are displayed.

ec2-upload-bundle

Description

Uploads the bundle for an instance store-backed Linux AMI to Amazon S3 and sets the appropriate ACLs on the uploaded objects. For more information, see [Creating an Instance Store-Backed Linux AMI](#).

Syntax

```
ec2-upload-bundle -b bucket -a access_key_id -s secret_access_key [-t token]
-m path [--url url] [--region region] [--sigv version] [--acl acl] [-d
directory] [--part part] [--retry] [--skipmanifest]
```

Options

Option	Description
<code>-b, --bucket <i>bucket</i></code>	The name of the Amazon S3 bucket in which to store the bundle, followed by an optional '/'-delimited path prefix. If the bucket doesn't exist, it's created if the bucket name is available. Required: Yes Example: <code>-b myawsbucket/bundles/ami-001</code>
<code>-a, --access-key <i>access_key_id</i></code>	Your AWS access key ID. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys . Required: Yes Example: <code>-a AKIAIOSFODNN7EXAMPLE</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<p><code>-s, --secret-key</code> <code>secret_access_key</code></p>	<p>Your AWS secret access key. Before you specify a value for this option, review and follow the guidance in Best Practices for Managing AWS Access Keys.</p> <p>Required: Yes</p> <p>Example: <code>-s wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY</code></p>
<p><code>-t, --delegation-token</code> <code>token</code></p>	<p>The delegation token to pass along to the AWS request. For more information, see Using Temporary Security Credentials.</p> <p>Required: Only when you are using temporary security credentials.</p> <p>Default: The value of the <code>AWS_DELEGATION_TOKEN</code> environment variable (if set).</p> <p>Example: <code>-t AQoDYXdzEJr...<remainder of security token></code></p>
<p><code>-m, --manifest path</code></p>	<p>The path to the manifest file. The manifest file is created during the bundling process and can be found in the directory containing the bundle.</p> <p>Required: Yes</p> <p>Example: <code>-m image.manifest.xml</code></p>
<p><code>--url url</code></p>	<p>Deprecated. Use the <code>--region</code> option instead unless your bucket is constrained to the EU location (and not <code>eu-west-1</code>). The <code>--location</code> flag is the only way to target that specific location restraint.</p> <p>The Amazon S3 endpoint service URL.</p> <p>Default: <code>https://s3.amazonaws.com</code></p> <p>Required: No</p> <p>Example: <code>--url https://s3.example.com</code></p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>--region <i>region</i></code>	<p>The region to use in the request signature for the destination Amazon S3 bucket.</p> <ul style="list-style-type: none"> • If the bucket doesn't exist and you don't specify a region, the tool creates the bucket without a location constraint (in <code>us-east-1</code>). • If the bucket doesn't exist and you specify a region, the tool creates the bucket in the specified region. • If the bucket exists and you don't specify a region, the tool uses the bucket's location. • If the bucket exists and you specify <code>us-east-1</code> as the region, the tool uses the bucket's actual location without any error message, any existing matching files are overwritten. • If the bucket exists and you specify a region (other than <code>us-east-1</code>) that doesn't match the bucket's actual location, the tool exits with an error. <p>If your bucket is constrained to the EU location (and not <code>eu-west-1</code>), use the <code>--location</code> flag instead. The <code>--location</code> flag is the only way to target that specific location restraint.</p> <p>Default: <code>us-east-1</code> Required: Conditional Condition: Required if using signature version 4 Example: <code>--region eu-west-1</code></p>
<code>--sigv <i>version</i></code>	<p>The signature version to use when signing the request.</p> <p>Valid values: <code>2 4</code> Default: <code>4</code> Required: No Example: <code>--sigv 2</code></p>
<code>--acl <i>acl</i></code>	<p>The access control list policy of the bundled image.</p> <p>Valid values: <code>public-read aws-exec-read</code> Default: <code>aws-exec-read</code> Required: No Example: <code>--acl public-read</code></p>
<code>-d, --directory <i>directory</i></code>	<p>The directory containing the bundled AMI parts.</p> <p>Default: The directory containing the manifest file (see the <code>-m</code> option).</p> <p>Required: No Example: <code>-d /var/run/my-bundle</code></p>
<code>--part <i>part</i></code>	<p>Starts uploading the specified part and all subsequent parts.</p> <p>Required: No Example: <code>--part 04</code></p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Setting Up the AMI Tools**

Option	Description
<code>--retry</code>	Automatically retries on all Amazon S3 errors, up to five times per operation. Required: No Example: <code>--retry</code>
<code>--skipmanifest</code>	Does not upload the manifest. Required: No Example: <code>--skipmanifest</code>
<code>--location location</code>	Deprecated. Use the <code>--region</code> option instead, unless your bucket is constrained to the EU location (and not <code>eu-west-1</code>). The <code>--location</code> flag is the only way to target that specific location restraint. The location constraint of the destination Amazon S3 bucket. If the bucket exists and you specify a location that doesn't match the bucket's actual location, the tool exits with an error. If the bucket exists and you don't specify a location, the tool uses the bucket's location. If the bucket doesn't exist and you specify a location, the tool creates the bucket in the specified location. If the bucket doesn't exist and you don't specify a location, the tool creates the bucket without a location constraint (in <code>us-east-1</code>). Default: If <code>--region</code> is specified, the location is set to that specified region. If <code>--region</code> is not specified, the location defaults to <code>us-east-1</code> . Required: No Example: <code>--location eu-west-1</code>
Common options	For options common to most of the AMI tools, see Common Options for AMI Tools (p. 115) .

Output

Amazon EC2 displays status messages that indicate the stages and status of the upload process.

Example

This example uploads the bundle specified by the `image.manifest.xml` manifest.

```
$ ec2-upload-bundle -b myawsbucket/bundles/bundle_name -m image.manifest.xml
-a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket myawsbucket ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
```

```
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

Common Options for AMI Tools

Most of the commands described in this section accept the set of optional parameters described in the following table.

Option	Description
--help, -h	Displays the help message.
--version	Displays the version and copyright notice.
--manual	Displays the manual entry.
--batch	Runs in batch mode, suppressing interactive prompts.
--debug	Displays debugging information that may be useful when troubleshooting problems.

Managing Signing Certificates

This section describes how to create and manage signing certificates; also known as X.509 certificates. These certificates are required for certain AMI tool commands.

Important

Amazon EC2 originally supported the SOAP protocol for making service calls, and SOAP-based calls use a signing certificate in order to digitally sign the requests. However, support for SOAP in Amazon EC2 is deprecated (see [SOAP Requests](#)), and you should use HTTP query requests instead. For more information, see [Making API Requests](#).

Each user can have two certificates for the purposes of credential rotation.

Note

You can give your users permission to list and manage their own certificates. For more information, see [Allow Users to Manage Their Own Passwords, Access Keys, and Signing Certificate](#) in the *IAM User Guide*.

Topics

- [Creating a User Signing Certificate \(p. 115\)](#)
- [Managing a User Signing Certificate \(p. 119\)](#)

Creating a User Signing Certificate

If you need a signing certificate, you must first obtain one, and then upload it to AWS. There is no Amazon EC2 API action to create signing certificates, so you must use a third-party tool such as OpenSSL to create the user signing certificate.

Note

Although you can use the security credentials page in the AWS Management Console to create an X.509 certificate, that method is only for the AWS account root credentials. You

can't upload a certificate generated using the console for individual Amazon EC2 users. Instead, use the process described in the next sections.

To create a signing certificate, you must do the following:

- Install and configure OpenSSL.
- Create a private key.
- Generate a certificate using the private key.
- Upload the certificate to AWS.

Install and Configure OpenSSL

Creating and uploading a certificate requires a tool that supports the SSL and TLS protocols. OpenSSL is an open-source tool that provides the basic cryptographic functions necessary to create an RSA token and sign it with your private key. If you don't already have OpenSSL installed, follow these instructions.

To install OpenSSL on Linux and UNIX

1. Go to [OpenSSL: Source, Tarballs](http://www.openssl.org/source/) (<http://www.openssl.org/source/>).
2. Download the latest source and build the package.

To install OpenSSL on Windows

1. Go to [Binaries](https://wiki.openssl.org/index.php/Binaries) (<https://wiki.openssl.org/index.php/Binaries>).
2. Choose the appropriate **OpenSSL for Windows** option.

A new page displays with links to the Windows downloads.

3. If it is not already installed on your system, select the **Microsoft Visual C++ 2008 Redistributables** link appropriate for your environment and click **Download**. Follow the instructions provided by the **Microsoft Visual C++ 2008 Redistributable Setup Wizard**.

Note

If you are not sure if the Microsoft Visual C++ 2008 Redistributable package is already installed on your system, you can try installing OpenSSL first. The OpenSSL installer displays an error if the Microsoft Visual C++ 2008 Redistributable package is not yet installed. Make sure you install the architecture (32-bit or 64-bit) that matches the version of OpenSSL that you install.

4. After you have installed the Microsoft Visual C++ 2008 Redistributable package, select the appropriate version of the OpenSSL binaries for your environment and save the file locally. Launch the **OpenSSL Setup Wizard**.
5. Follow the instructions described in the **OpenSSL Setup Wizard**.

Before you use OpenSSL commands, you must configure the operating system so that it has information about the location where OpenSSL is installed.

To configure OpenSSL on Linux or Unix

1. At the command line, set the `OpenSSL_HOME` variable to the location of the OpenSSL installation:

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

2. Set the path to include the OpenSSL installation:

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

Note

Any changes you make to environment variables using the `export` command are valid only for the current session. You can make persistent changes to the environment variables by setting them using your shell configuration file. For more information, see the documentation for your operating system.

To configure OpenSSL on Windows

1. Open a **Command Prompt** window.
2. Set the `OpenSSL_HOME` variable to the location of the OpenSSL installation:

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

3. Set the `OpenSSL_CONF` variable to the location of the configuration file in your OpenSSL installation:

```
set OpenSSL_CONF=path_to_your_OpenSSL_installation\bin\openssl.cfg
```

4. Set the path to include the OpenSSL installation:

```
set Path=%Path%;%OpenSSL_HOME%\bin
```

Note

Any changes you make to Windows environment variables in a **Command Prompt** window are valid only for the current command line session. You can make persistent changes to the environment variables by setting them as system properties. The exact procedures depends on what version of Windows you're using. For more information, see the Windows documentation.

Create a Private Key

You need a unique private key that you use when generating the user signing certificate.

To create a private key

1. At the command line, use the `openssl genrsa` command with the following syntax:

```
openssl genrsa 2048 > private-key.pem
```

For *private-key.pem*, specify your own file name. In the example, 2048 represents 2048-bit encryption. AWS also supports 1024-bit and 4096-bit encryption. We recommend that you create a 2048-bit or 4096-bit RSA key.

2. If you will be using the certificate to authenticate CLI commands for Auto Scaling, CloudWatch, or Elastic Load Balancing, generate the certificate in PKCS8 format using the following command:

```
openssl pkcs8 -topk8 -nocrypt -inform PEM -in private-key.pem -  
out private-key-in-PKCS8-format.pem
```

Create the User Signing Certificate

You can now create a user signing certificate.

To create a user signing certificate

- Use the `openssl req` command with the following syntax:

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -  
outform PEM -out certificate.pem
```

For *private-key.pem*, use the .pem file that you generated in a previous procedure. For *certificate.pem*, use the name of a file into which you want the certificate to be generated. The certificate must be in .pem format. For security, we recommend using either SHA-256, as in this example, or SHA-512 as your hash algorithm.

In this example, the `-days 365` switch specifies that the certificate is good for 365 days. For information about the other switches, enter `openssl req -h` at the command line.

OpenSSL displays a message similar to the following:

```
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a  
DN.  
There are quite a few fields but you can leave some blank.  
For some fields there will be a default value.  
If you enter '.', the field will be left blank.
```

Because you're creating a user signing certificate (not a server certificate), you can leave all the values blank when you're prompted. These values are used by the certificate authority (CA) to help authenticate the server certificate. However, because user signing certificates are uploaded in an authenticated session, AWS does not need any information in the certificate for further validation, and requires only the public-private key pair.

The .pem file contains the certificate value that you can copy and paste during the upload procedure that follows.

Upload the User Signing Certificate

You can upload a signing certificate using the console, API, or CLI.

To upload a signing certificate for a user using the console

- Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
- In the navigation pane, click **Users**.
- Click the name of the user for whom you want to upload the certificate, and then open the **Security Credentials** section. You may have to scroll down to see this section.
- Click **Manage Signing Certificates**.
- Click **Upload Signing Certificate** in the lower part of the page.
- Using a text editor, open the .pem file that contains the user signing certificate that you created using OpenSSL, as described earlier in this section, and copy the contents of the .pem file.
- Paste the contents of the user's signing certificate into the **Certificate Body** field, and then click **Upload Signing Certificate**.

Note

After the certificate has been uploaded, it cannot be retrieved, its contents cannot be viewed, and it cannot be reused.

To upload a certificate (CLI and API)

- [upload-signing-certificate](#) (AWS CLI)
- [UploadSigningCertificate](#) (IAM API)

Note

Use a POST request when uploading a signing certificate because of the certificate's size.

Managing a User Signing Certificate

You can manage a signing certificate using the console, API, or CLI.

As with access keys, each certificate can have a status of either `Active` or `Inactive`. By default, the status is `Active` when you upload the certificate. When you upload the certificate, it returns a certificate ID that you can save for your records. You can list the IDs for the user's certificates. You can delete a certificate at any time.

To modify or delete a signing certificate for a user

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, click **Users**.
3. Click the name of the user for whom you want to modify or delete the certificate, and then open the **Security Credentials** section.
4. Click **Manage Signing Certificates**.
5. To disable or re-enable a signing certificate, click **Make Inactive** or **Make Active**.
6. To delete a user's signing certificate, click **Delete**.

To view a list of signing certificates belonging to a user

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, click **Users**, and then click the name of the user whose signing certificate you want to view.
3. In the **Security Credentials** section, you can view the signing certificates and the status of certificates. Users cannot have more than two signing certificates.

Managing a User Signing Certificate (CLI and API)

You can use the AWS CLI or IAM API to manage certificates.

To disable or re-enable a certificate

- [update-signing-certificate](#) (AWS CLI)
- [UpdateSigningCertificate](#) (IAM API)

To list a user's certificates

- [list-signing-certificates](#) (AWS CLI)
- [ListSigningCertificates](#) (IAM API)

To delete a certificate

- [delete-signing-certificate](#) (AWS CLI)

- [DeleteSigningCertificate](#) (IAM API)

Creating an AMI from an Instance Store-Backed Instance

The following procedures are for creating an instance store-backed AMI from an instance store-backed instance. Before you begin, ensure that you've read the [Prerequisites](#) (p. 93).

Topics

- [Creating an AMI from an Instance Store-Backed Amazon Linux Instance](#) (p. 120)
- [Creating an AMI from an Instance Store-Backed Ubuntu Instance](#) (p. 124)

Creating an AMI from an Instance Store-Backed Amazon Linux Instance

This section describes the creation of an AMI from an Amazon Linux instance. The following procedures may not work for instances running other Linux distributions. For Ubuntu-specific procedures, see [Creating an AMI from an Instance Store-Backed Ubuntu Instance](#) (p. 124).

To prepare to use the Amazon EC2 AMI Tools (HVM instances only)

1. The Amazon EC2 AMI tools require GRUB Legacy to boot properly. Use the following command to install GRUB:

```
[ec2-user ~]$ sudo yum install -y grub
```

2. Install the partition management packages with the following command:

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

To create an AMI from an instance store-backed Linux instance

This procedure assumes that you have satisfied the prerequisites in [Prerequisites](#) (p. 93).

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.
 - a. Create a temporary directory on your instance for your credentials as follows:

```
[ec2-user ~]$ mkdir /tmp/cert
```

This enables you to exclude your credentials from the created image.

- b. Copy your X.509 certificate and corresponding private key from your computer to the `/tmp/cert` directory on your instance using a secure copy tool such as [scp](#) (p. 291). The `-i my-private-key.pem` option in the following `scp` command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXyIBH3HXV4ZBEXAMPLE.pem /  
path/to/cert-HKZYKTAIG2ECMXyIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

```
pk-HKZYKTAIG2ECMXIIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
cert-HKZYKTAIG2ECMXIIBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s
00:00
```

Alternatively, because these are plain text files, you can open the certificate and key in a text editor and copy their contents into new files in `/tmp/cert`.

2. Prepare the bundle to upload to Amazon S3 by running the [ec2-bundle-vol \(p. 99\)](#) command from inside your instance. Be sure to specify the `-e` option to exclude the directory where your credentials are stored. By default, the bundle process excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*`, `*.pgp`, `*.jks`, `*/.ssh/authorized_keys`, and `*/.bash_history`. To include all of these files, use the `--no-filter` option. To include some of these files, use the `--include` option.

Important

By default, the AMI bundling process creates a compressed, encrypted collection of files in the `/tmp` directory that represents your root volume. If you do not have enough free disk space in `/tmp` to store the bundle, you need to specify a different location for the bundle to be stored with the `-d /path/to/bundle/storage` option. Some instances have ephemeral storage mounted at `/mnt` or `/media/ephemeral10` that you can use, or you can also [create \(p. 650\)](#), [attach \(p. 654\)](#), and [mount \(p. 655\)](#) a new Amazon EBS volume to store the bundle.

- a. The **ec2-bundle-vol** command needs to run as `root`. For most commands, you can use **sudo** to gain elevated permissions, but in this case, you should run **sudo -E su** to keep your environment variables.

```
[ec2-user ~]$ sudo -E su
```

Note that bash prompt now identifies you as the root user, and that the dollar sign has been replaced by a hash tag, signalling that you are in a root shell:

```
[root ec2-user]#
```

- b. To create the AMI bundle, run the [ec2-bundle-vol \(p. 99\)](#) command with the following parameters:

- c Path and filename for RSA certificate
- k Path and filename for RSA certificate private key
- partition Partition type: `mbr`, `gpt`, or `none`. AMI s from HVM instances will not boot without this.
- r CPU architecture: `i386` or `x86_64`. You can check this by running the `arch` command.
- u Your AWS user account ID
- e Comma-separated list of directories to exclude from the created image.
- d If default directory `/tmp` has insufficient space to accommodate the bundle, this provides the path to a directory with sufficient space.

For more information on this command and its available options, see [ec2-bundle-vol \(p. 99\)](#).

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

The following is a sample command:

```
[root ec2-user]# $EC2_AMITOOL_HOME/bin/ec2-bundle-vol -k /tmp/  
cert/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-  
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64  
-e /tmp/cert --partition gpt
```

It can take a few minutes to create the image. When this command completes, your /tmp (or non-default) directory contains the bundle (image.manifest.xml, plus multiple image.part.xx files).

- c. Exit from the root shell.

```
[root ec2-user]# exit
```

3. (Optional) Edit the block device mappings in the image.manifest.xml file for your AMI. Instance store-backed AMIs can only specify instance store volumes in the block device mapping when the AMI is created, and these mappings are specified in the image.manifest.xml file. For more information, see [Block Device Mapping \(p. 731\)](#).

Note

This step is required only if you wish to add one or more additional instance store volumes on your AMI.

- a. Create a backup of your image.manifest.xml file.

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/  
image.manifest.xml.bak
```

- b. Reformat the image.manifest.xml file so that it is easier to read and edit.

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak >  
sudo /tmp/image.manifest.xml
```

- c. Edit the block device mappings in image.manifest.xml with a text editor. The example below shows a new entry for the *ephemeral1* instance store volume.

```
<block_device_mapping>  
  <mapping>  
    <virtual>ami</virtual>  
    <device>sda</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral0</virtual>  
    <device>sdb</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral1</virtual>  
    <device>sdc</device>  
  </mapping>  
  <mapping>  
    <virtual>root</virtual>  
    <device>/dev/sda1</device>  
  </mapping>  
</block_device_mapping>
```

- d. Save the image.manifest.xml file and exit your text editor.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

4. To upload your bundle to Amazon S3, run the [ec2-upload-bundle](#) (p. 111) command with the following parameters.
 - b
Location of S3 bucket: *my-s3-bucket/bundle_folder/bundle_name*. Note that if the bucket and folder path does not exist, the command creates it.
 - m
Path to `image.manifest.xml`. If you specified a path with `-d /path/to/bundle/storage` in [Step 2](#) (p. 121), use that same path with this parameter.
 - a
Your AWS account access key ID
 - s
Your AWS account secret access key
 - region
If you intend to register your AMI in a region other than US East (N. Virginia), you must specify both the target region with the `--region` option and a bucket path that already exists in the target region or a unique bucket path that can be created in the target region.

For more information on this command and its available options, see [ec2-upload-bundle](#) (p. 111).

The following is a sample command:

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name  
-m /tmp/image.manifest.xml -a your_access_key_id -  
s your_secret_access_key
```

5. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the `/tmp` directory on the instance using the following `rm` command:

Note

If you specified a path with the `-d /path/to/bundle/storage` option in [Step 2](#) (p. 121), use that same path below, instead of `/tmp`.

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

6. To register your AMI , run the [register-image](#) AWS CLI command with the following parameters.
 - image-location
my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml
 - name
A name for the AMI
 - virtualization-type
Possible values are `hvm` and `paravirtual`.
 - region
If you previously specified a region for the [ec2-upload-bundle](#) (p. 111) command, specify that region again for this command.

For more information on this command and its available options, see [register-image](#) in the *AWS Command Line Interface Reference*.

The following is a sample command:

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance

```
[ec2-user ~]$ aws ec2 register-image --image-location my-s3-  
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --  
virtualization-type hvm
```

Creating an AMI from an Instance Store-Backed Ubuntu Instance

This section describes the creation of an AMI from an Ubuntu Linux instance. The following procedures may not work for instances running other Linux distributions. For procedures specific to Amazon Linux, see [Creating an AMI from an Instance Store-Backed Amazon Linux Instance \(p. 120\)](#).

To prepare to use the Amazon EC2 AMI Tools (HVM instances only)

The Amazon EC2 AMI tools require GRUB Legacy to boot properly. However, Ubuntu is configured to use GRUB 2. You must check to see that your instance uses GRUB Legacy, and if not, you need to install and configure it.

HVM instances also require partitioning tools to be installed for the AMI tools to work properly.

1. GRUB Legacy (version 0.9x or less) must be installed on your instance. Check to see if GRUB Legacy is present and install it if necessary.
 - a. Check the version of your GRUB installation.

```
ubuntu:~$ grub-install --version  
grub-install (GRUB) 1.99-21ubuntu3.10
```

In this example, the GRUB version is greater than 0.9x, so GRUB Legacy must be installed. Proceed to [Step 1.b \(p. 124\)](#). If GRUB Legacy is already present, you can skip to [Step 2 \(p. 124\)](#).

- b. Install the `grub` package using the following command.

```
ubuntu:~$ sudo apt-get install -y grub
```

Verify that your instance is using GRUB Legacy.

```
ubuntu:~$ grub --version  
grub (GNU GRUB 0.97)
```

2. Install the following partition management packages using the package manager for your distribution.
 - `gdisk` (some distributions may call this package `gptfdisk` instead)
 - `kpartx`
 - `parted`

Use the following command.

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx parted
```

3. Check the kernel parameters for your instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

```
ubuntu:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-
aee7-72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

Note the options following the kernel and root device parameters: `ro`, `console=ttyS0`, and `xen_emul_unplug=unnecessary`. Your options may differ.

4. Check the kernel entries in `/boot/grub/menu.lst`.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
  console=hvc0
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
  single
kernel /boot/memtest86+.bin
```

Note that the `console` parameter is pointing to `hvc0` instead of `ttyS0` and that the `xen_emul_unplug=unnecessary` parameter is missing. Again, your options may differ.

5. Edit the `/boot/grub/menu.lst` file with your favorite text editor (such as `vim` or `nano`) to change the console and add the parameters you identified earlier to the boot entries.

```
title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root           (hd0)
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
  ro console=ttyS0 xen_emul_unplug=unnecessary
initrd        /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery
mode)
root           (hd0)
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
  ro single console=ttyS0 xen_emul_unplug=unnecessary
initrd        /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, memtest86+
root           (hd0)
kernel         /boot/memtest86+.bin
```

6. Verify that your kernel entries now contain the correct parameters.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
  console=ttyS0 xen_emul_unplug=unnecessary
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
  single console=ttyS0 xen_emul_unplug=unnecessary
kernel /boot/memtest86+.bin
```

7. (For Ubuntu 14.04 and later only) Starting with Ubuntu 14.04, instance store backed Ubuntu AMIs use a GPT partition table and a separate EFI partition mounted at `/boot/efi`. The `ec2-bundle-vol` command will not bundle this boot partition, so you need to comment out the `/etc/fstab` entry for the EFI partition as shown in the following example.

```
LABEL=cloudimg-rootfs /          ext4  defaults    0 0
#LABEL=UEFI           /boot/efi  vfat  defaults    0 0
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Creating an AMI from an Instance Store-Backed Instance

```
/dev/xvdb /mnt auto defaults,nobootwait,comment=cloudconfig 0  
2
```

To create an AMI from an instance store-backed Linux instance

This procedure assumes that you have satisfied the prerequisites in [Prerequisites \(p. 93\)](#).

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.
 - a. Create a temporary directory on your instance for your credentials as follows:

```
ubuntu:~$ mkdir /tmp/cert
```

This enables you to exclude your credentials from the created image.

- b. Copy your X.509 certificate and private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as [scp \(p. 291\)](#). The `-i my-private-key.pem` option in the following `scp` command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXIYBH3HXV4ZBEXAMPLE.pem /  
path/to/cert-HKZYKTAIG2ECMXIYBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXIYBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00  
cert-HKZYKTAIG2ECMXIYBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s  
00:00
```

Alternatively, because these are plain text files, you can open the certificate and key in a text editor and copy their contents into new files in `/tmp/cert`.

2. Prepare the bundle to upload to Amazon S3 by running the [ec2-bundle-vol \(p. 99\)](#) command from inside your instance. Be sure to specify the `-e` option to exclude the directory where your credentials are stored. By default, the bundle process excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*`, `*.gpg`, `*.jks`, `*/.ssh/authorized_keys`, and `*/*.bash_history`. To include all of these files, use the `--no-filter` option. To include some of these files, use the `--include` option.

Important

By default, the AMI bundling process creates a compressed, encrypted collection of files in the `/tmp` directory that represents your root volume. If you do not have enough free disk space in `/tmp` to store the bundle, you need to specify a different location for the bundle to be stored with the `-d /path/to/bundle/storage` option. Some instances have ephemeral storage mounted at `/mnt` or `/media/ephemeral0` that you can use, or you can also [create \(p. 650\)](#), [attach \(p. 654\)](#), and [mount \(p. 655\)](#) a new Amazon EBS volume to store the bundle.

- a. The `ec2-bundle-vol` command needs to run as `root`. For most commands, you can use `sudo` to gain elevated permissions, but in this case, you should run `sudo -E su` to keep your environment variables.

```
ubuntu:~$ sudo -E su
```

Note that bash prompt now identifies you as the root user, and that the dollar sign has been replaced by a hash tag, signalling that you are in a root shell:

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

```
root@ubuntu: #
```

- b. To create the AMI bundle, run the [ec2-bundle-vol](#) (p. 99) command with the following parameters.

-c
Path and filename for RSA certificate

-k
Path and filename for RSA certificate private key

--partition
Partition type: `mbr`, `gpt`, or `none`. For Ubuntu 14.04 and later HVM instances, add the `--partition mbr` flag to bundle the boot instructions properly; otherwise, your newly-created AMI will not boot.

-r
CPU architecture: `i386` or `x86_64`. You can check this by running the `arch` command.

-u
Your AWS user account ID

-e
Comma-separated list of directories to exclude from the created image.

-d
If default directory `/tmp` has insufficient space to accommodate the bundle, this provides the path to a directory with sufficient space.

For more information on this command and its available options, see [ec2-bundle-vol](#) (p. 99).

The following is a sample command:

```
root@ubuntu: # $EC2_AMITOOL_HOME/bin/ec2-bundle-vol -k /tmp/  
cert/pk-HKZYKTAIG2ECMXyIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-  
HKZYKTAIG2ECMXyIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64  
-e /tmp/cert --partition gpt
```

It can take a few minutes to create the image. When this command completes, your `tmp` directory contains the bundle (`image.manifest.xml`, plus multiple `image.part.xxx` files).

- c. Exit from the `root` shell.

```
root@ubuntu: # exit
```

3. (Optional) Edit the block device mappings in the `image.manifest.xml` file for your AMI. Instance store-backed AMIs can only specify instance store volumes in the block device mapping when the AMI is created, and these mappings are specified in the `image.manifest.xml` file. For more information, see [Block Device Mapping](#) (p. 731).

Note

This step is required only if you wish to add one or more additional instance store volumes on your AMI.

- a. Create a backup of your `image.manifest.xml` file.

```
ubuntu: ~$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. Reformat the `image.manifest.xml` file so that it is easier to read and edit.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Creating an AMI from an
Instance Store-Backed Instance**

```
ubuntu:~$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/  
image.manifest.xml
```

- c. Edit the block device mappings in `image.manifest.xml` with a text editor. The example below shows a new entry for the `ephemeral1` instance store volume.

```
<block_device_mapping>  
  <mapping>  
    <virtual>ami</virtual>  
    <device>sda</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral0</virtual>  
    <device>sdb</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral1</virtual>  
    <device>sdc</device>  
  </mapping>  
  <mapping>  
    <virtual>root</virtual>  
    <device>/dev/sda1</device>  
  </mapping>  
</block_device_mapping>
```

- d. Save the `image.manifest.xml` file and exit your text editor.
4. To upload your bundle to Amazon S3, run the `ec2-upload-bundle` (p. 111) command with the following parameters.
- b Location of S3 bucket: `my-s3-bucket/bundle_folder/bundle_name`. Note that if the bucket and folder path does not exist, the command creates it.
 - m Path to `image.manifest.xml`. If you specified a path with `-d /path/to/bundle/storage` in [Step 2](#) (p. 126), use that same path in this parameter.
 - a Your AWS account access key ID
 - s Your AWS account secret access key
 - region If you intend to register your AMI in a region other than US East (N. Virginia), you must specify both the target region with the `--region` option and a bucket path that already exists in the target region or a unique bucket path that can be created in the target region.

For more information on this command and its available options, see [ec2-upload-bundle](#) (p. 111).

The following is a sample command:

```
ubuntu:~$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /  
tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

5. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the `/tmp` directory on the instance using the following `rm` command:

Note

If you specified a path with the `-d /path/to/bundle/storage` option in [Step 2 \(p. 126\)](#), use that same path below, instead of `/tmp`.

```
ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

- To register your AMI, run the [register-image](#) AWS CLI command with the following parameters.

Path to manifest

`my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml`

`-n`

A name for the AMI

`--virtualization-type`

Possible values are `hvm` and `paravirtual`.

`--region`

If you previously specified a region for the [ec2-upload-bundle \(p. 111\)](#) command, specify that region again for this command.

For more information on this command and its available options, see [register-image](#) in the *AWS Command Line Interface Reference*.

The following is a sample command:

```
ubuntu:~$ aws ec2 register-image my-s3-bucket/bundle_folder/bundle_name/  
image.manifest.xml --name AMI_name --virtualization-type hvm
```

- (For Ubuntu 14.04 and later only) Uncomment the EFI entry in `/etc/fstab`; otherwise, your running instance will not be able to reboot.

Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI

You can convert an instance store-backed Linux AMI that you own to an Amazon EBS-backed Linux AMI.

Important

You can't convert an instance store-backed Windows AMI to an Amazon EBS-backed Windows AMI and you cannot convert an AMI that you do not own.

To convert an instance store-backed AMI to an Amazon EBS-backed AMI

- Launch an Amazon Linux instance from an Amazon EBS-backed AMI. For more information, see [Launching an Instance \(p. 279\)](#). Amazon Linux instances have the AWS CLI and AMI tools pre-installed.
- Upload the X.509 private key that you used to bundle your instance store-backed AMI to your instance. We use this key to ensure that only you and Amazon EC2 can access your AMI.
 - Create a temporary directory on your instance for your X.509 private key as follows:

```
[ec2-user ~]$ mkdir /tmp/cert
```

- Copy your X.509 private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as [scp \(p. 291\)](#). The `my-private-key` parameter

Amazon Elastic Compute Cloud User Guide for Linux Instances Converting to an Amazon EBS-Backed AMI

in the following command is the private key you use to connect to your instance with SSH. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
```

3. Set environment variables for your AWS access key and secret key.

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id  
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

4. Prepare an Amazon EBS volume for your new AMI.
 - a. Create an empty Amazon EBS volume in the same Availability Zone as your instance using the [create-volume](#) command. Note the volume ID in the command output.

Important

This Amazon EBS volume must be the same size or larger than the original instance store root volume.

```
[ec2-user ~]$ aws ec2 create-volume --size 10 --region us-west-2 --  
availability-zone us-west-2b
```

- b. Attach the volume to your Amazon EBS-backed instance using the [attach-volume](#) command.

```
[ec2-user ~]$ aws ec2 attach-volume --volumeid volume_id --instance-  
id instance_id --device /dev/sdb --region us-west-2
```

5. Create a folder for your bundle.

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. Download the bundle for your instance store-based AMI to `/tmp/bundle` using the [ec2-download-bundle](#) (p. 106) command.

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-  
bucket/bundle_folder/bundle_name -m image.manifest.xml -a  
$AWS_ACCESS_KEY -s $AWS_SECRET_KEY --privatekey /path/to/pk-  
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. Reconstitute the image file from the bundle using the [ec2-unbundle](#) (p. 110) command.

- a. Change directories to the bundle folder.

```
[ec2-user ~]$ cd /tmp/bundle/
```

- b. Run the [ec2-unbundle](#) (p. 110) command.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. Copy the files from the unbundled image to the new Amazon EBS volume.

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Converting to an Amazon EBS-Backed AMI**

9. Probe the volume for any new partitions that were unbundled.

```
[ec2-user bundle]$ sudo partprobe /dev/sdb
```

10. List the block devices to find the device name to mount.

```
[ec2-user bundle]$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/sda      202:0   0    8G  0 disk
##/dev/sda1  202:1   0    8G  0 part /
/dev/sdb      202:80   0   10G  0 disk
##/dev/sdb1  202:81   0   10G  0 part
```

In this example, the partition to mount is `/dev/sdb1`, but your device name will likely be different. If your volume is not partitioned, then the device to mount will be similar to `/dev/sdb` (without a device partition trailing digit).

11. Create a mount point for the new Amazon EBS volume and mount the volume.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. Open the `/etc/fstab` file on the EBS volume with your favorite text editor (such as `vim` or `nano`) and remove any entries for instance store (ephemeral) volumes. Because the Amazon EBS volume is mounted on `/mnt/ebs`, the `fstab` file is located at `/mnt/ebs/etc/fstab`.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/      /          ext4        defaults,noatime 1 1
tmpfs        /dev/shm   tmpfs       defaults          0 0
devpts       /dev/pts   devpts      gid=5,mode=620   0 0
sysfs        /sys       sysfs       defaults          0 0
proc         /proc      proc        defaults          0 0
/dev/sdb     /media/ephemeral0 auto
defaults,comment=cloudconfig 0 2
```

In this example, the last line should be removed.

13. Unmount the volume and detach it from the instance.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id volume_id --
region us-west-2
```

14. Create an AMI from the new Amazon EBS volume as follows.

- a. Create a snapshot of the new Amazon EBS volume.

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --
description "your_snapshot_description" --volume-id volume_id
```

- b. Check to see that your snapshot is complete.

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --
snapshot-id snapshot_id
```

- c. Identify the processor architecture, virtualization type, and the kernel image (`aki`) used on the original AMI with the `describe-images` command. You need the AMI ID of the original instance store-backed AMI for this step.

```
[ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-id ami-id --output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be
amazon available public machine aki-fc8f11cc instance-store
paravirtual xen
```

In this example, the architecture is `x86_64` and the kernel image ID is `aki-fc8f11cc`. Use these values in the following step. If the output of the above command also lists an `ari` ID, take note of that as well.

- d. Register your new AMI with the snapshot ID of your new Amazon EBS volume and the values from the previous step. If the previous command output listed an `ari` ID, include that in the following command with `--ramdisk-id ari_id`.

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --name your_new_ami_name --block-device-mappings Ebs={SnapshotId=snapshot_id} --virtualization-type hvm --architecture x86_64 --kernel-id aki-fc8f11cc
```

15. (Optional) After you have tested that you can launch an instance from your new AMI, you can delete the Amazon EBS volume that you created for this procedure.

```
$ aws ec2 delete-volume --volume-id volume_id
```

AMIs with Encrypted Snapshots

AMIs that are backed by Amazon EBS snapshots can take advantage of Amazon EBS encryption. Snapshots of both data and root volumes can be encrypted and attached to an AMI.

EC2 instances with encrypted volumes are launched from AMIs in the same way as other instances.

The `CopyImage` action can be used to create an AMI with encrypted snapshots from an AMI with unencrypted snapshots. By default, `CopyImage` preserves the encryption status of source snapshots when creating destination copies. However, you can configure the parameters of the copy process to also encrypt the destination snapshots.

Snapshots can be encrypted with either your default AWS Key Management Service customer master key (CMK), or with a custom key that you specify. You must in all cases have permission to use the selected key. If you have an AMI with encrypted snapshots, you can choose to re-encrypt them with a different encryption key as part of the `CopyImage` action. `CopyImage` accepts only one key at a time and encrypts all of an image's snapshots (whether root or data) to that key. However, it is possible to manually build an AMI with snapshots encrypted to multiple keys.

Support for creating AMIs with encrypted snapshots is accessible through the Amazon EC2 console, Amazon EC2 API, or the AWS CLI.

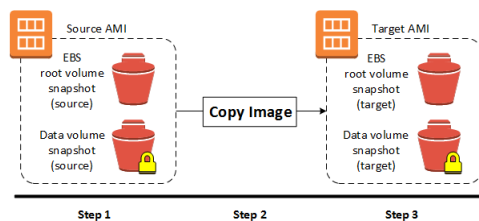
The encryption parameters of `CopyImage` are available in all regions where AWS KMS is available.

AMI Scenarios Involving Encrypted EBS Snapshots

You can copy an AMI and simultaneously encrypt its associated EBS snapshots using the AWS Management Console or the command line.

Copying an AMI with an Encrypted Data Snapshot

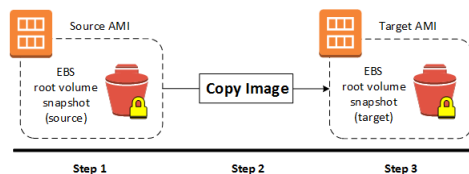
In this scenario, an EBS-backed AMI has an unencrypted root snapshot and an encrypted data snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of each snapshot is preserved, so that the destination AMI, in step 3, is also backed by an unencrypted root snapshot and an encrypted data snapshot. Though the snapshots contain the same data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 135\)](#).

Copying an AMI Backed by An Encrypted Root Snapshot

In this scenario, an Amazon EBS-backed AMI has an encrypted root snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of the snapshot is preserved, so that the destination AMI, in step 3, is also backed by an encrypted root snapshot. Though the root snapshots contain identical system data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



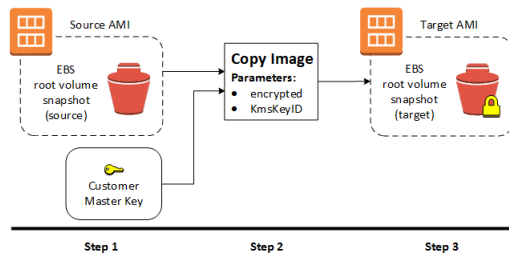
You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 135\)](#).

Creating an AMI with Encrypted Root Snapshot from an Unencrypted AMI

In this scenario, an Amazon EBS-backed AMI has an unencrypted root snapshot, shown in step 1, and an AMI is created with an encrypted root snapshot, shown in step 3. The `CopyImage` action in step 2 is invoked with two encryption parameters, including the choice of a CMK. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the

Amazon Elastic Compute Cloud User Guide for Linux Instances AMI Scenarios Involving Encrypted EBS Snapshots

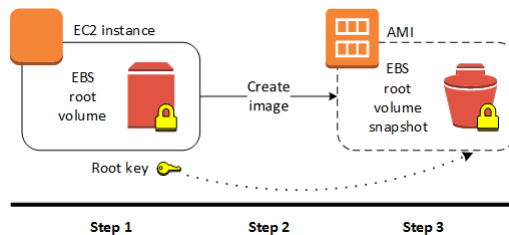
same data as the source snapshot, but encrypted using the specified key. You will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a copy and encrypt operation such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 135\)](#).

Creating an AMI with an Encrypted Root Snapshot from a Running Instance

In this scenario, an AMI is created from a running EC2 instance. The running instance in step 1 has an encrypted root volume, and the created AMI in step 3 has a root snapshot encrypted to the same key as the source volume. The `CreateImage` action has exactly the same behavior whether or not encryption is present.



You can create an AMI from a running Amazon EC2 instance (with or without encrypted volumes) using either the Amazon EC2 console or the command line. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#).

Creating an AMI with Unique CMKs for Each Encrypted Snapshot

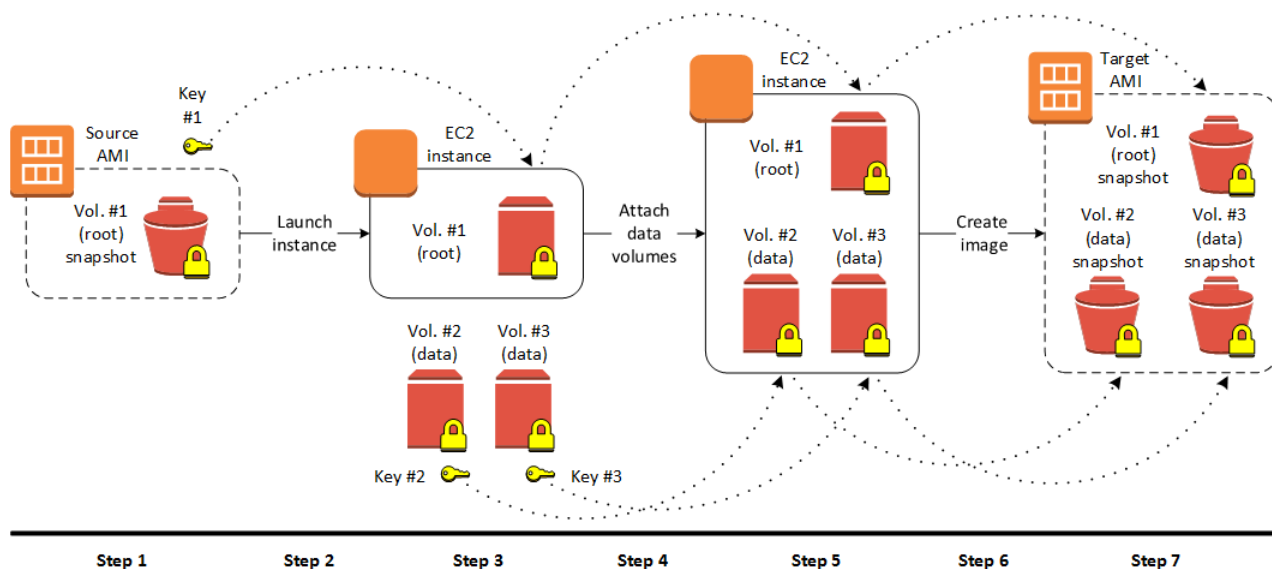
This scenario starts with an AMI backed by a root-volume snapshot (encrypted to key #1), and finishes with an AMI that has two additional data-volume snapshots attached (encrypted to key #2 and key #3). The `CopyImage` action cannot apply more than one encryption key in a single operation. However, you can create an AMI from an instance that has multiple attached volumes encrypted to different keys. The resulting AMI has snapshots encrypted to those keys and any instance launched from this new AMI also has volumes encrypted to those keys.

The steps of this example procedure correspond to the following diagram.

1. Start with the source AMI backed by vol. #1 (root) snapshot, which is encrypted with key #1.
2. Launch an EC2 instance from the source AMI.
3. Create EBS volumes vol. #2 (data) and vol. #3 (data), encrypted to key #2 and key #3 respectively.
4. Attach the encrypted data volumes to the EC2 instance.

Amazon Elastic Compute Cloud User Guide for Linux Instances Copying an AMI

5. The EC2 instance now has an encrypted root volume as well as two encrypted data volumes, all using different keys.
6. Use the `CreateImage` action on the EC2 instance.
7. The resulting target AMI contains encrypted snapshots of the three EBS volumes, all using different keys.



You can carry out this procedure using either the Amazon EC2 console or the command line. For more information, see the following topics:

- [Launch Your Instance](#) (p. 278)
- [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89).
- [Amazon EBS Volumes](#) (p. 636)
- [AWS Key Management](#) in the *AWS Key Management Service Developer Guide*

Copying an AMI

You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the command line, or the Amazon EC2 API, all of which support the `CopyImage` action. Both Amazon EBS-backed AMIs and instance store-backed AMIs can be copied.

Copying a source AMI results in an identical but distinct target AMI with its own unique identifier. In the case of an Amazon EBS-backed AMI, each of its backing snapshots is, by default, copied to an identical but distinct target snapshot. (The one exception is when you choose to encrypt the snapshot, as described below.) The source AMI can be changed or deregistered with no effect on the target AMI. The reverse is also true.

There are no charges for copying an AMI. However, standard storage and data transfer rates apply.

AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI.

Copying an AMI You Own

You can copy any AMI that belongs to your AWS account using the `CopyImage` action. This includes AMIs with encrypted snapshots and encrypted AMIs.

Copying an AMI Across AWS Accounts

You can copy an AMI across AWS accounts. This includes AMIs with encrypted snapshots, but does not include encrypted AMIs.

The owner of the account must grant read permissions on the storage that backs the AMI, whether it is an associated EBS snapshot (for an Amazon EBS-backed AMI) or an associated Amazon S3 bucket (for an instance-store-backed AMI). To allow other accounts to copy your AMIs, you must grant read permissions on your associated snapshot or bucket using the Amazon EBS or Amazon S3 access management tools.

When an AMI is copied, the owner of the source AMI is charged standard Amazon EBS or Amazon S3 transfer fees, and the owner of the target AMI is charged for storage in the destination region.

Limits

- You can't copy an encrypted AMI between accounts. Instead, if the underlying snapshot and encryption key have been shared with you, you can copy the snapshot to another account while re-encrypting it with a key of your own, and then register this privately owned snapshot as a new AMI.
- You can't directly copy an AMI that has a `billingProduct` code associated with it. This includes Windows AMIs and other AMIs from the AWS Marketplace that are owned and shared by another AWS account. To create a private copy of an AMI that has a `billingProduct` code associated with it, we recommend the following procedure:

1. Launch an EC2 instance in the target account using the shared AMI.
2. Create an image from the new instance.

The result will be a private AMI that you own and can customize like any other AMI you own. For example, if you have created a private copy of an EBS-backed AMI, you can use `CopyImage` to create an AMI with an encrypted root volume. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#).

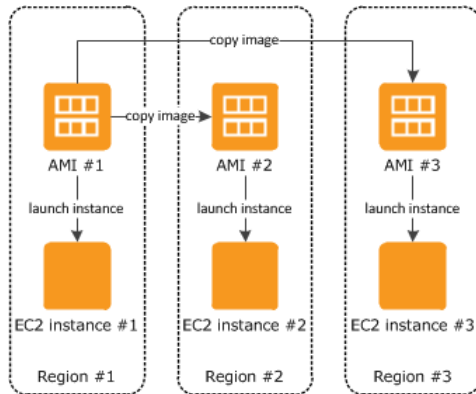
Copying an AMI Across Regions

Copying an AMI across geographically diverse regions provides the following benefits:

- **Consistent global deployment:** Copying an AMI from one region to another enables you to launch consistent instances based from the same AMI into different regions.
- **Scalability:** You can more easily design and build world-scale applications that meet the needs of your users, regardless of their location.
- **Performance:** You can increase performance by distributing your application, as well as locating critical components of your application in closer proximity to your users. You can also take advantage of region-specific features, such as instance types or other AWS services.
- **High availability:** You can design and deploy applications across AWS regions, to increase availability.

The following diagram shows the relations among a source AMI and two copied AMIs in different regions, as well as the EC2 instances launched from each. When you launch an instance from an AMI, it resides in the same region where the AMI resides. If you make changes to the source AMI and want

those changes to be reflected in the AMIs in the target regions, you must recopy the source AMI to the target regions.



When you first copy an instance store-backed AMI to a region, we create an Amazon S3 bucket for the AMIs copied to that region. All instance store-backed AMIs that you copy to that region are stored in this bucket. The names of these buckets have the following format: amis-for-*account*-in-*region-hash*. For example: amis-for-123456789012-in-us-west-2-yhjmxvp6.

Note

Destination regions are limited to 50 concurrent AMI copies at a time, with no more than 25 of those coming from a single source region. To request an increase to this limit, see [Amazon EC2 Service Limits \(p. 760\)](#).

Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination region may still use the resources from the source region, which can impact performance and cost.

Copying to Encrypt

Encrypting during copying applies only to Amazon EBS-backed AMIs. Because an instance-store-backed AMIs does not rely on snapshots, the `CopyImage` action cannot be used to change its encryption status.

The `CopyImage` action can also be used to create a new AMI backed by encrypted Amazon EBS snapshots. If you invoke encryption while copying an AMI, each snapshot taken of its associated Amazon EBS volumes—including the root volume—will be encrypted using a key that you specify. For more information about using AMIs with encrypted snapshots, see [AMIs with Encrypted Snapshots \(p. 132\)](#).

By default, the backing snapshot of an AMI will be copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot will result in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying it will result in a target snapshot encrypted to the specified key. Copying an AMI backed by multiple snapshots preserves the source encryption status in each target snapshot. For more information about copying AMIs with multiple snapshots, see [AMIs with Encrypted Snapshots \(p. 132\)](#).

The following table shows encryption support for various scenarios. Note that while it is possible to copy an unencrypted snapshot to yield an encrypted snapshot, you cannot copy an encrypted snapshot to yield an unencrypted one.

Scenario	Description	Supported
1	Unencrypted-to-unencrypted	Yes

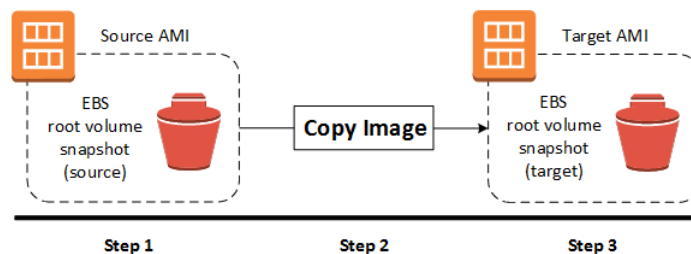
Scenario	Description	Supported
2	Encrypted-to-encrypted	Yes
3	Unencrypted-to-encrypted	Yes
4	Encrypted-to-unencrypted	No

AMI Copying Scenarios

This section describes basic scenarios for copying AMIs and provides copy procedures using the Amazon EC2 console and the command line.

Copy an unencrypted source AMI to an unencrypted target AMI

In the simplest case, a copy of an AMI with an unencrypted single backing snapshot is created in the specified geographical region (not shown).

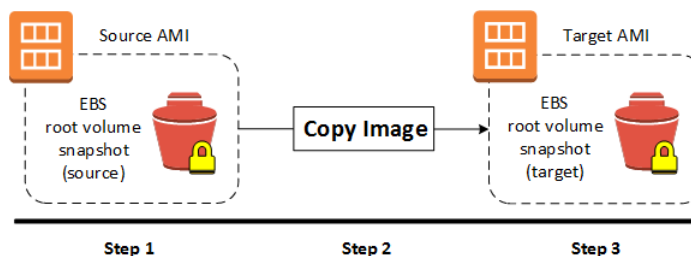


Note

Although the diagram above shows an AMI with a single backing snapshot, the `CopyImage` action also works for AMIs with multiple snapshots. The encryption status of each snapshot is preserved. This means that an unencrypted snapshot in the source AMI will cause an unencrypted snapshot to be created in the target AMI, and an encrypted snapshot in the source AMI will cause an encrypted snapshot to be created in the target AMI.

Copy an encrypted source AMI to an encrypted target AMI

Although this scenario involves encrypted snapshots, it is functionally equivalent to the previous scenario.

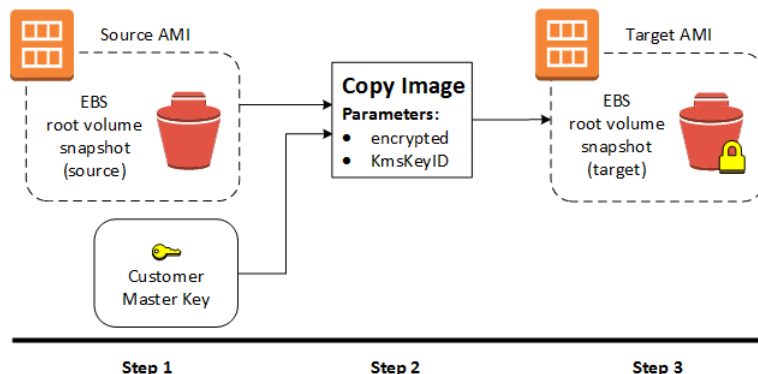


Note

If you apply encryption while copying a multi-snapshot AMI, all of the target snapshots are encrypted using the specified key or the default key if none is specified. For information about creating an AMI with multiple snapshots encrypted to multiple keys, see [AMIs with Encrypted Snapshots](#) (p. 132).

Copy an unencrypted source AMI to an encrypted target AMI

In this last scenario, the `CopyImage` action changes the encryption status of the destination image, for instance, by encrypting an unencrypted snapshot, or re-encrypting an encrypted snapshot with a different key. To apply encryption during the copy, you must supply encryption parameters: an encryption flag and a key. Volumes created from the target snapshot are accessible only if you supply this key. For more information about supported encryption scenarios for AMIs, see [AMIs with Encrypted Snapshots](#) (p. 132).



Copying an AMI Using the Console or Command Line

The steps in the following procedure correspond to the three steps in each scenario diagram. Apart from the configuration of encryption options, the procedure for implementing the `CopyImage` action is identical in all cases.

To copy an AMI using the console

1. Create or obtain an AMI backed by an Amazon EBS snapshot. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89). A wide variety of AWS-supplied AMIs are available through the Amazon EC2 console.

From the console navigation bar, select the region that contains the AMI you wish to copy. In the navigation pane, expand **Images** and select **AMIs** to display the list of AMIs available to you in the selected region.

2. Select the AMI to copy and choose **Actions** and **Copy AMI**.

In the **AMI Copy** page, set the following fields and choose **Copy AMI**:

- **Destination region:** Choose the region into which to copy the AMI.
 - **Name:** Provide a name for the new AMI. You may want to include operating system information in the name, as we do not provide this information when displaying details about the AMI.
 - **Description:** By default, the description includes information about the source AMI so that you can distinguish a copy from its original. You can change this description as needed.
 - **Encryption:** Select this field to encrypt the target Amazon EBS snapshots, or to re-encrypt them using a different key.
 - **Master Key:** The KMS key that will be used to encrypt the target Amazon EBS snapshots if **Encryption** has been chosen.
3. We display a confirmation page to let you know that the copy operation has been initiated and to provide you with the ID of the new AMI.

To check on the progress of the copy operation immediately, follow the provided link. To check on the progress later, choose **Done**, and then when you are ready, use the navigation bar to switch to the target region (if applicable) and locate your AMI in the list of AMIs.

The initial status of the target AMI is `pending` and the operation is complete when the status is `available`.

To copy an AMI using the command line

Copying an AMI using the command line requires that you specify both the source and destination regions. You specify the source region using the `--source-region` parameter. For the destination region, you have two options:

- Use the `--region` parameter.
- Set an environmental variable. For more information, see [Configuring the AWS Command Line Interface](#).

When you encrypt a target snapshot during copying, you will need to supply two additional parameters:

- A Boolean, `--encrypted`
- A string, `--kms-key-id`, providing the master encryption key ID

You can copy an AMI using one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `copy-image` (AWS CLI)
- `Copy-EC2Image` (AWS Tools for Windows PowerShell)

Stopping a Pending AMI Copy Operation

You can stop a pending AMI copy using the AWS Management Console or the command line.

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the destination region from the region selector.
3. In the navigation pane, choose **AMIs**.
4. Select the AMI to stop copying and choose **Actions** and **Deregister**.
5. When asked for confirmation, choose **Continue**.

To stop an AMI copy operation using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `deregister-image` (AWS CLI)
- `Unregister-EC2Image` (AWS Tools for Windows PowerShell)

Deregistering Your AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI. You'll continue to incur usage costs for these instances. Therefore, if you are finished with these instances, you should terminate them.

The procedure that you'll use to clean up your AMI depends on whether it is backed by Amazon EBS or instance store. (Note that the only Windows AMIs that can be backed by instance store are those for Windows Server 2003.)

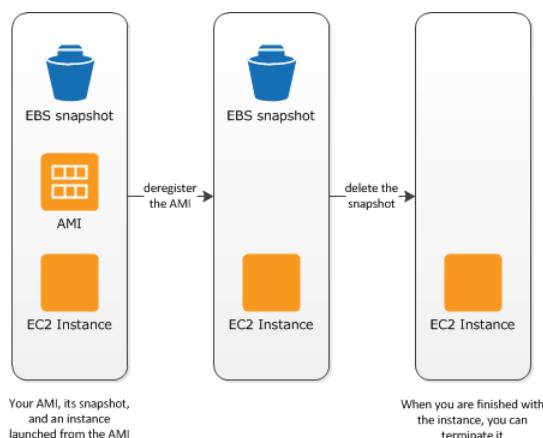
Contents

- [Cleaning Up Your Amazon EBS-Backed AMI \(p. 141\)](#)
- [Cleaning Up Your Instance Store-Backed AMI \(p. 142\)](#)

Cleaning Up Your Amazon EBS-Backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that was created for the root volume of the instance during the AMI creation process. You'll continue to incur storage costs for this snapshot. Therefore, if you are finished with the snapshot, you should delete it.

The following diagram illustrates the process for cleaning up your Amazon EBS-backed AMI.



To clean up your Amazon EBS-backed AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**. Select the AMI, and take note of its ID — this can help you find the correct snapshot in the next step. Choose **Actions**, and then **Deregister**. When prompted for confirmation, choose **Continue**.

The AMI status is now `unavailable`.

Note

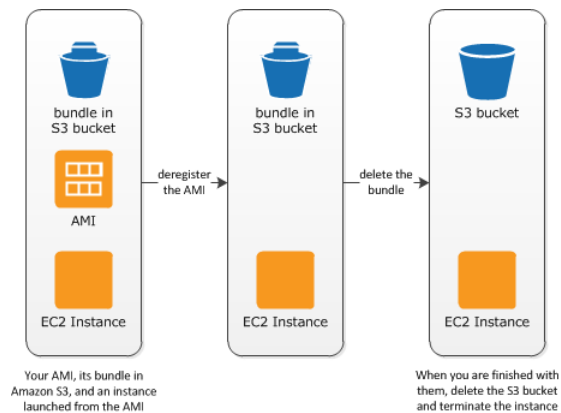
It may take a few minutes before the console changes the status from `available` to `unavailable`, or removes the AMI from the list altogether. Choose **Refresh** to refresh the status.

3. In the navigation pane, choose **Snapshots**, and select the snapshot (look for the AMI ID in the **Description** column). Choose **Actions**, and then choose **Delete Snapshot**. When prompted for confirmation, choose **Yes, Delete**.
4. (Optional) If you are finished with an instance that you launched from the AMI, terminate it. In the navigation pane, choose **Instances**. Select the instance, choose **Actions**, then **Instance State**, and then **Terminate**. When prompted for confirmation, choose **Yes, Terminate**.

Cleaning Up Your Instance Store-Backed AMI

When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.

The following diagram illustrates the process for cleaning up your instance store-backed AMI.



To clean up your instance store-backed AMI

1. Deregister the AMI using the `deregister-image` command as follows.

```
aws ec2 deregister-image --image-id ami_id
```

The AMI status is now `unavailable`.

2. Delete the bundle in Amazon S3 using the `ec2-delete-bundle` (p. 104) (AMI tools) command as follows.

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s your_secret_access_key -p image
```

3. (Optional) If you are finished with an instance that you launched from the AMI, you can terminate it using the `terminate-instances` command as follows.

```
aws ec2 terminate-instances --instance-ids instance_id
```

4. (Optional) If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, choose **Actions**, and then choose **Delete**.

Amazon Linux

Amazon Linux is provided by Amazon Web Services (AWS). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2. It also includes packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. AWS provides ongoing security and maintenance updates to all instances running Amazon Linux.

Note

The Amazon Linux AMI repository structure is configured to deliver a continuous flow of updates that allow you to roll from one version of the Amazon Linux AMI to the next. To lock existing instances to their current version, see [Repository Configuration \(p. 146\)](#).

To launch an Amazon Linux instance, use an Amazon Linux AMI. AWS provides Amazon Linux AMIs to Amazon EC2 users at no additional cost.

Topics

- [Finding the Amazon Linux AMI \(p. 143\)](#)
- [Launching and Connecting to an Amazon Linux Instance \(p. 143\)](#)
- [Identifying Amazon Linux AMI Images \(p. 143\)](#)
- [Included AWS Command Line Tools \(p. 144\)](#)
- [cloud-init \(p. 145\)](#)
- [Repository Configuration \(p. 146\)](#)
- [Adding Packages \(p. 147\)](#)
- [Accessing Source Packages for Reference \(p. 147\)](#)
- [Developing Applications \(p. 148\)](#)
- [Instance Store Access \(p. 148\)](#)
- [Product Life Cycle \(p. 148\)](#)
- [Security Updates \(p. 148\)](#)
- [Support \(p. 149\)](#)

Finding the Amazon Linux AMI

For a list of the latest Amazon Linux AMIs, see [Amazon Linux AMIs](#).

Launching and Connecting to an Amazon Linux Instance

After locating your desired AMI, note the AMI ID. You can use the AMI ID to launch and then connect to your instance.

Amazon Linux does not allow remote root SSH by default. Also, password authentication is disabled to prevent brute-force password attacks. To enable SSH logins to an Amazon Linux instance, you must provide your key pair to the instance at launch. You must also set the security group used to launch your instance to allow SSH access. By default, the only account that can log in remotely using SSH is `ec2-user`; this account also has `sudo` privileges. If you want to enable remote root log in, please be aware that it is less secure than relying on key pairs and a secondary user.

For information about launching and using your Amazon Linux instance, see [Launch Your Instance \(p. 278\)](#). For information about connecting to your Amazon Linux instance, see [Connecting to Your Linux Instance \(p. 290\)](#).

Identifying Amazon Linux AMI Images

Each image contains a unique `/etc/image-id` that identifies the AMI. This file contains information about the image.

The following is an example of the `/etc/image-id` file:

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2016.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2016.03.3.x86_64.ext4.gpt"
image_stamp="72d4-6a2a"
image_date="20160621201058"
recipe_name="amzn ami"
recipe_id="0275ad70-5d13-1d50-a976-482a-73b6-2208-14cb7bc0"
```

The `image_name`, `image_version`, and `image_arch` items come from the build recipe that Amazon used to construct the image. The `image_stamp` is simply a unique random hex value generated during image creation. The `image_date` item is in YYYYMMDDhhmmss format, and is the UTC time of image creation. The `recipe_name` and `recipe_id` refer to the name and ID of the build recipe Amazon used to construct the image, which identifies the current running version of Amazon Linux. This file will not change as you install updates from the **yum** repository.

Amazon Linux contains an `/etc/system-release` file that specifies the current release that is installed. This file is updated through **yum** and is part of the `system-release` RPM.

The following is an example of an `/etc/system-release` file:

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2016.03
```

Amazon Linux also contains a machine readable version of the `/etc/system-release` file found in `/etc/system-release-cpe` and follows the CPE specification from MITRE ([CPE](#)).

Included AWS Command Line Tools

The following popular command line tools for AWS integration and usage have been included in Amazon Linux or in the default repositories:

- `aws-amitools-ec2`
- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-ec2`
- `aws-apitools-elb`
- `aws-apitools-iam`
- `aws-apitools-mon`
- `aws-apitools-rds`
- `aws-cfn-bootstrap`
- `aws-cli`
- `aws-scripts-ses`

Note

The minimal versions of Amazon Linux (`amzn-ami-minimal-*`) do not contain the above packages; however, they are available in the default **yum** repositories, and you can install them with the following command:

```
[ec2-user ~]$ sudo yum install -y package_name
```

Although the `aws-apitools-*` command line tools are included with every Amazon Linux version, the `aws-cli` command line tools provide a standard experience across all Amazon Web Services and will eventually replace the service-specific tool sets.

For instances launched using IAM roles, a simple script has been included to prepare `AWS_CREDENTIAL_FILE`, `JAVA_HOME`, `AWS_PATH`, `PATH`, and product-specific environment variables after a credential file has been installed to simplify the configuration of these tools.

Also, to allow the installation of multiple versions of the API and AMI tools, we have placed symbolic links to the desired versions of these tools in `/opt/aws`, as described here:

```
/opt/aws/bin
```

Symbolic links to `/bin` directories in each of the installed tools directories.

```
/opt/aws/{apitools|amitools}
```

Products are installed in directories of the form `name-version` and a symbolic link `name` that is attached to the most recently installed version.

```
/opt/aws/{apitools|amitools}/{name}/environment.sh
```

Used by `/etc/profile.d/aws-apitools-common.sh` to set product-specific environment variables, such as `EC2_HOME`.

cloud-init

The `cloud-init` package is an open source application built by Canonical that is used to bootstrap Linux images in a cloud computing environment, such as Amazon EC2. Amazon Linux contains a customized version of `cloud-init`. It enables you to specify actions that should happen to your instance at boot time. You can pass desired actions to `cloud-init` through the user data fields when launching an instance. This means you can use common AMIs for many use cases and configure them dynamically at startup. Amazon Linux also uses `cloud-init` to perform initial configuration of the `ec2-user` account.

For more information about `cloud-init`, see <http://cloudinit.readthedocs.org/en/latest/>.

Amazon Linux uses the following `cloud-init` actions (configurable in `/etc/sysconfig/cloudinit`):

- action: `INIT` (always runs)
 - Sets a default locale
 - Sets the hostname
 - Parses and handles user data
- action: `CONFIG_SSH`
 - Generates host private SSH keys
 - Adds a user's public SSH keys to `.ssh/authorized_keys` for easy login and administration
- action: `PACKAGE_SETUP`
 - Prepares **yum** repo
 - Handles package actions defined in user data
- action: `RUNCMD`
 - Runs a shell command
- action: `RUN_USER_SCRIPTS`
 - Executes user scripts found in user data
- action: `CONFIG_MOUNTS`
 - Mounts ephemeral drives
- action: `CONFIG_LOCALE`
 - Sets the locale in the locale configuration file according to user data

Supported User-Data Formats

The `cloud-init` package supports user-data handling of a variety of formats:

- Gzip
 - If user-data is gzip compressed, `cloud-init` decompresses the data and handles it appropriately.
- MIME multipart
 - Using a MIME multipart file, you can specify more than one type of data. For example, you could specify both a user-data script and a cloud-config type. Each part of the multipart file can be handled by `cloud-init` if it is one of the supported formats.
- Base64 decoding
 - If user-data is base64-encoded, `cloud-init` determines if it can understand the decoded data as one of the supported types. If it understands the decoded data, it decodes the data and handles it appropriately. If not, it returns the base64 data intact.
- User-Data script
 - Begins with `#!` or `Content-Type: text/x-shellscript`.
 - The script is executed by `/etc/init.d/cloud-init-user-scripts` during the first boot cycle. This occurs late in the boot process (after the initial configuration actions are performed).
- Include file
 - Begins with `#include` or `Content-Type: text/x-include-url`.
 - This content is an include file. The file contains a list of URLs, one per line. Each of the URLs is read, and their content passed through this same set of rules. The content read from the URL can be gzipped, MIME-multi-part, or plain text.
- Cloud Config Data
 - Begins with `#cloud-config` or `Content-Type: text/cloud-config`.
 - This content is cloud-config data. See the examples for a commented example of supported configuration formats.
- Cloud Boothook
 - Begins with `#cloud-boothook` or `Content-Type: text/cloud-boothook`.
 - This content is boothook data. It is stored in a file under `/var/lib/cloud` and then executed immediately.
 - This is the earliest "hook" available. Note that there is no mechanism provided for running it only one time. The boothook must take care of this itself. It is provided with the instance ID in the environment variable `INSTANCE_ID`. Use this variable to provide a once-per-instance set of boothook data.

Repository Configuration

Beginning with the 2011.09 release of Amazon Linux, Amazon Linux AMIs are treated as snapshots in time, with a repository and update structure that always gives you the latest packages when you run `yum update -y`.

The repository structure is configured to deliver a continuous flow of updates that allow you to roll from one version of Amazon Linux to the next. For example, if you launch an instance from an older version of the Amazon Linux AMI (such as 2015.09 or earlier) and run `yum update -y`, you end up with the latest packages.

You can disable rolling updates for Amazon Linux by enabling the *lock-on-launch* feature. The lock-on-launch feature locks your newly launched instance to receive updates only from the specified release of the AMI. For example, you can launch a 2015.09 AMI and have it receive only the updates that were released prior to the 2016.03 AMI, until you are ready to migrate to the 2016.03 AMI. To enable lock-

on-launch in new instances, launch it with the following user data passed to `cloud-init`, using either the Amazon EC2 console or the `ec2-run-instances` command with the `-f` flag.

Important

If you lock your AMI to a version of the repositories that is not `latest`, you will not receive any further updates. The only way to receive a continuous flow of updates for the Amazon Linux AMI is to be using the latest AMI, or to be consistently updating your old AMI with the repositories pointed to `latest`.

```
#cloud-config
repo_releasever: 2015.09
```

To lock existing instances to their current AMI release version

1. Edit `/etc/yum.conf`.
2. Comment out `releasever=latest`.
3. Run **yum clean all** to clear the cache.

Adding Packages

Amazon Linux is designed to be used with online package repositories hosted in each Amazon EC2 region. These repositories provide ongoing updates to packages in the Amazon Linux AMI, as well as access to hundreds of additional common open source server applications. The repositories are available in all regions and are accessed using **yum** update tools, as well as on the [Amazon Linux AMI packages site](#). Hosting repositories in each region enables us to deploy updates quickly and without any data transfer charges. The packages can be installed by issuing **yum** commands, such as the following example:

```
[ec2-user ~]$ sudo yum install httpd
```

Access to the Extra Packages for Enterprise Linux (EPEL) repository is configured, but it is not enabled by default. EPEL provides third-party packages in addition to those that are in the Amazon Linux repositories. The third-party packages are not supported by AWS.

If you find that Amazon Linux does not contain an application you need, you can simply install the application directly on your Amazon Linux instance. Amazon Linux uses RPMs and **yum** for package management, and that is likely the simplest way to install new applications. You should always check to see if an application is available in our central Amazon Linux repository first, because many applications are available there. These applications can easily be added to your Amazon Linux instance.

To upload your applications onto a running Amazon Linux instance, use `scp` or `sftp` and then configure the application by logging on to your instance. Your applications can also be uploaded during the instance launch by using the `PACKAGE_SETUP` action from the built-in `cloud-init` package. For more information, see [cloud-init](#) (p. 145).

Important

If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

Accessing Source Packages for Reference

You can view the source of packages you have installed on your instance for reference purposes by using tools provided in Amazon Linux. Source packages are available for all of the packages included

in Amazon Linux and the online package repository. Simply determine the package name for the source package you want to install and use the `get_reference_source` command to view source within your running instance. For example:

```
[ec2-user ~]$ get_reference_source -p bash
```

The following is a sample response:

```
Requested package: bash
Found package from local RPM database: bash-4.2.46-19.35.amzn1.x86_64
Corresponding source RPM to found package : bash-4.2.46-19.35.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to: /usr/src/srpm/debug/bash-4.2.46-19.35.amzn1.src.rpm
```

The source RPM is placed in the `/usr/src/srpm/debug` directory of your instance. From there, it can be unpacked, and, for reference, you can view the source tree using standard RPM tools. After you finish debugging, the package is available for use.

Important

If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

Developing Applications

A full set of Linux development tools is provided in the **yum** repository for Amazon Linux. To develop applications on Amazon Linux, select the development tools you need with **yum**. Alternatively, many applications developed on CentOS and other similar distributions should run on Amazon Linux.

Instance Store Access

The instance store drive `ephemeral0` is mounted in `/media/ephemeral0` only on Amazon instance store-backed AMIs. This is different than many other images that mount the instance store drive under `/mnt`.

Product Life Cycle

The Amazon Linux AMI is updated regularly with security and feature enhancements. If you do not need to preserve data or customizations on your Amazon Linux instances, you can simply relaunch new instances with the latest Amazon Linux AMI. If you need to preserve data or customizations for your Amazon Linux instances, you can maintain those instances through the Amazon Linux **yum** repositories. The **yum** repositories contain all the updated packages. You can choose to apply these updates to your running instances.

Older versions of the AMI and update packages will continue to be available for use, even as new versions are released. In some cases, if you're seeking support for an older version of Amazon Linux; through AWS Support, we might ask you to move to newer versions as part of the support process.

Security Updates

Security updates are provided via the Amazon Linux AMI **yum** repositories as well as via updated Amazon Linux AMIs. Security alerts are published in the [Amazon Linux AMI Security Center](#). For more information on AWS security policies or to report a security problem, go to the [AWS Security Center](#).

Amazon Linux AMIs are configured to download and install security updates at launch time. This is controlled via a `cloud-init` setting called `repo_upgrade`. The following snippet of `cloud-init` configuration shows how you can change the settings in the user data text you pass to your instance initialization:

```
#cloud-config
repo_upgrade: security
```

The possible values for the `repo_upgrade` setting are as follows:

`security`

Apply outstanding updates that Amazon marks as security updates.

`bugfix`

Apply updates that Amazon marks as bug fixes. Bug fixes are a larger set of updates, which include security updates and fixes for various other minor bugs.

`all`

Apply all applicable available updates, regardless of their classification.

`none`

Do not apply any updates to the instance on startup.

The default setting for `repo_upgrade` is `security`. That is, if you don't specify a different value in your user data, by default, the Amazon Linux AMI performs the security upgrades at launch for any packages installed at that time. The Amazon Linux AMI also notifies you of any updates to the installed packages by listing the number of available updates upon login using the `/etc/motd` file. To install these updates, you need to run **`sudo yum upgrade`** on the instance.

Important

If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

Support

Support for installation and use of the base Amazon Linux AMI is included through subscriptions to AWS Support. For more information, see [AWS Support](#).

We encourage you to post any questions you have about Amazon Linux to the [Amazon EC2 forum](#).

User Provided Kernels

If you have a need for a custom kernel on your Amazon EC2 instances, you can start with an AMI that is close to what you want, compile the custom kernel on your instance, and modify the `menu.lst` file to point to the new kernel. This process varies depending on the virtualization type that your AMI uses. For more information, see [Linux AMI Virtualization Types \(p. 74\)](#).

Contents

- [HVM AMIs \(GRUB\) \(p. 149\)](#)
- [Paravirtual AMIs \(PV-GRUB\) \(p. 150\)](#)

HVM AMIs (GRUB)

HVM instance volumes are treated like actual physical disks. The boot process is similar to that of a bare metal operating system with a partitioned disk and bootloader, which allows it to work with all

currently supported Linux distributions. The most common bootloader is GRUB, and the following section describes configuring GRUB to use a custom kernel.

Configuring GRUB for HVM AMIs

The following is an example of a `menu.lst` configuration file for an HVM AMI. In this example, there are two kernel entries to choose from: `Amazon Linux 2016.03` (the original kernel for this AMI), and `Vanilla Linux 4.7.4` (a newer version of the Vanilla Linux kernel from <https://www.kernel.org/>). The Vanilla entry was copied from the original entry for this AMI, and the `kernel` and `initrd` paths were updated to the new locations. The `default 0` parameter points the bootloader to the first entry that it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

By default, GRUB does not send its output to the instance console because it creates an extra boot delay. For more information, see [Instance Console Output \(p. 806\)](#). If you are installing a custom kernel, you should consider enabling GRUB output by deleting the `hiddenmenu` line and adding `serial` and `terminal` lines to `/boot/grub/menu.lst` as shown in the example below.

Important

Avoid printing large amounts of debug information during the boot process; the serial console does not support high rate data transfer.

```
default=0
fallback=1
timeout=5
serial --unit=0 --speed=9600
terminal --dumb --timeout=5 serial console

title Vanilla Linux 4.7.4
root (hd0)
kernel /boot/vmlinuz-4.7.4 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initrd.img-4.7.4

title Amazon Linux 2016.03 (4.4.11-23.53.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-4.4.11-23.53.amzn1.x86_64 root=LABEL=/ console=tty1
console=ttyS0
initrd /boot/initramfs-4.4.11-23.53.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel isn't found.

If your new Vanilla Linux kernel fails, the output will be similar to the example below.

```
^M Entry 0 will be booted automatically in 3 seconds. ^M Entry 0 will be
booted automatically in 2 seconds. ^M Entry 0 will be booted automatically
in 1 seconds.

Error 13: Invalid or unsupported executable format
[ 0.000000] Initializing cgroup subsys cpuset
```

Paravirtual AMIs (PV-GRUB)

Amazon Machine Images that use paravirtual (PV) virtualization use a system called *PV-GRUB* during the boot process. PV-GRUB is a paravirtual bootloader that runs a patched version of GNU GRUB 0.97. When you start an instance, PV-GRUB starts the boot process and then chain loads the kernel specified by your image's `menu.lst` file.

PV-GRUB understands standard `grub.conf` or `menu.lst` commands, which allows it to work with all currently supported Linux distributions. Older distributions such as Ubuntu 10.04 LTS, Oracle Enterprise Linux or CentOS 5.x require a special "ec2" or "xen" kernel package, while newer distributions include the required drivers in the default kernel package.

Most modern paravirtual AMIs use a PV-GRUB AKI by default (including all of the paravirtual Linux AMIs available in the Amazon EC2 Launch Wizard Quick Start menu), so there are no additional steps that you need to take to use a different kernel on your instance, provided that the kernel you want to use is compatible with your distribution. The best way to run a custom kernel on your instance is to start with an AMI that is close to what you want and then to compile the custom kernel on your instance and modify the `menu.lst` file as shown in [Configuring GRUB for Paravirtual AMIs \(p. 151\)](#) to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI by executing the following [describe-images](#) command with the Amazon EC2 command line tools (substituting the kernel image ID you want to check:

```
$ aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Check whether the `Name` field starts with `pv-grub`.

Topics

- [Limitations of PV-GRUB \(p. 151\)](#)
- [Configuring GRUB for Paravirtual AMIs \(p. 151\)](#)
- [Amazon PV-GRUB Kernel Image IDs \(p. 152\)](#)
- [Updating PV-GRUB \(p. 154\)](#)

Limitations of PV-GRUB

PV-GRUB has the following limitations:

- You can't use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You can't specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.
- AWS has tested and verified that PV-GRUB works with these file system formats: EXT2, EXT3, EXT4, JFS, XFS, and ReiserFS. Other file system formats might not work.
- PV-GRUB can boot kernels compressed using the gzip, bzip2, lzo, and xz compression formats.
- Cluster AMIs don't support or need PV-GRUB, because they use full hardware virtualization (HVM). While paravirtual instances use PV-GRUB to boot, HVM instance volumes are treated like actual disks, and the boot process is similar to the boot process of a bare metal operating system with a partitioned disk and bootloader.
- PV-GRUB versions 1.03 and earlier don't support GPT partitioning; they support MBR partitioning only.
- If you plan to use a logical volume manager (LVM) with Amazon EBS volumes, you need a separate boot partition outside of the LVM. Then you can create logical volumes with the LVM.

Configuring GRUB for Paravirtual AMIs

To boot PV-GRUB, a GRUB `menu.lst` file must exist in the image; the most common location for this file is `/boot/grub/menu.lst`.

The following is an example of a `menu.lst` configuration file for booting an AMI with a PV-GRUB AKI. In this example, there are two kernel entries to choose from: `Amazon Linux 2016.03` (the original kernel for this AMI), and `Vanilla Linux 4.7.4` (a newer version of the Vanilla Linux kernel from

<https://www.kernel.org/>). The Vanilla entry was copied from the original entry for this AMI, and the `kernel` and `initrd` paths were updated to the new locations. The `default 0` parameter points the bootloader to the first entry it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.7.4
root (hd0)
kernel /boot/vmlinuz-4.7.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.7.4

title Amazon Linux 2016.03 (4.4.11-23.53.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.4.11-23.53.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.4.11-23.53.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. PV-GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel isn't found.

PV-GRUB checks the following locations for `menu.lst`, using the first one it finds:

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

Note that PV-GRUB 1.03 and earlier only check one of the first two locations in this list.

Amazon PV-GRUB Kernel Image IDs

PV-GRUB AKIs are available in all Amazon EC2 regions. There are AKIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AKI by default.

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Use the following [describe-images](#) command to get a list of the PV-GRUB AKIs for the current region:

```
$ aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-*.gz
```

Note that PV-GRUB is the only AKI available in the `ap-southeast-2` region. You should verify that any AMI you want to copy to this region is using a version of PV-GRUB that is available in this region.

The following are the current AKI IDs for each region. Register new AMIs using an `hd0` AKI.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Paravirtual AMIs (PV-GRUB)**

Note

We continue to provide hd00 AKIs for backward compatibility in regions where they were previously available.

ap-northeast-1, Asia Pacific (Tokyo)

Image ID	Image Name
aki-136bf512	pv-grub-hd0_1.04-i386.gz
aki-176bf516	pv-grub-hd0_1.04-x86_64.gz

ap-northeast-2, Asia Pacific (Seoul)

Image ID	Image Name
aki-01a66b6f	pv-grub-hd0_1.04-x86_64.gz

ap-southeast-1, Asia Pacific (Singapore) Region

Image ID	Image Name
aki-ae3973fc	pv-grub-hd0_1.04-i386.gz
aki-503e7402	pv-grub-hd0_1.04-x86_64.gz

ap-southeast-2, Asia Pacific (Sydney)

Image ID	Image Name
aki-cd62fff7	pv-grub-hd0_1.04-i386.gz
aki-c362fff9	pv-grub-hd0_1.04-x86_64.gz

ap-south-1, Asia Pacific (Mumbai)

Image ID	Image Name
aki-a7305ac8	pv-grub-hd0_1.04-x86_64.gz

eu-central-1, EU (Frankfurt)

Image ID	Image Name
aki-3e4c7a23	pv-grub-hd0_1.04-i386.gz
aki-184c7a05	pv-grub-hd0_1.04-x86_64.gz

eu-west-1, EU (Ireland)

Image ID	Image Name
aki-68a3451f	pv-grub-hd0_1.04-i386.gz
aki-52a34525	pv-grub-hd0_1.04-x86_64.gz

sa-east-1, South America (São Paulo)

Image ID	Image Name
aki-5b53f446	pv-grub-hd0_1.04-i386.gz
aki-5553f448	pv-grub-hd0_1.04-x86_64.gz

us-east-1, US East (N. Virginia)

Image ID	Image Name
aki-8f9dcae6	pv-grub-hd0_1.04-i386.gz
aki-919dcaf8	pv-grub-hd0_1.04-x86_64.gz

us-gov-west-1, AWS GovCloud (US)

Image ID	Image Name
aki-1fe98d3c	pv-grub-hd0_1.04-i386.gz
aki-1de98d3e	pv-grub-hd0_1.04-x86_64.gz

us-west-1, US West (N. California)

Image ID	Image Name
aki-8e0531cb	pv-grub-hd0_1.04-i386.gz
aki-880531cd	pv-grub-hd0_1.04-x86_64.gz

us-west-2, US West (Oregon)

Image ID	Image Name
aki-f08f11c0	pv-grub-hd0_1.04-i386.gz
aki-fc8f11cc	pv-grub-hd0_1.04-x86_64.gz

Updating PV-GRUB

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Also, older versions of PV-GRUB are not available in all regions, so if you copy an AMI that uses an older version to a region that does not support that version, you will be unable to boot instances launched from that AMI until you update the kernel image. Use the following procedures to check your instance's version of PV-GRUB and update it if necessary.

To check your PV-GRUB version

1. Find the kernel ID for your instance.

```
$ ec2-describe-instance-attribute instance_id --kernel --region region
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Paravirtual AMIs (PV-GRUB)

```
kernel instance_id aki-fc8f11cc
```

The kernel ID for this instance is `aki-fc8f11cc`.

2. View the version information of that kernel ID.

```
$ ec2-describe-images aki-fc8f11cc --region region  
IMAGE aki-fc8f11cc amazon/pv-grub-hd0_1.04-x86_64.gz ...
```

This kernel image is PV-GRUB 1.04. If your PV-GRUB version is not the newest version (as shown in [Amazon PV-GRUB Kernel Image IDs \(p. 152\)](#)), you should update it using the following procedure.

To update your PV-GRUB version

If your instance is using an older version of PV-GRUB, you should update it to the latest version.

1. Identify the latest PV-GRUB AKI for your region and processor architecture from [Amazon PV-GRUB Kernel Image IDs \(p. 152\)](#).
2. Stop your instance. Your instance must be stopped to modify the kernel image used.

```
$ ec2-stop-instances instance_id --region region  
INSTANCE instance_id stopped stopped
```

3. Modify the kernel image used for your instance.

```
$ ec2-modify-instance-attribute --kernel kernel_id --  
region region instance_id
```

4. Restart your instance.

```
$ ec2-start-instances --region region instance_id
```

Amazon EC2 Instances

If you're new to Amazon EC2, see the following topics to get started:

- [What Is Amazon EC2? \(p. 1\)](#)
- [Setting Up with Amazon EC2 \(p. 19\)](#)
- [Getting Started with Amazon EC2 Linux Instances \(p. 26\)](#)
- [Instance Lifecycle \(p. 275\)](#)

Before you launch a production environment, you need to answer the following questions.

Q. What instance type best meets my needs?

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see [Instance Types \(p. 156\)](#).

Q. What purchasing option best meets my needs?

Amazon EC2 supports On-Demand instances (the default), Spot instances, and Reserved Instances. For more information, see [Instance Purchasing Options \(p. 193\)](#).

Q. Which type of root volume meets my needs?

Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see [Storage for the Root Device \(p. 71\)](#).

Q. Would I benefit from using a virtual private cloud?

If you can launch instances in either EC2-Classic or EC2-VPC, you'll need to decide which platform meets your needs. For more information, see [Supported Platforms \(p. 549\)](#) and [Amazon EC2 and Amazon Virtual Private Cloud \(p. 543\)](#).

Instance Types

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

Contents

- [Available Instance Types \(p. 157\)](#)
- [Hardware Specifications \(p. 158\)](#)
- [Virtualization Types \(p. 158\)](#)
- [Networking and Storage Features \(p. 158\)](#)
- [Instance Limits \(p. 159\)](#)

Available Instance Types

Amazon EC2 provides the instance types listed in the following tables.

Current Generation Instances

For the best performance, we recommend that you use the current generation instance types when you launch new instances. For more information about the current generation instance types, see [Amazon EC2 Instances](#).

Instance Family	Current Generation Instance Types
General purpose	t2.nano t2.micro t2.small t2.medium t2.large m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m3.medium m3.large m3.xlarge m3.2xlarge
Compute optimized	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
Memory optimized	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge x1.32xlarge
Storage optimized	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge
GPU instances	g2.2xlarge g2.8xlarge

Previous Generation Instances

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. We encourage you to use the latest generation of instances to get the best performance, but we will continue to support these previous

generation instances. If you are currently using a previous generation instance, you can see which current generation instance would be a suitable upgrade. For more information, see [Previous Generation Instances](#).

Instance Family	Previous Generation Instance Types
General purpose	m1.small m1.medium m1.large m1.xlarge
Compute optimized	c1.medium c1.xlarge cc2.8xlarge
Memory optimized	m2.xlarge m2.2xlarge m2.4xlarge cr1.8xlarge
Storage optimized	hi1.4xlarge hs1.8xlarge
GPU instances	cgl.4xlarge
Micro instances	t1.micro

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance hour, it's convenient and inexpensive to test multiple instance types before making a decision.

Even after you make a decision, if your needs change, you can resize your instance later on. For more information, see [Resizing Your Instance \(p. 189\)](#).

Virtualization Types

Each instance type supports one or both of the following types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The virtualization type of your instance is determined by the AMI that you use to launch it.

For best performance, we recommend that you use an HVM AMI. In addition, HVM AMIs are required to take advantage of enhanced networking. HVM virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses PV network and storage drivers for improved performance. For more information, see [Linux AMI Virtualization Types \(p. 74\)](#).

Networking and Storage Features

When you select an instance type, this determines which of the following networking and storage features are available:

- Some instance types are not available in EC2-Classic, so you must launch them in a VPC. By launching an instance in a VPC, you can leverage features that are not available in EC2-Classic, such as enhanced networking, assigning multiple private IP addresses to the instance, and changing the security groups assigned to your instance. For more information, see [Instance Types Available Only in a VPC \(p. 548\)](#).
- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instances that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. For more information, see [Storage \(p. 633\)](#).

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Limits**

- To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS-optimized instances. Some instance types are EBS-optimized by default. For more information, see [Amazon EBS-Optimized Instances \(p. 692\)](#).
- To maximize the networking and bandwidth performance of your instance type, you can do the following:
 - Launch supported instance types into a placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common placement group can benefit from high-bandwidth (10 Gbps), low-latency networking. For more information, see [Placement Groups \(p. 602\)](#). Instance types that support 10 Gbps network speeds can only take advantage of those network speeds when launched in a placement group.
 - Enable enhanced networking for supported current generation instance types to get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 608\)](#).
- The maximum supported MTU varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 605\)](#).

The following table summarizes the networking and storage features supported by the current generation instance types.

	VPC only	EBS only	SSD volumes	Placement group	HVM only	Enhanced networking
C3			Yes	Yes		Intel 82599 VF
C4	Yes	Yes		Yes	Yes	Intel 82599 VF
D2				Yes	Yes	Intel 82599 VF
G2			Yes	Yes	Yes	
I2			Yes	Yes	Yes	Intel 82599 VF
M3			Yes			
M4	Yes	Yes		Yes	Yes	Intel 82599 VF
R3			Yes	Yes	Yes	Intel 82599 VF
T2	Yes	Yes			Yes	
X1	Yes		Yes	Yes	Yes	ENA

Instance Limits

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.

For more information about the default limits, see [How many instances can I run in Amazon EC2?](#)

For more information about viewing your current limits or requesting an increase in your current limits, see [Amazon EC2 Service Limits](#) (p. 760).

T2 Instances

T2 instances are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload. They are intended for workloads that don't use the full CPU often or consistently, but occasionally need to burst. T2 instances are well suited for general purpose workloads, such as web servers, developer environments, and small databases. For more information about T2 instance pricing and additional hardware details, see [Amazon EC2 Instances](#).

If your account is less than 12 months old, you can use a `t2.micro` instance for free within certain usage limits. For more information, see [AWS Free Tier](#).

Contents

- [Hardware Specifications](#) (p. 160)
- [T2 Instance Requirements](#) (p. 160)
- [CPU Credits](#) (p. 161)
- [Monitoring Your CPU Credits](#) (p. 162)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

T2 Instance Requirements

The following are the requirements for T2 instances:

- You must launch your T2 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. You cannot change the instance type of an existing instance in EC2-Classic to a T2 instance type. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms](#) (p. 549) For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC](#) (p. 548).
- You must launch a T2 instance using an HVM AMI. For more information, see [Linux AMI Virtualization Types](#) (p. 74).
- You must launch your T2 instances using an EBS volume as the root device. For more information, see [Amazon EC2 Root Device Volume](#) (p. 14).
- T2 instances are available as On-Demand instances and Reserved Instances, but they are not available as Spot instances, Scheduled Instances, or Dedicated instances. They are also not supported on a Dedicated Host. For more information about these options, see [Instance Purchasing Options](#) (p. 193).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. By default, you can run up to 20 T2 instances simultaneously. If you need more T2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).
- Ensure that the T2 instance size you choose passes the minimum memory requirements of your operating system and applications. Operating systems with graphical user interfaces that consume significant memory and CPU resources (for example, Windows) may require a `t2.micro`, or larger, instance size for many use cases. As the memory and CPU requirements of your workload grows over time, you can scale to larger T2 instance sizes, or other EC2 instance types.

CPU Credits

A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits.

What is a CPU credit?

One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes.

How are CPU credits earned?

Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. The accounting process for whether credits are accumulated or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU takes a small fraction of a CPU credit.

When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed.

The following table lists the initial CPU credit allocation received at launch, the rate at which CPU credits are received, the baseline performance level as a percentage of a full core performance, and the maximum earned CPU credit balance that an instance can accrue.

Instance type	Initial CPU credits	CPU credits earned per hour	Base performance (CPU utilization)	Maximum earned CPU credit balance ^{***}
t2.nano	30	3	5%	72
t2.micro	30	6	10%	144
t2.small	30	12	20%	288
t2.medium	60	24	40%**	576
t2.large	60	36	60%**	864

* There are limits to how many T2 instances will launch or start with the initial CPU credit, which by default is set to 100 launches or starts of any T2 instance per account, per 24-hour period, per region. If you'd like to increase this limit, you can file a customer support limit increase request by using the [Amazon EC2 Credit Based Instances Launch Credits Form](#). If your account does not launch or start more than 100 T2 instances in 24 hours, this limit will not affect you.

** t2.medium and t2.large instances have two vCPUs. The base performance is an aggregate of the two vCPUs. For example, if you use 100% of a single vCPU and a small amount of the other, your CloudWatch metrics will show over 50% utilization.

*** This maximum does not include the initial CPU credits, which are used first and do not expire. For example, a t2.micro instance that was launched and then remained idle for over 24 hours could reach a credit balance of up to 174 (30 initial CPU credits + 144 earned credits). However, once the

Amazon Elastic Compute Cloud User Guide for Linux Instances T2 Instances

instance uses the initial 30 CPU credits, the credit balance can never exceed 144 unless a new initial CPU credit balance is issued by stopping and starting the instance again.

The initial credit balance is designed to provide a good startup experience. The maximum earned credit balance for an instance is equal to the number of CPU credits received per hour times 24 hours. For example, a `t2.micro` instance earns 6 CPU credits per hour and can accumulate a maximum earned CPU credit balance of 144 CPU credits.

Do CPU credits expire?

Initial CPU credits do not expire, but they are used first when an instance uses CPU credits. Unused earned credits from a given 5 minute interval expire 24 hours after they are earned, and any expired credits are removed from the CPU credit balance at that time, before any newly earned credits are added. Additionally, the CPU credit balance for an instance does not persist between instance stops and starts; stopping an instance causes it to lose its credit balance entirely, but when it restarts it will receive its initial credit balance again.

For example, if a `t2.small` instance had a CPU utilization of 5% for the hour, it would have used 3 CPU credits (5% of 60 minutes), but it would have earned 12 CPU credits during the hour, so the difference of 9 CPU credits would be added to the CPU credit balance. Any CPU credits in the balance that reached their 24 hour expiration date during that time (which could be as many as 12 credits if the instance was completely idle 24 hours ago) would also be removed from the balance. If the amount of credits expired is greater than those earned, the credit balance will go down; conversely, if the amount of credits expired is fewer than those earned, the credit balance will go up.

What happens if I use all of my credits?

If your instance uses all of its CPU credit balance, performance remains at the baseline performance level. If your instance is running low on credits, your instance's CPU credit consumption (and therefore CPU performance) is gradually lowered to the base performance level over a 15-minute interval, so you will not experience a sharp performance drop-off when your CPU credits are depleted. If your instance consistently uses all of its CPU credit balance, we recommend a larger T2 size or a fixed performance instance type such as M3 or C3.

Monitoring Your CPU Credits

You can see the credit balance for each T2 instance presented in the Amazon EC2 per-instance metrics of the CloudWatch console. T2 instances have two metrics, `CPUCreditUsage` and `CPUCreditBalance`. The `CPUCreditUsage` metric indicates the number of CPU credits used during the measurement period. The `CPUCreditBalance` metric indicates the number of unused CPU credits a T2 instance has earned. This balance is depleted during burst time as CPU credits are spent more quickly than they are earned.

The following table describes the new available CloudWatch metrics. For more information about using these metrics in CloudWatch, see [List the Available CloudWatch Metrics for Your Instances \(p. 421\)](#).

Metric	Description
<code>CPUCreditUsage</code>	<p>(Only valid for T2 instances) The number of CPU credits consumed during the specified period.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

Metric	Description
CPUCreditBalance	<p>(Only valid for T2 instances) The number of CPU credits that an instance has accumulated.</p> <p>This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

C4 Instances

C4 instances are ideal for compute-bound applications that benefit from high performance processors. C4 instances are well suited for the following applications:

- Batch processing workloads
- Media transcoding
- High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines
- High performance computing (HPC) and other compute-intensive applications

Contents

- [Hardware Specifications \(p. 163\)](#)
- [C4 Instance Features \(p. 164\)](#)
- [C4 Instance Requirements \(p. 164\)](#)
- [Support for 36 vCPUs \(p. 165\)](#)

Hardware Specifications

C4 instances are based on custom 2.9 GHz Intel Xeon E5-2666 v3 (Haswell) processors, optimized specifically for Amazon EC2. With Intel Turbo Boost Technology, the processor clock speed in C4 instances can reach as high as 3.5GHz with 1 or 2 core Turbo Boost on `c4.8xlarge` instances.

The following table highlights the feature set of the Intel Xeon E5-2666 v3 processor. For more information, see [Intel and Amazon Web Services](#).

Feature	Specification
Processor Number	E5-2666 v3
Intel Smart Cache	25 MiB
Instruction Set	64-bit
Instruction Set Extensions	AVX 2.0
Lithography	22 nm
Processor Base Frequency	2.9 GHz
Max All Core Turbo Frequency	3.2 GHz

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
C4 Instances**

Feature	Specification
Max Turbo Frequency	3.5 GHz (available on <code>c4.8xlarge</code>)
Intel Turbo Boost Technology	2.0
Intel vPro Technology	Yes
Intel Hyper-Threading Technology	Yes
Intel 64	Yes
Idle States	Yes
Enhanced Intel SpeedStep Technology	Yes
Thermal Monitoring Technologies	Yes
AES New Instructions	Yes
Secure Key	Yes
Execute Disable Bit	Yes

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

C4 Instance Features

The following is a summary of the features for C4 instances:

- C4 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 500 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your C4 instance. For more information, see [Amazon EBS–Optimized Instances](#) (p. 692).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux](#) (p. 608).
- You can cluster C4 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups](#) (p. 602).
- The `c4.8xlarge` instance type provides the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor State Control for Your EC2 Instance](#) (p. 322).

C4 Instance Requirements

The following are the requirements for C4 instances:

- C4 instances require 64-bit HVM AMIs. They have high-memory (up to 60 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- You must launch your C4 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual

network that you've defined. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 549\)](#) For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC \(p. 548\)](#).

- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some C4 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more C4 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

Support for 36 vCPUs

The `c4.8xlarge` instance type provides 36 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit of 32. We strongly recommend that you use the latest AMIs when you launch `c4.8xlarge` instances.

The following Linux AMIs support launching `c4.8xlarge` instances with 36 vCPUs:

- Amazon Linux AMI 2016.03 (HVM)
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 (HVM)

If you must use a different AMI for your application, and your `c4.8xlarge` instance launch does not complete successfully (for example, if your instance status changes to `stopped` during launch with a `Client.InstanceInitiatedShutdown` state transition reason), modify your instance as described in the following procedure to support more than 32 vCPUs so that you can use the `c4.8xlarge` instance type.

To update an instance to support more than 32 vCPUs

1. Launch a C4 instance using your AMI, choosing any C4 instance type other than `c4.8xlarge`.
2. Update the kernel to the latest version by following your operating system-specific instructions. For example, for RHEL 6, use the **`sudo yum update -y kernel`** command.
3. Stop the instance.
4. (Optional) Create an AMI from the instance that you can use to launch any additional `c4.8xlarge` instances that you need in the future.
5. Change the instance type of your stopped instance to `c4.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
6. Start the instance. If the instance launches properly, you are done. If the instance still does not boot properly, proceed to the next step.
7. (Optional) If the instance still does not boot properly, the kernel on your instance may not support more than 32 vCPUs. However, you may be able to boot the instance if you limit the vCPUs.
 - a. Change the instance type of your stopped instance to any C4 instance type other than `c4.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
 - b. Add the `maxcpus=32` option to your boot kernel parameters by following your operating system-specific instructions. For example, for RHEL 6, edit the `/boot/grub/menu.lst` file and add the following option to the most recent and active `kernel` entry:

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
```

```
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32
console=ttyS0 ro root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9
rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8
xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8 console=tty0
rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_NO_LVM
rd_NO_DM
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. Stop the instance.
- d. (Optional) Create an AMI from the instance that you can use to launch any additional `c4.8xlarge` instances that you need in the future.
- e. Change the instance type of your stopped instance to `c4.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
- f. Start the instance.

Linux GPU Instances

If you require high parallel processing capability, you'll benefit from using GPU instances, which provide access to NVIDIA GPUs with up to 1,536 CUDA cores and 4 GB of video memory. You can use GPU instances to accelerate many scientific, engineering, and rendering applications by leveraging the Compute Unified Device Architecture (CUDA) or OpenCL parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

GPU instances run as HVM-based instances. Hardware virtual machine (HVM) virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. This enables Amazon EC2 to provide dedicated access to one or more discrete GPUs in each GPU instance.

You can cluster GPU instances into a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 602\)](#).

Contents

- [Hardware Specifications \(p. 166\)](#)
- [GPU Instance Limitations \(p. 166\)](#)
- [AMIs for GPU Instances \(p. 167\)](#)
- [Installing the NVIDIA Driver on Linux \(p. 167\)](#)

For information about Windows GPU Instances, see [Windows GPU Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

GPU Instance Limitations

GPU instances have the following limitations:

- You must launch the instance using an HVM AMI.
- They can't access the GPU unless the NVIDIA drivers are installed.
- There is a limit on the number of instances that you can run. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ. To request an increase in these limits, use the following form: [Request to Increase Amazon EC2 Instance Limit](#).

AMIs for GPU Instances

To help you get started, NVIDIA provides AMIs for GPU instances. These reference AMIs include the NVIDIA driver, which enables full functionality and performance of the NVIDIA GPUs. For a list of AMIs with the NVIDIA driver, see [AWS Marketplace \(NVIDIA GRID\)](#).

You can launch CG1 and G2 instances using any HVM AMI.

Installing the NVIDIA Driver on Linux

A GPU instance must have the appropriate NVIDIA driver. The NVIDIA driver you install must be compiled against the kernel that you intend to run on your instance.

Amazon provides AMIs with updated and compatible builds of the NVIDIA kernel drivers for each official kernel upgrade in the AWS Marketplace. If you decide to use a different NVIDIA driver version than the one that Amazon provides, or decide to use a kernel that's not an official Amazon build, you must uninstall the Amazon-provided NVIDIA packages from your system to avoid conflicts with the versions of the drivers that you are trying to install.

Use this command to uninstall Amazon-provided NVIDIA packages:

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

The Amazon-provided CUDA toolkit package has dependencies on the NVIDIA drivers. Uninstalling the NVIDIA packages erases the CUDA toolkit. You must reinstall the CUDA toolkit after installing the NVIDIA driver.

You can download NVIDIA drivers from <http://www.nvidia.com/Download/Find.aspx>. Select the appropriate driver for your instance:

G2 Instances

Product Type	GRID
Product Series	GRID Series
Product	GRID K520
Operating System	Linux 64-bit
Recommended/Beta	Recommended/Certified

CG1 Instances

Product Type	Tesla
Product Series	M-Class
Product	M2050
Operating System	Linux 64-bit

Recommended/Beta

Recommended/Certified

For more information about installing and configuring the driver, open the **ADDITIONAL INFORMATION** tab on the download page for the driver on the NVIDIA website and click the README link.

Install the NVIDIA Driver Manually

To install the driver for an Amazon Linux AMI

1. Run the **yum update** command to get the latest versions of packages for your instance.

```
[ec2-user ~]$ sudo yum update -y
```

2. Reboot your instance to load the latest kernel version.

```
[ec2-user ~]$ sudo reboot
```

3. Reconnect to your instance after it has rebooted.
4. Install the "Development tools" package group.

```
[ec2-user ~]$ sudo yum groupinstall -y "Development tools"
```

5. Make sure the `kernel-devel` package is installed and matches the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install kernel-devel-`uname -r`
```

6. Download the driver package that you identified earlier. For example, the following command downloads the 361.45.11 version of the G2 instance driver.

```
[ec2-user ~]$ wget http://us.download.nvidia.com/XFree86/Linux-x86_64/361.45.11/NVIDIA-Linux-x86_64-361.45.11.run
```

7. Run the self-install script to install the NVIDIA driver. For example:

```
[ec2-user ~]$ sudo /bin/bash ./NVIDIA-Linux-x86_64-361.45.11.run
```

8. Reboot the instance. For more information, see [Reboot Your Instance \(p. 302\)](#).
9. Confirm that the driver is functional. The response for the following command lists the installed NVIDIA driver version and details about the GPUs.

```
[ec2-user ~]$ nvidia-smi -q | head

=====NVSMI LOG=====

Timestamp                : Thu Oct  2 17:28:29 2014
Driver Version           : 361.45.11

Attached GPUs            : 1
GPU 0000:00:03.0
  Product Name           : GRID K520
  Product Brand          : Grid
```

I2 Instances

I2 instances are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

Contents

- [Hardware Specifications](#) (p. 169)
- [I2 Instance Features](#) (p. 169)
- [I2 Instance Requirements](#) (p. 169)
- [SSD I/O Performance](#) (p. 170)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

I2 Instance Features

The following is a summary of the features for I2 instances:

- The primary data storage is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you stop or terminate an instance, the applications and data in its instance store are erased. We recommend that you regularly back up or replicate the data that you've stored in instance storage. For more information, see [SSD Instance Store Volumes](#) (p. 722).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux](#) (p. 608).
- You can cluster I2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups](#) (p. 602).
- You can enable EBS-optimization to obtain additional, dedicated capacity for Amazon EBS I/O. For more information, see [Amazon EBS-Optimized Instances](#) (p. 692).

I2 Instance Requirements

The following are the requirements for I2 instances:

- You must launch an I2 instance using an HVM AMI.
- To ensure the best IOPS performance from your instance on Linux, we recommend that you use the most recent version of the [Amazon Linux AMI](#), or another Linux AMI with a kernel version of 3.8 or later. If you do not use a Linux AMI with a kernel version of 3.8 or later, your instance will not achieve the maximum IOPS performance available for this instance type. For more information, see [SSD I/O Performance](#) (p. 170).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some I2 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more I2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

SSD I/O Performance

If you use a Linux AMI with kernel version 3.8 or later and utilize all the SSD-based instance store volumes available to the instance, you can get at least the minimum random IOPS (4,096 byte block size) listed in the following table. Otherwise, you'll get lower IOPS performance than what is shown in the table.

Instance Size	Read IOPS	First Write IOPS
i2.xlarge	35,000	35,000
i2.2xlarge	75,000	75,000
i2.4xlarge	175,000	155,000
i2.8xlarge	365,000	315,000

As you fill the SSD-based instance storage for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an I2 instance don't have any space reserved for over-provisioning. To reduce write amplification, you should leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance.

I2 instance store-backed volumes support TRIM. You can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance Store Volume TRIM Support \(p. 722\)](#).

D2 Instances

D2 instances are designed for workloads that require high sequential read and write access to very large data sets on local storage. D2 instances are well suited for the following applications:

- Massive parallel processing (MPP) data warehouse
- MapReduce and Hadoop distributed computing
- Log or data processing applications

Contents

- [Hardware Specifications \(p. 171\)](#)

- [D2 Instance Features](#) (p. 171)
- [D2 Instance Requirements](#) (p. 171)
- [Support for 36 vCPUs](#) (p. 172)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

D2 Instance Features

The following is a summary of the features for D2 instances:

- The primary data storage for D2 instances is HDD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. For more information about instance store volumes, see [Amazon EC2 Instance Store](#) (p. 715).
- D2 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 750 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your D2 instance. For more information, see [Amazon EBS–Optimized Instances](#) (p. 692).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux](#) (p. 608).
- You can cluster D2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups](#) (p. 602).
- The `d2.8xlarge` instance type provides the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor State Control for Your EC2 Instance](#) (p. 322).

D2 Instance Requirements

The following are the requirements for D2 instances:

- D2 instances require 64-bit HVM AMIs. They have high-memory (up to 244 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some D2 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more D2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

- Your `d2.8xlarge` instances are capable of providing up to 3.5 GB/s read performance and 3.1 GB/s write performance with a 2 MiB block size. To ensure the best disk throughput performance from your D2 instances on Linux, we recommend that you use the most recent version of the Amazon Linux AMI, or another Linux AMI with a kernel version of 3.8 or later. D2 instances provide the best disk performance when you use a Linux kernel that supports persistent grants, an extension to the Xen block ring protocol that significantly improves disk throughput and scalability. For more information about persistent grants, see [this article](#) in the Xen Project Blog.

- The `d2.8xlarge` instance type has 36 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit of 32. For more information, see [Support for 36 vCPUs](#) (p. 172).

Support for 36 vCPUs

The `d2.8xlarge` instance type provides 36 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit of 32. We strongly recommend that you use the latest AMIs when you launch `d2.8xlarge` instances.

The following Linux AMIs support launching `d2.8xlarge` instances with 36 vCPUs:

- Amazon Linux AMI 2016.03 (HVM)
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 (HVM)

If you must use a different AMI for your application, and your `d2.8xlarge` instance launch does not complete successfully (for example, if your instance status changes to `stopped` during launch with a `Client.InstanceInitiatedShutdown` state transition reason), modify your instance as described in the following procedure to support more than 32 vCPUs so that you can use the `d2.8xlarge` instance type.

To update an instance to support more than 32 vCPUs

1. Launch a D2 instance using your AMI, choosing any D2 instance type other than `d2.8xlarge`.
2. Update the kernel to the latest version by following your operating system-specific instructions. For example, for RHEL 6, use the `sudo yum update -y kernel` command.
3. Stop the instance.
4. (Optional) Create an AMI from the instance that you can use to launch any additional `d2.8xlarge` instances that you need in the future.
5. Change the instance type of your stopped instance to `d2.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
6. Start the instance. If the instance launches properly, you are done. If the instance still does not boot properly, proceed to the next step.
7. (Optional) If the instance still does not boot properly, the kernel on your instance may not support more than 32 vCPUs. However, you may be able to boot the instance if you limit the vCPUs.
 - a. Change the instance type of your stopped instance to any D2 instance type other than `d2.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
 - b. Add the `maxcpus=32` option to your boot kernel parameters by following your operating system-specific instructions. For example, for RHEL 6, edit the `/boot/grub/menu.lst` file and add the following option to the most recent and active `kernel` entry:

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32
console=ttyS0 ro root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9
```

```
rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us LANG=en_US.UTF-8  
xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8 console=tty0  
rd_NO_MD SYSFONT=latacyrheb-sun16 crashkernel=auto rd_NO_LVM  
rd_NO_DM  
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. Stop the instance.
- d. (Optional) Create an AMI from the instance that you can use to launch any additional `d2.8xlarge` instances that you need in the future.
- e. Change the instance type of your stopped instance to `d2.8xlarge` (choose **Actions**, select **Instance Settings**, choose **Change Instance Type**, and then follow the directions).
- f. Start the instance.

H1 Instances

H1 instances (`hi1.4xlarge`) can deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

You can cluster H1 instances in a placement group. For more information, see [Placement Groups \(p. 602\)](#).

By default, you can run up to two `hi1.4xlarge` instances. If you need more than two `hi1.4xlarge` instances, contact <http://aws.amazon.com/premiumsupport/>.

Contents

- [Hardware Specifications \(p. 173\)](#)
- [Disk I/O Performance \(p. 173\)](#)
- [SSD Storage \(p. 174\)](#)

Hardware Specifications

The `hi1.4xlarge` instance type is based on solid-state drive (SSD) technology.

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Disk I/O Performance

Using Linux paravirtual (PV) AMIs, H1 instances can deliver more than 120,000 4 KB random read IOPS and between 10,000 and 85,000 4 KB random write IOPS (depending on active logical block addressing span) to applications across two SSD data volumes. Using hardware virtual machine (HVM) AMIs, performance is approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

H1 Windows instances deliver approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

The maximum sequential throughput is approximately 2 GB read per second and 1.1 GB write per second.

SSD Storage

With SSD storage on H1 instances:

- The primary data source is an instance store with SSD storage.
- Read performance is consistent and write performance can vary.
- Write amplification can occur.
- The TRIM command is not currently supported.

Instance Store with SSD Storage

The `hi1.4xlarge` instances use an Amazon EBS-backed root device. However, their primary data storage is provided by the SSD volumes in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Because the root device of the `hi1.4xlarge` instance is Amazon EBS-backed, you can still start and stop your instance. When you stop an instance, your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 715\)](#).

Variable Write Performance

Write performance depends on how your applications utilize logical block addressing (LBA) space. If your applications use the total LBA space, write performance can degrade by about 90 percent. Benchmark your applications and monitor the queue length (the number of pending I/O requests for a volume) and I/O size.

Write Amplification

Write amplification refers to an undesirable condition associated with flash memory and SSDs, where the actual amount of physical information written is a multiple of the logical amount intended to be written. Because flash memory must be erased before it can be rewritten, the process to perform these operations results in moving (or rewriting) user data and metadata more than once. This multiplying effect increases the number of writes required over the life of the SSD, which shortens the time that it can reliably operate. The `hi1.4xlarge` instances are designed with a provisioning model intended to minimize write amplification.

Random writes have a much more severe impact on write amplification than serial writes. If you are concerned about write amplification, allocate less than the full terabyte of storage for your application (also known as over provisioning).

The TRIM Command

The TRIM command enables the operating system to notify an SSD that blocks of previously saved data are considered no longer in use. TRIM limits the impact of write amplification.

TRIM support is not available for H1 instances. For information about instances that support TRIM, see [Instance Store Volume TRIM Support \(p. 722\)](#).

HS1 Instances

HS1 instances (`hs1.8xlarge`) provide very high storage density and high sequential read and write performance per instance. They are well suited for the following scenarios:

- Data warehousing
- Hadoop/MapReduce
- Parallel file systems

You can cluster HS1 instances in a placement group. For more information, see [Placement Groups \(p. 602\)](#).

By default, you can run up to two HS1 instances. If you need more than two HS1 instances, you can request more using the [Amazon EC2 Instance Request Form](#).

Contents

- [Hardware Specifications \(p. 175\)](#)
- [Instance Store \(p. 175\)](#)
- [Disk Initialization \(p. 175\)](#)
- [Setting the Memory Limit \(p. 175\)](#)
- [Setting the User Limit \(ulimit\) \(p. 175\)](#)

Hardware Specifications

HS1 instances support both Amazon Elastic Block Store (Amazon EBS)-backed and instance store-backed Amazon Machine Images (AMIs). HS1 instances support both paravirtual (PV) and hardware virtual machine (HVM) AMIs.

HS1 instances provide high bandwidth networking and can also be used with Provisioned IOPS SSD (i.o1) volumes for improved consistency and performance.

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Instance Store

HS1 instances support both instance store and Amazon EBS root device volumes. However, even when using an Amazon EBS-backed instance, primary data storage is provided by the hard disk drives in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 715\)](#).

Disk Initialization

If you plan to run an HS1 instance in a steady state for long periods of time, we recommend that you zero the hard disks first for improved performance. This process can take as long as six hours to complete.

Setting the Memory Limit

Many Linux-based AMIs come with `CONFIG_XEN_MAX_DOMAIN_MEMORY` set to 70. We recommend that you set this as appropriate for 117 GiB of memory.

Setting the User Limit (ulimit)

Many Linux-based AMIs come with a default ulimit of 1024. We recommend that you increase the ulimit to 2048.

T1 Micro Instances

T1 Micro instances (`t1.micro`) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

Note

The `t1.micro` is a previous generation instance and it has been replaced by the `t2.micro`, which has a much better performance profile. We recommend using the `t2.micro` instance type instead of the `t1.micro`. For more information, see [T2 Instances \(p. 160\)](#).

The `t1.micro` instance is available as an Amazon EBS-backed instance only.

This documentation describes how `t1.micro` instances work so that you can understand how to apply them. It's not our intent to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance.

Topics

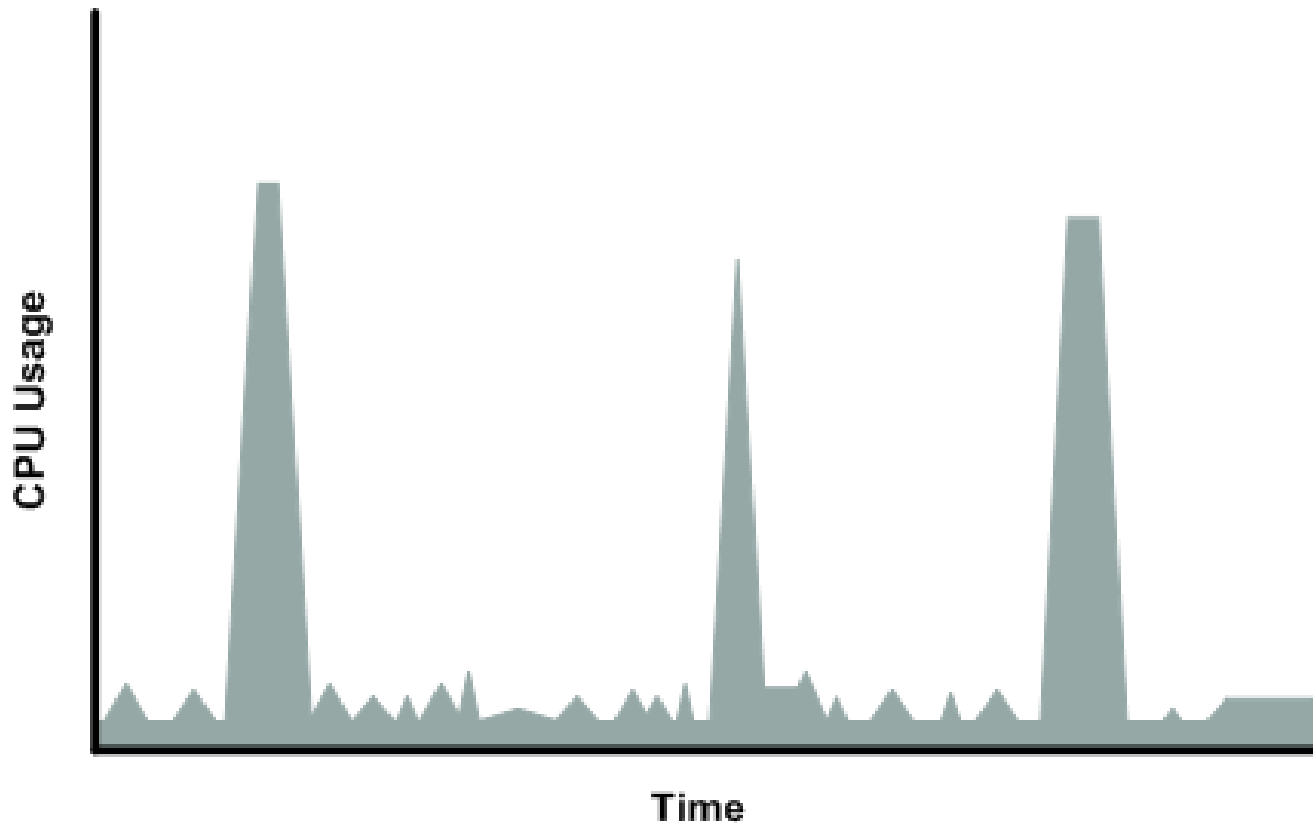
- [Hardware Specifications \(p. 176\)](#)
- [Optimal Application of T1 Micro Instances \(p. 176\)](#)
- [Available CPU Resources During Spikes \(p. 180\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 180\)](#)
- [Comparison with the m1.small Instance Type \(p. 183\)](#)
- [AMI Optimization for Micro Instances \(p. 186\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

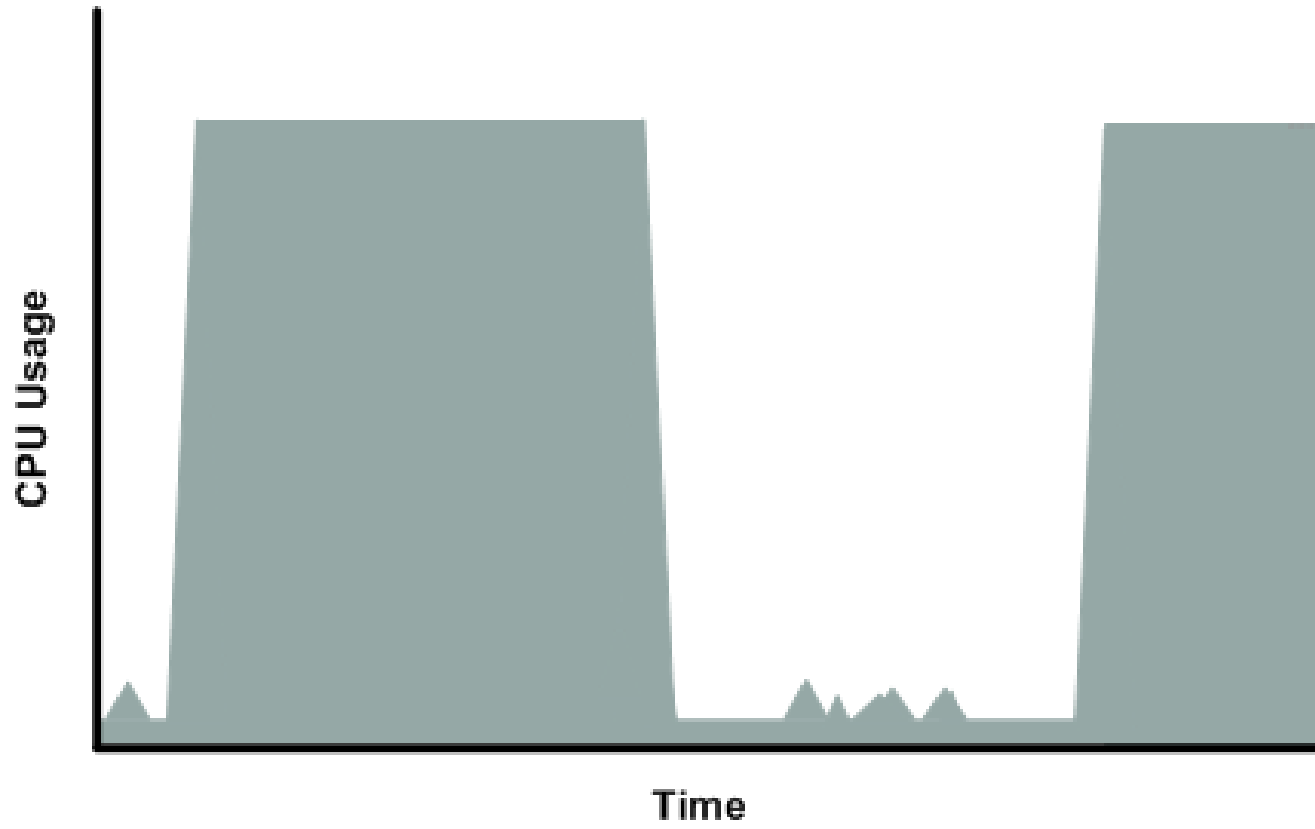
Optimal Application of T1 Micro Instances

A `t1.micro` instance provides spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.

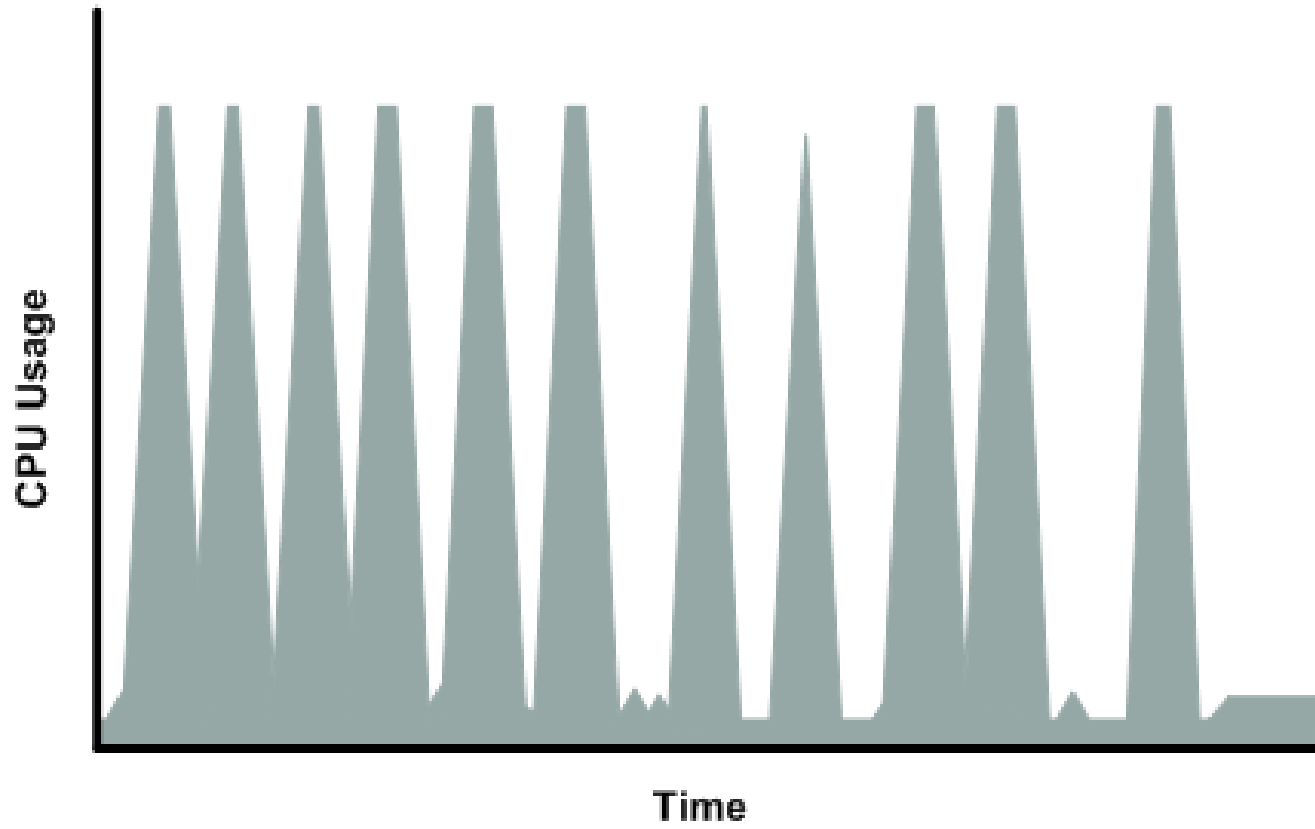


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor). The ratio between the maximum level and the background level is designed to be large. We designed `t1.micro` instances to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

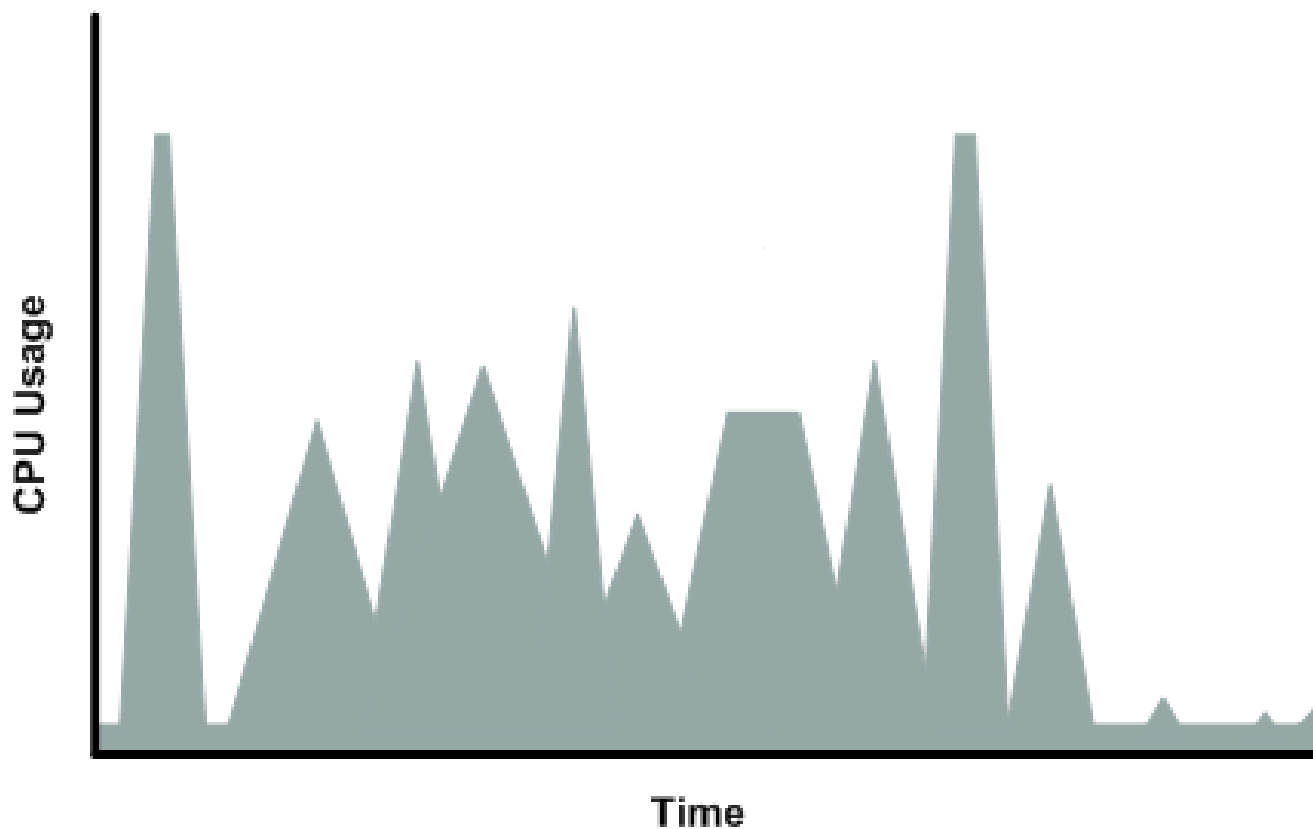
Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a `t1.micro` instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus of CPU usage that the `t1.micro` instance isn't designed to handle.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a `t1.micro` instance.



In each of the preceding cases of workloads not appropriate for a `t1.micro` instance, we recommend that you consider using a different instance type. For more information about instance types, see [Instance Types](#) (p. 156).

Available CPU Resources During Spikes

When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

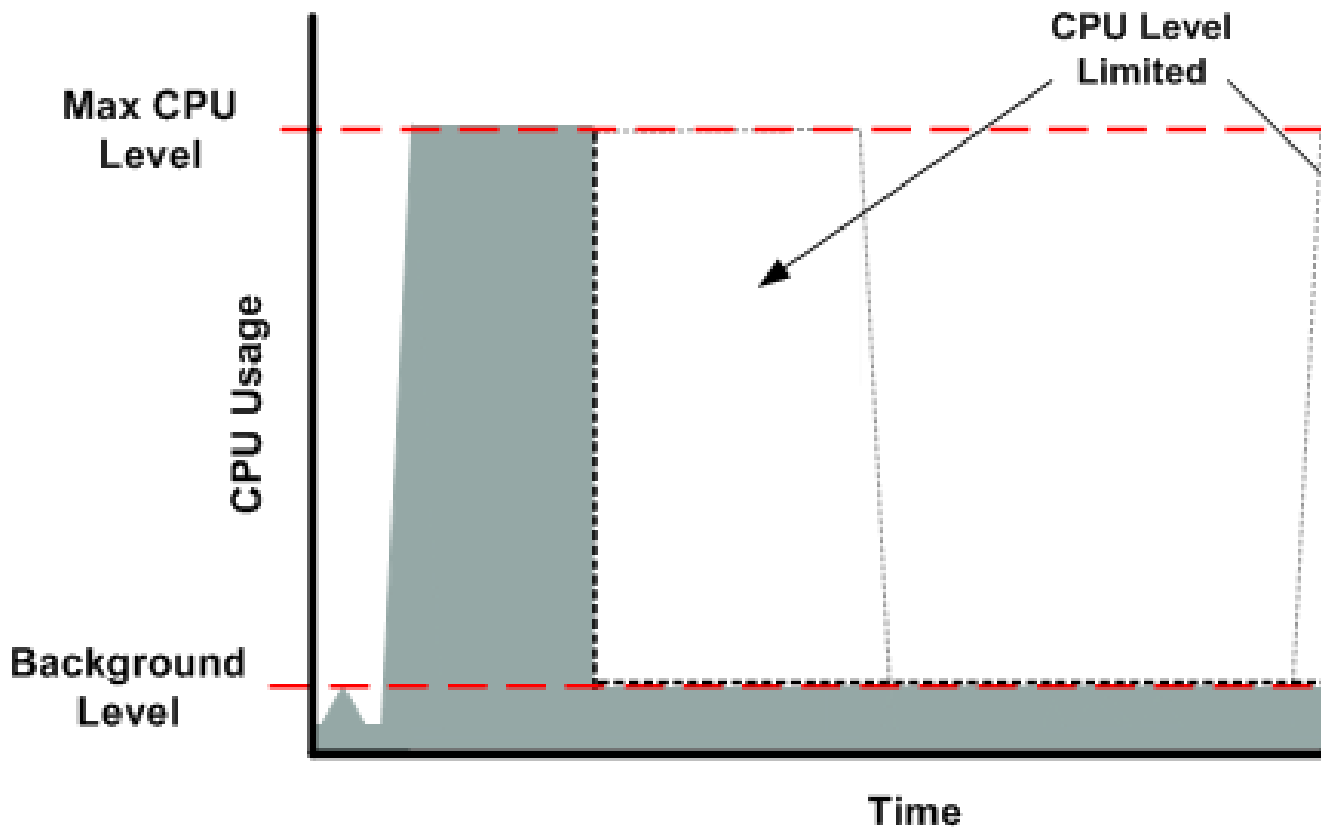
When the Instance Uses Its Allotted Resources

We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

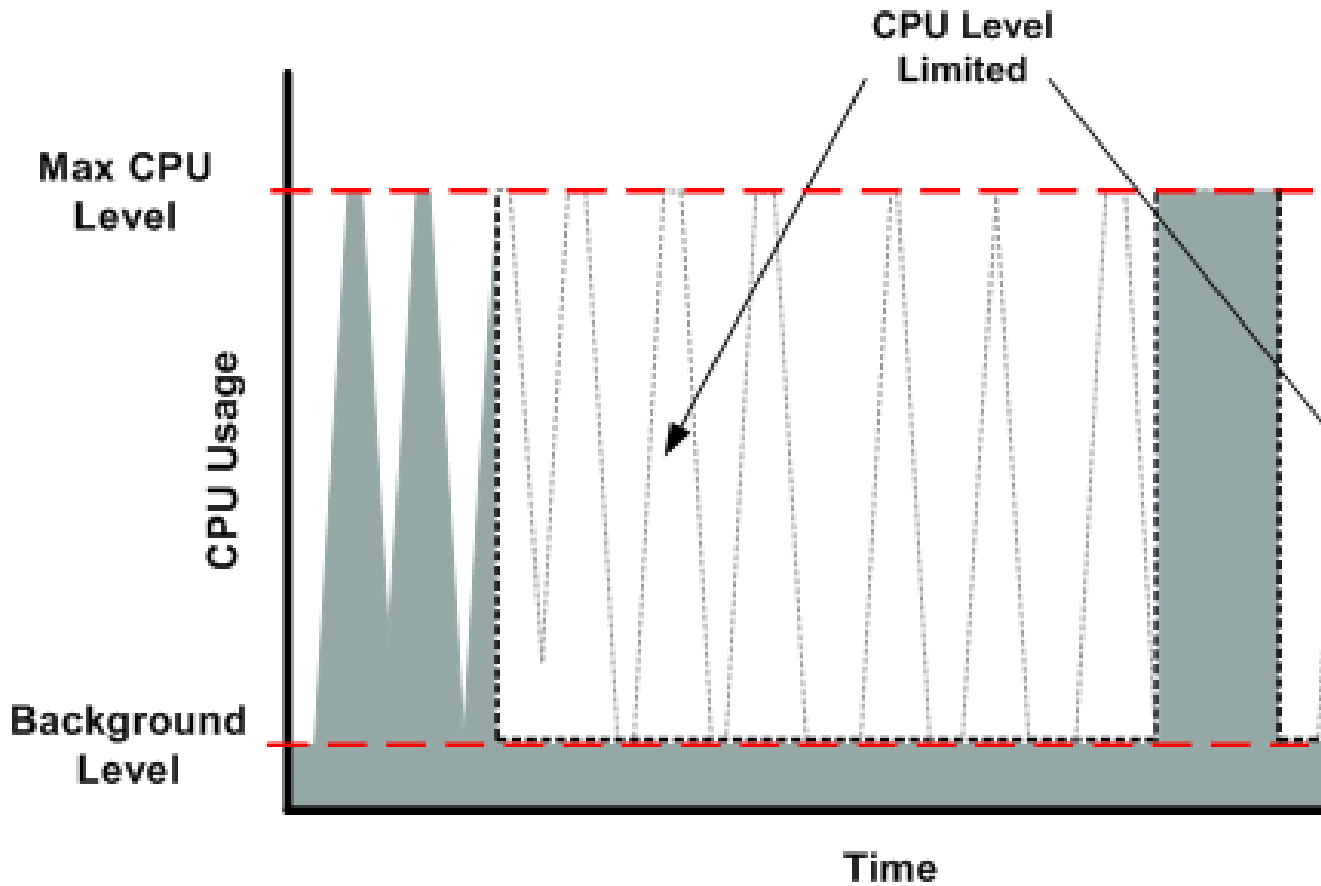
If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Auto Scaling to scale out (with additional `t1.micro` instances and a load balancer), or move to a larger instance type. For more information, see the [Auto Scaling User Guide](#).

The following figures show the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we have to limit its CPU level. If the instance consumes its allotted resources, we restrict it to the low background level.

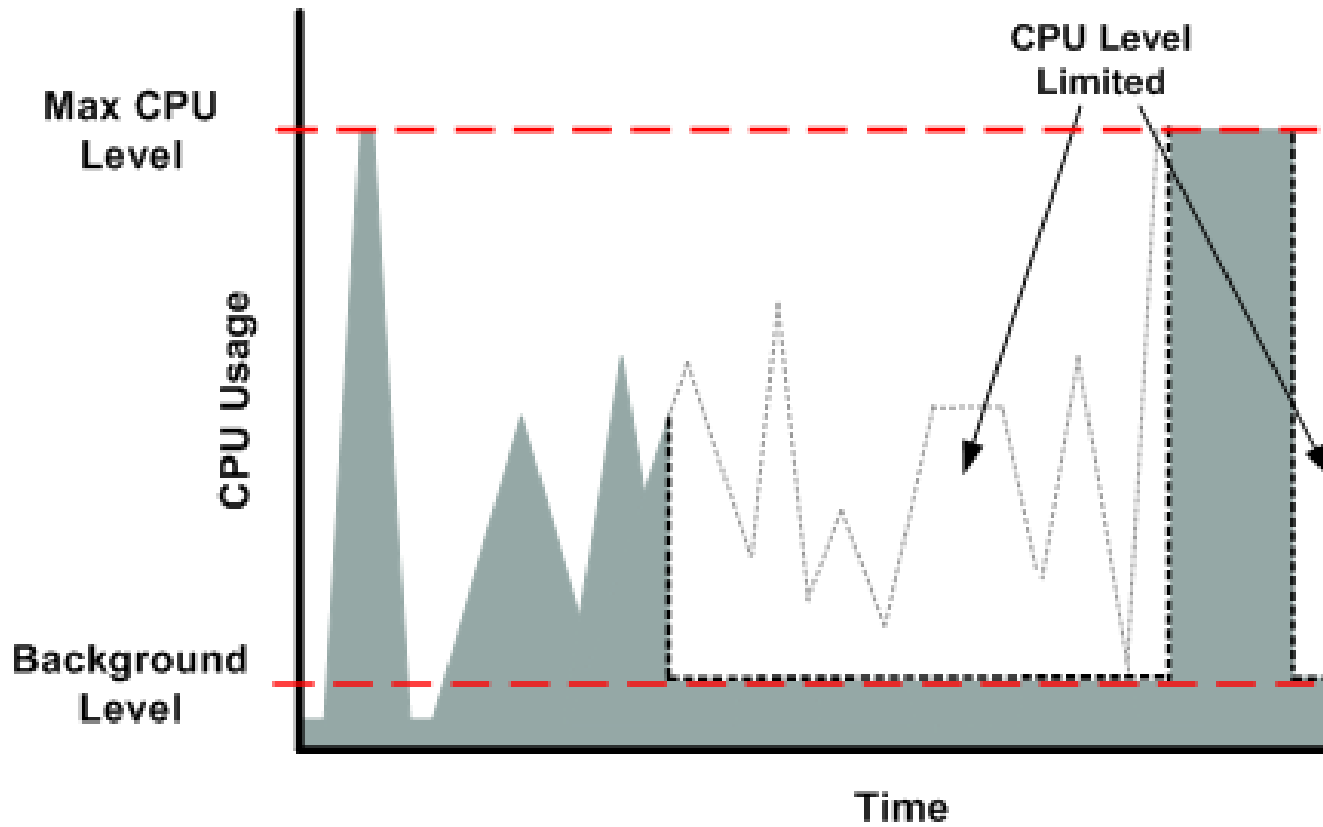
The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).



The next figure shows the situation where the requests are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.

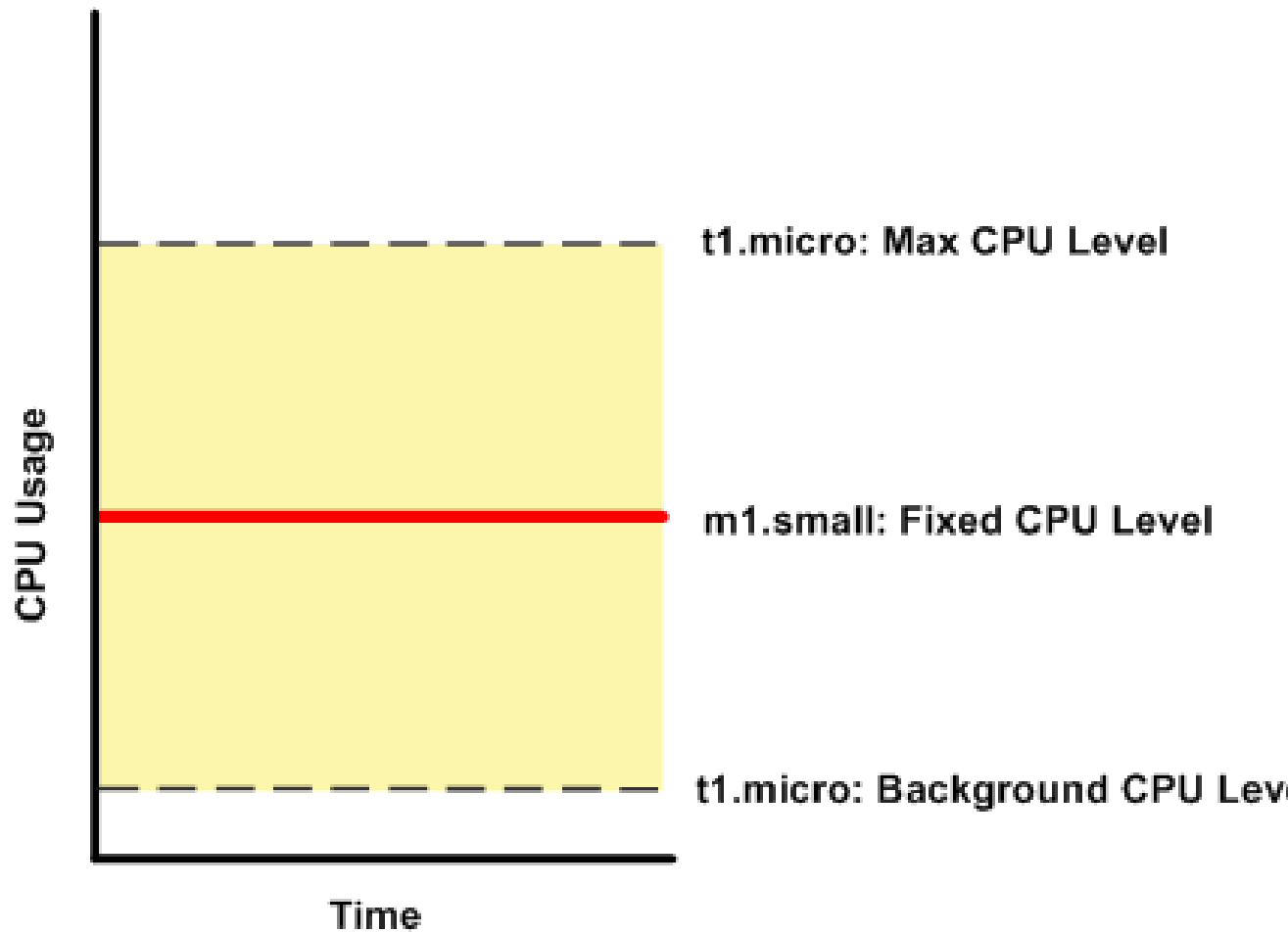


The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.



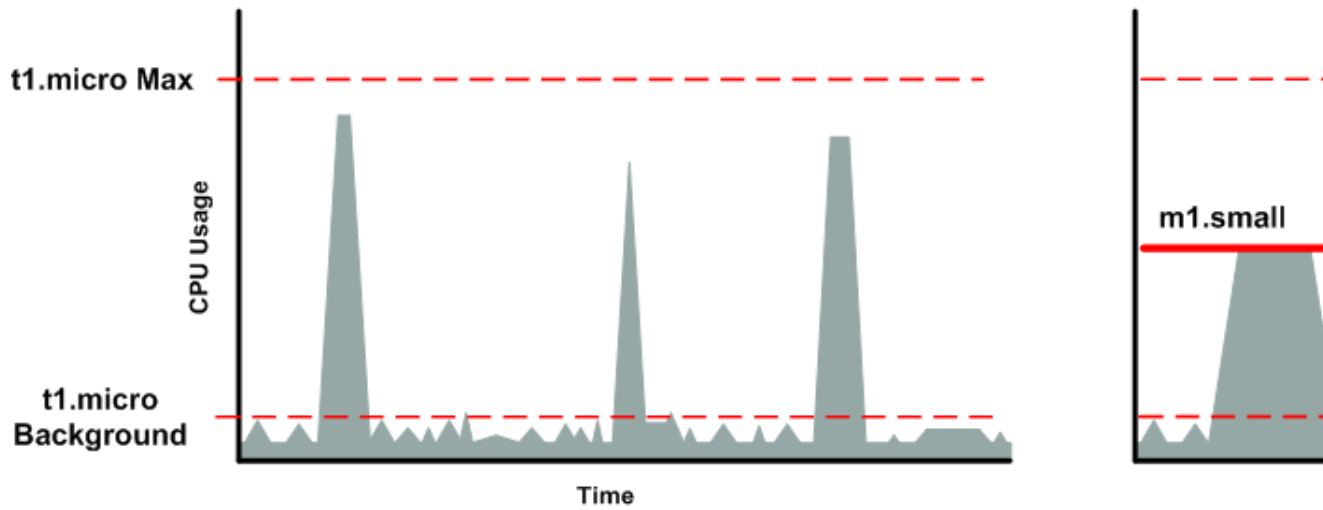
Comparison with the m1.small Instance Type

The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.

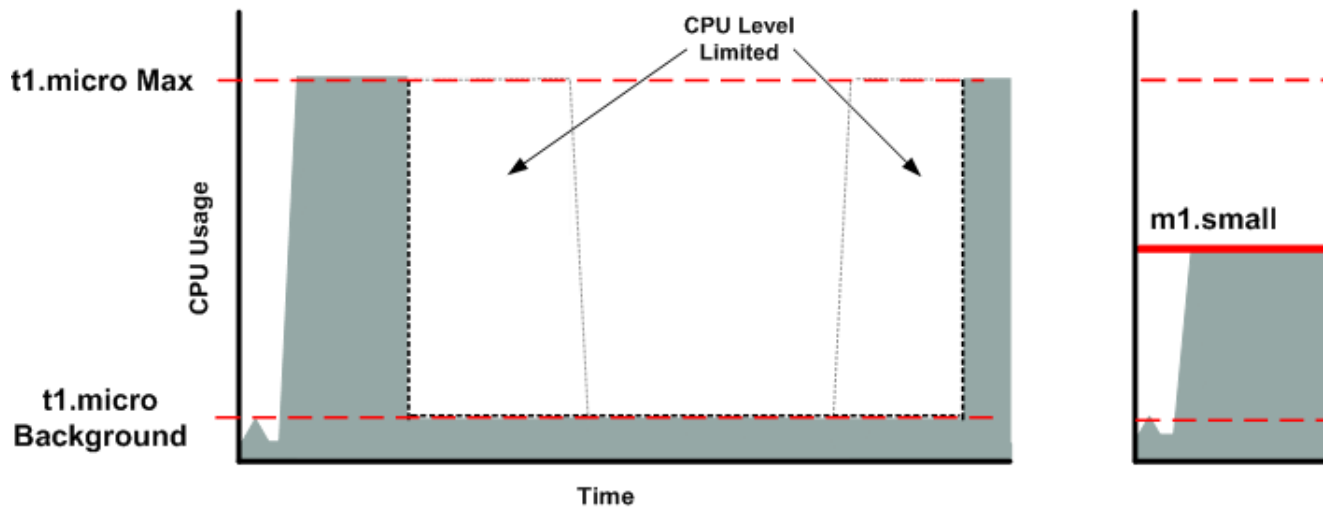


The following figures compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections.

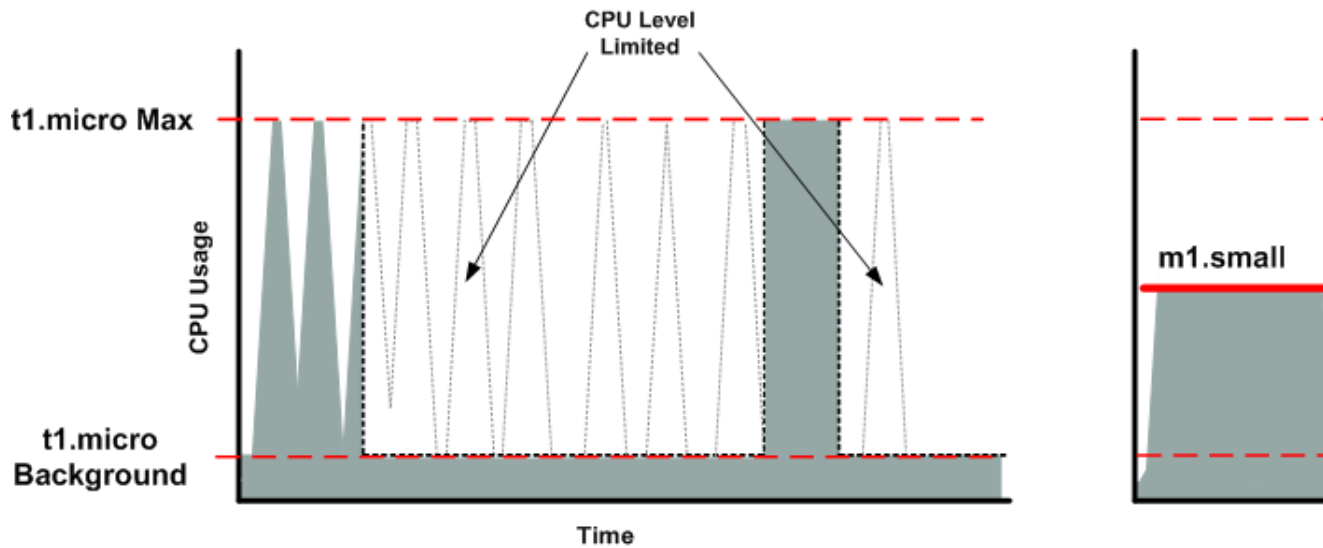
The first figure that follows shows an optimal scenario for a `t1.micro` instance (the left graph) and how it might look for an `m1.small` instance (the right graph). In this case, we don't need to limit the `t1.micro` instance. The processing time on the `m1.small` instance would be longer for each spike in CPU demand compared to the `t1.micro` instance.



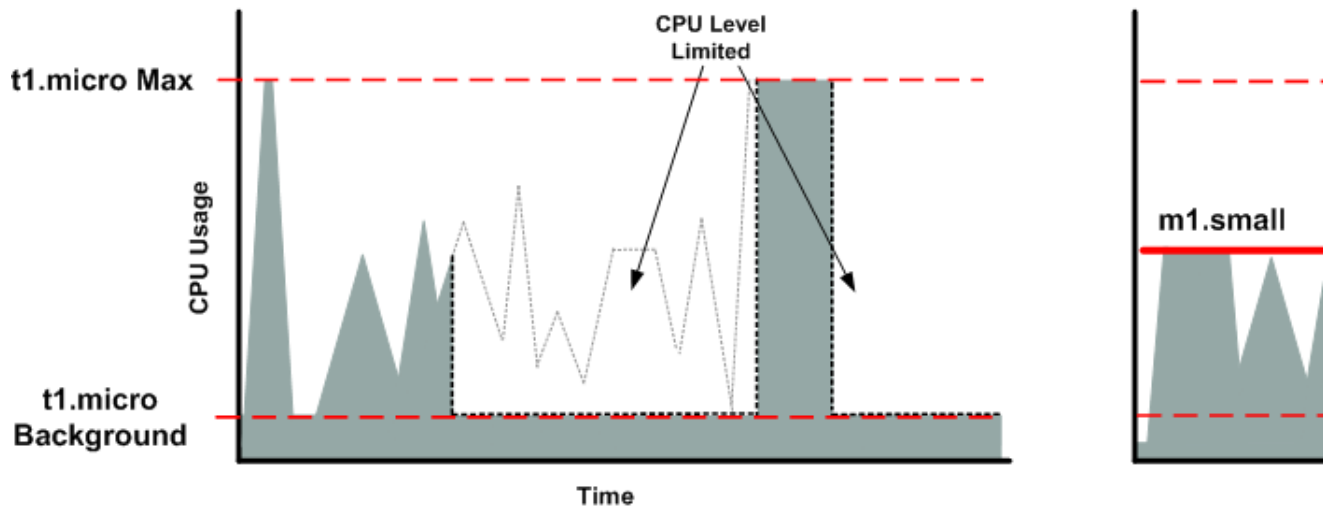
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the `t1.micro` instance, and how they might look with the `m1.small` instance.



The next figure shows the frequent requests that used up the allotted resources on the `t1.micro` instance, and how they might look on the `m1.small` instance.



The next figure shows the situation where the background level used up the allotted resources on the `t1.micro` instance, and how it might look on the `m1.small` instance.



AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the `t1.micro` instance type:

- Design the AMI to run on 600 MB of RAM
- Limit the number of recurring processes that use CPU time (for example, cron jobs, daemons)

You can optimize performance using swap space and virtual memory (for example, by setting up swap space in a separate partition from the root file system).

X1 Instances

X1 instances are designed to deliver fast performance for workloads that process large data sets in memory. X1 instances are well suited for the following applications:

- In-memory databases such as SAP HANA, including SAP-certified support for Business Suite S/4HANA, Business Suite on HANA (SoH), Business Warehouse on HANA (BW), and Data Mart Solutions on HANA. For more information, see [SAP HANA on the AWS Cloud](#).
- Big-data processing engines such as Apache Spark or Presto.
- High-performance computing (HPC) applications.

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Memory Performance

X1 instances include Intel Scalable Memory Buffers, providing 300 GiB/s of sustainable memory-read bandwidth and 140 GiB/s of sustainable memory-write bandwidth.

Compute Performance

X1 instances feature up to 128 vCPUs and are powered by four Intel Xeon E7-8880 v3 processors that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications. X1 instances also enable increased cryptographic performance via the latest Intel AES-NI feature, support Intel Transactional Synchronization Extensions (TSX) to boost the performance of in-memory transactional data processing, and support Advanced Vector Extensions 2 (Intel AVX2) processor instructions to expand most integer commands to 256 bits.

The following table highlights the feature set of the Intel Xeon E7-8880 v3 processor. For more information, see [Intel and Amazon Web Services](#).

Feature	Specification
Processor number	E7-8880 v3
Last level cache	45 MB
Intel QPI speed	9.6 GT/s
Number of QPI links	3
Instruction set	64-bit
Instruction set extensions	AVX 2.0
Lithography	22 nm
Processor base frequency	2.3 GHz
Max turbo frequency	3.1 GHz
Intel turbo boost technology	2.0
Intel hyperthreading technology	Yes

Feature	Specification
Intel TSX-NI	Yes
Intel 64	Yes
Idle states	Yes
Enhanced Intel SpeedStep technology	Yes
Thermal monitoring technologies	Yes
Intel AES new instructions	Yes

X1 Instance Features

The following is a summary of the features for X1 instances:

- X1 instances are EBS-optimized by default, providing dedicated storage bandwidth of up to 10 Gbps with Amazon EBS volumes. EBS-optimized instances enable consistently high performance by eliminating contention between storage I/O and other network traffic from your X1 instance. For more information, see [Amazon EBS–Optimized Instances](#).
- X1 instances support SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. For more information about instance store volumes, see [Amazon EC2 Instance Store](#).
- You can cluster X1 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups](#).
- The `x1.32xlarge` instance type provides the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (measured by CPU frequency) from a core. For more information, see [Processor State Control for Your EC2 Instance](#).
- X1 instances are available as On-Demand Instances, Reserved Instances, Spot Instances, or Scheduled Reserved Instances, but they are not available as Dedicated Instances. They are also supported on Dedicated Hosts. For more information, see [Instance Purchasing Options](#).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. By default, you can run up to two X1 instances simultaneously. If you need more X1 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

X1 Instance Requirements

The following are the requirements for X1 instances:

- The `x1.32xlarge` instance type offers 128 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit lower than 128. For more information, see [Support for 128 vCPUs \(p. 189\)](#).
- The instance store for `x1.32xlarge` is capable of providing up to 126,000 random-read IOPS with a 4 KiB block size. To ensure the best disk I/O performance from your X1 instances on Linux, we recommend that you use the most recent version of the Amazon Linux AMI, or another Linux AMI with a kernel version of 3.8 or later. X1 instances provide the best disk performance when you use a Linux kernel that supports persistent grants, an extension to the Xen block ring protocol that significantly improves disk throughput and scalability. For more information about persistent grants, see the [Improving block protocol scalability with persistent grants](#) post on the Xen Project Blog.

- X1 instances have high-memory (up to 1,952 GiB of RAM) and require 64-bit HVM AMIs to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. For more information, see [Linux AMI Virtualization Types](#).
- You must launch your X1 instances into a virtual private cloud (VPC); the EC2-Classic platform is not supported. EC2-VPC enables you to launch AWS resources into a virtual network that you've defined. You cannot change the instance type of an existing instance in EC2-Classic to an X1 instance type. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms](#). For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC](#).

Support for 128 vCPUs

The `x1.32xlarge` instance type provides 128 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit lower than 128. We strongly recommend that you use the latest AMIs when you launch `x1.32xlarge` instances.

The following Linux AMIs support launching `x1.32xlarge` instances with 128 vCPUs:

- Amazon Linux AMI 2016.03 (HVM)
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 (HVM)

Resizing Your Instance

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t2.micro` instance is too small for its workload, you can change it to an `m3.medium` instance.

If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its instance type, which is known as *resizing* it. If the root device for your instance is an instance store volume, you must migrate your application to a new instance with the instance type that you want. For more information about root device volumes, see [Storage for the Root Device \(p. 71\)](#).

When you resize an instance, you must select an instance type that is compatible with the configuration of the instance. If the instance type that you want is not compatible with the instance configuration you have, then you must migrate your application to a new instance with the instance type that you want.

Important

When you resize an instance, the resized instance usually has the same number of instance store volumes that you specified when you launched the original instance. If you want to add instance store volumes, you must migrate your application to a completely new instance with the instance type and instance store volumes that you want. An exception to this rule is when you resize to a storage-intensive instance type that by default contains a higher number of volumes. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 715\)](#).

Contents

- [Compatibility for Resizing Instances \(p. 190\)](#)
- [Resizing an Amazon EBS-backed Instance \(p. 190\)](#)
- [Migrating an Instance Store-backed Instance \(p. 191\)](#)

- [Migrating to a New Instance Configuration](#) (p. 192)

Compatibility for Resizing Instances

You can resize an instance only if its current instance type and the new instance type that you want are compatible in the following ways:

- **Virtualization type.** Linux AMIs use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). You can't resize an instance that was launched from a PV AMI to an instance type that is HVM only. For more information, see [Linux AMI Virtualization Types](#) (p. 74).
- **Network.** Some instance types are not supported in EC2-Classic and must be launched in a VPC. Therefore, you can't resize an instance in EC2-Classic to a instance type that is available only in a VPC unless you have a nondefault VPC. For more information, see [Instance Types Available Only in a VPC](#) (p. 548).
- **Platform.** All Amazon EC2 instance types support 64-bit AMIs, but only the following instance types support 32-bit AMIs: `t2.nano`, `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are resizing a 32-bit instance, you are limited to these instance types.

For example, T2 instances are not supported in EC2-Classic and they are HVM only. Therefore, you can't resize a T1 instance to a T2 instance because T1 instances do not support HVM and must be launched from PV AMIs. If you want to resize a T2 instance to a larger instance type, you can select any current generation instance type, such as M3, because all current generation instance types support HVM AMIs. For more information, see [Available Instance Types](#) (p. 157).

Resizing an Amazon EBS-backed Instance

You must stop your Amazon EBS-backed instance before you can change its instance type. When you stop and start an instance, be aware of the following:

- We move the instance to new hardware; however, the instance ID does not change.
- If your instance is running in a VPC and has a public IP address, we release the address and give it a new public IP address. The instance retains its private IP addresses and any Elastic IP addresses.
- If your instance is running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. Therefore, to ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the Auto Scaling processes for the group while you're resizing your instance. For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling User Guide*.

For more information, see [Stop and Start Your Instance](#) (p. 299).

Use the following procedure to resize an Amazon EBS-backed instance using the AWS Management Console.

To resize an Amazon EBS-backed instance

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**, and select the instance.
3. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.

4. Choose **Actions**, select **Instance State**, and then choose **Stop**.
5. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.
6. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **Change Instance Type**. Note that this action is disabled if the instance state is not `stopped`.
7. In the **Change Instance Type** dialog box, do the following:
 - a. From **Instance Type**, select the instance type that you want. If the instance type that you want does not appear in the list, then it is not compatible with the configuration of your instance (for example, because of virtualization type).
 - b. (Optional) If the instance type that you selected supports EBS-optimization, select **EBS-optimized** to enable EBS-optimization or deselect **EBS-optimized** to disable EBS-optimization. Note that if the instance type that you selected is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it.
 - c. Choose **Apply** to accept the new settings.
8. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
9. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.
10. [EC2-Classic] When the instance state is `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance. If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions** and then choose **Associate Address**.
 - d. From **Instance**, select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

Migrating an Instance Store-backed Instance

When you want to move your application from one instance store-backed instance to an instance store-backed instance with a different instance type, you must migrate it by creating an image from your instance, and then launching a new instance from this image with the instance type that you need. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your original instance and associate it with the new instance. Then you can terminate the original instance.

To migrate an instance store-backed instance

1. [EC2-Classic] If the instance you are migrating has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the new instance later.
2. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, take a snapshot of the volumes (see [Creating an Amazon EBS Snapshot \(p. 686\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS Volume from an Instance \(p. 669\)](#)).
3. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#). When you are finished creating an AMI from your instance, return to this procedure.

4. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **Owned by me**, and select the image that you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

Note

If you do not see the AMI that you created in the previous step, make sure that you have selected the region in which you created your AMI.

5. Choose **Launch**. When you specify options for the instance, be sure to select the new instance type that you want. If the instance type that you want can't be selected, then it is not compatible with configuration of the AMI that you created (for example, because of virtualization type). You can also specify any EBS volumes that you detached from the original instance.

Note that it can take a few minutes for the instance to enter the `running` state.

6. [EC2-Classic] If the instance that you started with had an associated Elastic IP address, you must associate it with the new instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you recorded at the beginning of this procedure.
 - c. Choose **Actions** and then choose **Associate Address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
7. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

Migrating to a New Instance Configuration

If the current configuration of your instance is incompatible with the new instance type that you want, then you can't resize the instance to that instance type. Instead, you can migrate your application to a new instance with a configuration that is compatible with the new instance type that you want.

If you want to move from an instance launched from a PV AMI to an instance type that is HVM only, the general process is as follows:

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see [Creating an Amazon EBS Snapshot \(p. 686\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS Volume from an Instance \(p. 669\)](#)).
2. Launch a new instance, selecting the following:
 - An HVM AMI.
 - The HVM only instance type.
 - [EC2-VPC] If you are using an Elastic IP address, select the VPC that the original instance is currently running in.
 - Any EBS volumes that you detached from the original instance and want to attach to the new instance, or new EBS volumes based on the snapshots that you created.
 - If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
3. Install your application and any required software on the instance.
4. Restore any data that you backed up from the instance store volumes of the original instance.
5. If you are using an Elastic IP address, assign it to the newly launched instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.

- b. Select the Elastic IP address that is associated with the original instance, choose **Actions**, and then choose **Disassociate Address**. When prompted for confirmation, choose **Yes, Disassociate**.
 - c. With the Elastic IP address still selected, choose **Actions**, and then choose **Associate Address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

For information about migrating an application from an instance in EC2-Classic to an instance in a VPC, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 560\)](#).

Instance Purchasing Options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

- **On-Demand instances** — Pay, by the hour, for the instances that you launch.
- **Reserved Instances** — Purchase, at a significant discount, instances that are always available, for a term from one to three years.
- **Scheduled Instances** — Purchase instances that are always available on the specified recurring schedule, for a one-year term.
- **Spot instances** — Bid on unused instances, which can run as long as they are available and your bid is above the Spot price, at a significant discount.
- **Dedicated hosts** — Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated instances** — Pay, by the hour, for instances that run on single-tenant hardware. For more information, see [Dedicated Instances](#) in the *Amazon VPC User Guide*.

If you require a capacity reservation, consider Reserved Instances or Scheduled Instances. Spot instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted. Dedicated hosts can help you address compliance requirements and reduce costs by using your existing server-bound software licenses. For more information, see [Amazon EC2 Instance Purchasing Options](#).

Contents

- [Determining the Instance Lifecycle \(p. 193\)](#)
- [Reserved Instances \(p. 194\)](#)
- [Scheduled Reserved Instances \(p. 215\)](#)
- [Spot Instances \(p. 219\)](#)
- [Dedicated Hosts \(p. 264\)](#)

Determining the Instance Lifecycle

The lifecycle of an instance starts when it is launched and ends when it is terminated. The purchasing option that you choose effects the lifecycle of the instance. For example, an On-Demand instance runs when you launch it and ends when you terminate it. A Spot instance runs as long as its capacity

is available and your bid price is higher than the Spot price. You can launch a Scheduled Instance during its scheduled time period; Amazon EC2 launches the instances and then terminates them three minutes before the time period ends.

Use the following procedure to determine the lifecycle of an instance.

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Description** tab, find **Tenancy**. If the value is `host`, the instance is running on a Dedicated Host. If the value is `dedicated`, the instance is a Dedicated Instance.
5. On the **Description** tab, find **Lifecycle**. If the value is `spot`, the instance is a Spot instance. If the value is `scheduled`, the instance is a Scheduled Instance. If the value is `normal`, the instance is either an On-Demand instance or a Reserved Instance.
6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see [Reserved Instance Utilization Reports \(p. 767\)](#).

To determine the instance lifecycle using the AWS CLI

Use the following `describe-instances` command:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

If the instance is running on a Dedicated host, the output contains the following information:

```
"Tenancy": "host"
```

If the instance is a Dedicated instance, the output contains the following information:

```
"Tenancy": "dedicated"
```

If the instance is a Spot instance, the output contains the following information:

```
"InstanceLifecycle": "spot"
```

If the instance is a Scheduled Instance, the output contains the following information:

```
"InstanceLifecycle": "scheduled"
```

Otherwise, the output contains the following information:

```
"InstanceLifecycle": "normal"
```

Reserved Instances

Reserved Instances are a billing discount and capacity reservation that is applied to instances to lower hourly running costs. A Reserved Instance is not a physical instance. The discounted usage price is

fixed for as long as you own the Reserved Instance, allowing you to predict compute costs over the term of the reservation. If you are expecting consistent, heavy, use, Reserved Instances can provide substantial savings over owning your own hardware or running only On-Demand instances. For more information, see [Choosing a Reserved Instance Payment Option \(p. 197\)](#).

When you purchase a Reserved Instance, the reservation is automatically applied to running instances that match your specified parameters. Alternatively, you can launch an On-Demand EC2 instance with the same configuration as the purchased reserved capacity. No Upfront and Partial Upfront Reserved Instances are billed for usage on an hourly basis, regardless of whether or not they are being used. All Upfront Reserved Instances have no additional hourly charges.

Reserved Instances do not renew automatically; you can continue using the EC2 instance without interruption, but you will be charged On-Demand rates. New Reserved Instances can have the same parameters as the expired ones, or you can purchase Reserved Instances with different parameters.

You can use Auto Scaling or other AWS services to launch the On-Demand instances that use your Reserved Instance benefits. For information about launching On-Demand instances, see [Launch Your Instance](#). For information about launching instances using Auto Scaling, see the [Auto Scaling User Guide](#).

For product pricing information, see the following pages:

- [AWS Service Pricing Overview](#)
- [Amazon EC2 On-Demand Instances Pricing](#)
- [Amazon EC2 Reserved Instance Pricing](#)
- For information about the Reserved Instance pricing tiers, see [Understanding Reserved Instance Discount Pricing Tiers \(p. 199\)](#).

Note

Light, Medium, and Heavy Utilization Reserved Instances are no longer available for purchase. For more information about how these options are affected by changes to the Reserved Instances pricing model, see [Reserved Instances FAQ](#).

Topics

- [How Reserved Instances Work \(p. 195\)](#)
- [Billing Benefits and Payment Options \(p. 197\)](#)
- [Buying Reserved Instances \(p. 201\)](#)
- [Selling in the Reserved Instance Marketplace \(p. 204\)](#)
- [Modifying Your Reserved Instances \(p. 210\)](#)
- [Troubleshooting Modification Requests \(p. 215\)](#)

How Reserved Instances Work

Amazon EC2 Reserved Instances and the Reserved Instance Marketplace can be a powerful and cost-saving strategy for running your business. However, before you use Reserved Instances or the Reserved Instance Marketplace, ensure that you meet the requirements for purchase and sale. You also must understand the details and restrictions on certain elements of Reserved Instances and the Reserved Instance Marketplace—including seller registration, banking, using the AWS free tier, dealing with cancelled instances, and so on. Use this topic as a checklist for buying and selling Reserved Instances, and for buying and selling in the Reserved Instance Marketplace.

Note

To purchase and modify Reserved Instances, ensure that your IAM user account has the appropriate permissions, such as the ability to describe Availability Zones. For information,

see the [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Getting Started

- **AWS account**—You need to have an AWS account in order to purchase Reserved Instances. If you don't have an AWS account, read and complete the instructions described in [Setting Up with Amazon EC2 \(p. 19\)](#), which provides information on signing up for your Amazon EC2 account and credentials.
- **AWS free tier**—The AWS free usage tier is available for new AWS accounts. If you are using the AWS free usage tier to run Amazon EC2 instances, and then you purchase a Reserved Instance, you will be charged under standard pricing guidelines. For information about the free tier and the applicable services and usage amounts, see [AWS Free Tier](#).

Buying Reserved Instances

- **Usage fee**—With Reserved Instances, you pay for the entire term regardless of *whether or not you use it*.
- **Tiered discounts on purchases**—The Reserved Instance pricing tier discounts only apply to purchases made from AWS. These discounts do not apply to purchases of third-party Reserved Instances. For information, see [Understanding Reserved Instance Discount Pricing Tiers \(p. 199\)](#).
- **Cancellation of purchase**—Before you confirm your purchase, review the details of the Reserved Instances that you plan to buy, and make sure that all the parameters are accurate. After you purchase a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase. However, you may be able to sell the Reserved Instance if your needs change. For information, see [Listing Your Reserved Instances \(p. 207\)](#).

Selling Reserved Instances and the Reserved Instance Marketplace

- **Seller requirement**—To become a seller in the Reserved Instance Marketplace, you must register as a seller. For information, see [Listing Your Reserved Instances \(p. 207\)](#).
- **Bank requirement**—AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address. For more information, see [Bank Accounts \(p. 205\)](#).
- **Tax requirement**—Sellers who have 50 or more transactions or who plan to sell \$20,000 or more in Reserved Instances will have to provide additional information about their business for tax reasons. For information, see [Tax Information \(p. 206\)](#).
- **Minimum selling price**—The minimum price allowed in the Reserved Instance Marketplace is \$0.00.
- **When Reserved Instances can be sold**—Reserved Instances can be sold only after AWS has received the upfront payment and the Reserved Instance has been active (you've owned it) for at least 30 days. In addition, there must be at least one month remaining in the term of the Reserved Instance you are listing.
- **Modifying your listing**—It is not possible to modify your listing in the Reserved Instance Marketplace directly. However, you can change your listing by first cancelling it and then creating another listing with new parameters. For information, see [Pricing Your Reserved Instances \(p. 209\)](#). You can also change your Reserved Instances before listing them. For information, see [Modifying Your Reserved Instances \(p. 210\)](#).
- **Selling discounted Reserved Instances**—Amazon EC2 Reserved Instances purchased at a reduced cost resulting from a tiering discount cannot be sold in the Reserved Instance Marketplace. For more information, see [Reserved Instance Marketplace \(p. 197\)](#).
- **Service fee**—AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace. (The upfront price is the price the seller is charging for the Reserved Instance.)

- **Other AWS Reserved Instances**—Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other AWS Reserved Instances, such as Amazon Relational Database Service (Amazon RDS) and Amazon ElastiCache Reserved Instances cannot be sold in the Reserved Instance Marketplace.

Using Reserved Instances in an Amazon VPC

You can launch instances into an Amazon Virtual Private Cloud (Amazon VPC) and benefit from your Reserved Instances. For more information see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

If you have an EC2-Classic account, you can purchase Reserved Instances to apply to instances launched into a nondefault VPC by selecting a platform that includes *Amazon VPC* in its name. For more information, see [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).

If you have an EC2-VPC-only account, the listed platforms available do not include *Amazon VPC* in its name because all platforms have default subnets. When you launch an instance with the same configuration as the capacity you reserved, and that instance is launched into your default or nondefault VPC, the capacity reservation and billing benefits are automatically applied to your instance. For information about default VPCs, see [Your Default VPC and Subnets](#) in the *Amazon VPC User Guide*.

You can also choose to purchase Reserved instances that are physically isolated at the host hardware level by specifying *dedicated* as the instance tenancy. For more information about Dedicated instances, see [Dedicated Instances](#) in the *Amazon VPC User Guide*.

Reserved Instance Marketplace

Note

Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other types, such as Amazon RDS and Amazon ElastiCache Reserved Instances, cannot be sold on the Reserved Instance Marketplace.

The Reserved Instance Marketplace is a platform that supports the sale of third-party and AWS customers' unused Reserved Instances, which will vary in term lengths and pricing options. For example, an AWS customer may want to sell capacity after moving instances to a new AWS region, changing to a new instance type, or ending projects before the term expiration. The Reserved Instance Marketplace provides increased selection and flexibility by allowing you to address your specific business and searching for Reserved Instances that most closely match your preferred combination of instance type, region, and duration. Amazon EC2 instances purchased on the Reserved Instance Marketplace offer the same capacity reservations as Reserved Instances purchased directly from AWS.

Billing Benefits and Payment Options

The billing benefit of Reserved Instances is automatically applied to matching running instances. You can also purchase Reserved Instances and then launch On-Demand instances with matching specifications—the pricing benefit is automatically applied to those instances. The capacity reservation ensures that you will always be able to launch instances with those specifications. You also can sell your unused Reserved Instances in the Reserved Instance Marketplace, provided you meet certain requirements.

Choosing a Reserved Instance Payment Option

There are three payment options for Reserved Instances:

- **No Upfront**—You are billed a discounted hourly rate for every hour within the term, regardless of usage, and no upfront payment is required. This option is only available as a 1-year reservation.

Note

No Upfront Reserved Instances are based on a contractual obligation to pay monthly for the entire term of the reservation. For this reason, a successful billing history is required before an account is eligible to purchase No Upfront Reserved Instances.

- **Partial Upfront**—A portion of the cost must be paid upfront and the remaining hours in the term are billed at a discounted hourly rate, regardless of usage.
- **All Upfront**—Full payment is made at the start of the term, with no other costs incurred for the remainder of the term regardless of the number of hours used.

Understanding hourly billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running or not. It's important to understand the difference between instance states and how these impact billing hours. For more information, see [Instance Lifecycle \(p. 275\)](#).

Reserved Instance billing benefits only apply to one instance-hour per clock-hour. An instance-hour begins when an instance is started and continues for 60 minutes or until the instance is stopped or terminated—whichever happens first. A clock-hour is defined as the standard 24-hour clock that runs from midnight to midnight, and is divided into 24 hours (for example, 1:00:00 to 1:59:59 is one clock-hour).

A new instance-hour begins after an instance has run for 60 continuous minutes, or if an instance is stopped and then started. Rebooting an instance does not reset the running instance-hour.

For example, if an instance is stopped and then started again during a clock-hour and continues running for two more clock-hours, the first instance-hour (before the restart) is charged at the discounted Reserved Instance rate. The next instance-hour (after restart) is charged at the On-Demand rate and the next two instance-hours are charged at the discounted Reserved Instance rate.

The [Reserved Instance Utilization Reports \(p. 767\)](#) section includes sample reports which illustrate the savings against running On-Demand instances. The [Reserved Instances FAQ](#) includes a sample list value calculation.

How to apply a Reserved Instance to a running instance

To cover a running instance with reservation, you can either modify an existing Reserved Instance or purchase a Reserved Instance by selecting the Availability Zone (such as us-east-1b), instance type (such as `m3.large`), platform (such as Amazon Linux VPC), and tenancy (such as default) to match the configuration of the running instance.

Below is an example scenario where a customer has purchased the following On-Demand instances:

- 4 x `m1.small` instances in Availability Zone us-east-1a
- 4 x `c1.medium` instances in Availability Zone us-east-1b
- 2 x `c1.xlarge` instances in Availability Zone us-east-1b

The customer then purchases the following reservations:

- 2 x `m1.small` in Availability Zone us-east-1a
- 3 x `c1.medium` in Availability Zone us-east-1a
- 1 x `c1.xlarge` in Availability Zone us-east-1b

The pricing benefit is applied in the following way:

- The discount for the two `m1.small` reservations is applied to two of the four running `m1.small` instances in Availability Zone us-east-1a.

The other two instances in Availability Zone us-east-1a continue to be charged at the current On-Demand rate.

- The three `c1.medium` reservations don't match the Availability Zone of the running `c1.medium` instances. The four running `c1.medium` instances continue to be charged at the current On-Demand rate.

If the customer launches a `c1.medium` instance in Availability Zone us-east-1a, then the discounted usage fee is applied to that instance.

- The discounted usage fee rate for one `c1.xlarge` reservation is applied to one of the two running `c1.xlarge` instances in Availability Zone us-east-1b.

The other `c1.xlarge` instance in Availability Zone us-east-1b continues to be charged at the current On-Demand rate.

In this example the hourly usage discount is immediately applied to the hourly fee for the two `m1.small` and one `c1.xlarge` On-Demand instances that were already running; and the customer is assured of the capacity to run the remaining four instances when they are needed.

Understanding Reserved Instance Discount Pricing Tiers

When your account qualifies for a discount pricing tier, it automatically receives discounts on upfront and hourly usage fees for all Reserved Instance purchases that you make within that tier level from that point on. To qualify for a discount, the list value of your Reserved Instances in the region must be \$500,000 USD or more.

Topics

- [Calculating Reserved Instance Pricing Discounts \(p. 199\)](#)
- [Consolidated Billing for Pricing Tiers \(p. 200\)](#)
- [Buying with a Discount Tier \(p. 200\)](#)
- [Current Pricing Tier Limits \(p. 200\)](#)
- [Crossing Pricing Tiers \(p. 201\)](#)

Calculating Reserved Instance Pricing Discounts

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a region. Multiply the hourly recurring price for each reservation by the hours left in each term and add the undiscounted upfront price (known as the fixed price) listed on the [AWS marketing website](#) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

To view the fixed price values for Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Turn on the display of the **Fixed Price column** by choosing **Show/Hide** in the top right corner.

To view the fixed price values for Reserved Instances using the command line

- Using the AWS CLI, see [describe-reserved-instances](#)
- Using the AWS Tools for Windows PowerShell, see [Get-EC2ReservedInstance](#)

- Using the Amazon EC2 API, see [DescribeReservedInstances](#)

Consolidated Billing for Pricing Tiers

A consolidated billing account aggregates the list value of member accounts within a region. When the list value of all active Reserved Instances for the consolidated billing account reaches a discount pricing tier, any Reserved Instances purchased after this point by any member of the consolidated billing account are charged at the discounted rate (as long as the list value for that consolidated account stays above the discount pricing tier threshold). For more information, see [Reserved Instances and Consolidated Billing](#) (p. 201).

Buying with a Discount Tier

When you buy Reserved Instances, Amazon EC2 automatically applies any discounts to the part of your purchase that falls within a discount pricing tier. You don't need to do anything differently, and you can buy using any of the Amazon EC2 tools. For more information, see [Buying in the Reserved Instance Marketplace](#) (p. 204).

Note

Reserved Instance purchases are the only purchases that determine your discount pricing tiers, and the discounts apply only to Amazon EC2 Reserved Instance purchases.

After the list value of your active Reserved Instances in a region crosses into a discount pricing tier, any future purchase of Reserved Instances in that region are charged at a discounted rate. If a single purchase of Reserved Instances in a region takes you over the threshold of a discount tier, then the portion of the purchase that is above the price threshold is charged at the discounted rate. For more information about temporary Reserved Instance IDs created during the purchase process, see [Crossing Pricing Tiers](#) (p. 201).

If your list value falls below the price point for that discount pricing tier—for example, if some of your Reserved Instances expire—future purchases of Reserved Instances in the region are not discounted. However, you continue to get the discount applied against any Reserved Instances that were originally purchased within the discount pricing tier.

When you buy Reserved Instances, one of four possible scenarios occurs:

- **No discount**—Your purchase within a region is still below the discount threshold.
- **Partial discount**—Your purchase within a region crosses the threshold of the first discount tier. No discount is applied to one or more reservations and the discounted rate is applied to the remaining reservations.
- **Full discount**—Your entire purchase within a region falls within one discount tier and is discounted appropriately.
- **Two discount rates**—Your purchase within a region crosses from a lower discount tier to a higher discount tier. You are charged two different rates: one or more reservation at the lower discounted rate, and the remaining reservations at the higher discounted rate.

Current Pricing Tier Limits

The following limitations currently apply to Reserved Instance pricing tiers:

- Reserved Instance pricing tiers and related discounts apply only to purchases of Amazon EC2 Reserved Instances.
- Reserved Instance pricing tiers do not apply to Reserved Instances for Windows with SQL Server Standard or Windows with SQL Server Web.
- Reserved Instances purchased as part of a tiered discount cannot be sold in the Reserved Instance Marketplace. For more information, see the [Reserved Instance Marketplace](#) (p. 197) page.

Crossing Pricing Tiers

If your purchase crosses into a discounted pricing tier, you see multiple entries for that purchase: one for that part of the purchase charged at the regular price, and another for that part of the purchase charged at the applicable discounted rate.

The Reserved Instance service generates several Reserved Instance IDs because your purchase crossed from an undiscounted tier, or from one discounted tier to another. There is an ID for each set of reservations in a tier. Consequently, the ID returned by your purchase CLI command or API action will be different from the actual ID of the new Reserved Instances.

Reserved Instances and Consolidated Billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The hourly usage across all sub-accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Instance logic is applied to calculate the bill. For more information, see *Consolidated Billing* in [AWS Billing and Cost Management User Guide](#).

For more information about how the discounts of the Reserved Instance pricing tiers apply to consolidated billing accounts, see [Amazon EC2 Reserved Instances](#).

Reading Your Statement (Invoice)

You can find out about the charges and fees to your account by viewing the **Billing & Cost Management** page in the AWS Management Console. Choose the arrow beside your account name to access it.

- The **Dashboard** page displays charges against your account—such as upfront and one-time fees, and recurring charges. You can get both a summary or detailed list of your charges.
- The upfront charges from your purchase of third-party Reserved Instances in the Reserved Instance Marketplace are listed in the **AWS Marketplace Charges** section, with the name of the seller displayed beside it. All recurring or usage charges for these Reserved Instances are listed in the **AWS Service Charges** section.
- The **Detail** section contains information about the Reserved Instance—such as the Availability Zone, instance type, cost, and number of instances.

You can view the charges online, and you can also download a PDF rendering of the charge information.

Buying Reserved Instances

You can search for specific types of Reserved Instances to buy, adjusting your parameters until you find the exact match you're looking for.

It's important to note the following for any Reserved Instance purchase:

- **Usage fee**—With Reserved Instances, you pay for the entire term regardless of actual use.
- **Tiered discounts on purchases**—Pricing tier discounts apply only to AWS purchases. These discounts do not apply to purchases of third-party Reserved Instances. For more information, see [Understanding Reserved Instance Discount Pricing Tiers \(p. 199\)](#).
- **Cancellation of purchase**—After the purchase is confirmed, it cannot be cancelled. Before you confirm, review the details of the Reserved Instances that you plan to buy, and make sure that all the parameters are accurate. However, you may be able to sell the Reserved Instances if your needs change and you meet the requirements. For more information, see [Selling in the Reserved Instance Marketplace \(p. 204\)](#).

After you select Reserved Instances to buy, you receive a quote on the total cost of your selections. When you decide to proceed with the purchase, AWS automatically places a limit price on the purchase price, so that the total cost of your Reserved Instances does not exceed the amount you were quoted.

If the price rises or changes for any reason, you are returned to the previous screen and the purchase is not completed. If, at the time of purchase, there are offerings similar to your choice but at a lower price, AWS sells you the offerings at the lower price.

Buying Reserved Instances using the AWS Management Console

To buy Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, choose **Reserved Instances**.
3. On the **Reserved Instances** page, choose **Purchase Reserved Instances**.
4. Specify details for the Reserved Instances to purchase, and choose **Search**.

Note

The **Seller** column in the search results indicates whether the seller is a third-party. If so, the **Term** column displays non-standard terms.

5. Select the Reserved Instances that you want, enter the quantity, and choose **Add to Cart**.
6. To see a summary of the Reserved Instances that you have selected, choose **View Cart**.
7. To complete the order, choose **Purchase**.

Note

If, at the time of purchase, there are offerings similar to your choice but with a lower price, AWS will sell you the offerings at the lower price.

To apply your reservation, launch an On-Demand instance, ensuring that you match the same criteria that you specified for your Reserved Instance. AWS automatically charges you the lower hourly rate. You do not have to restart your instance.

To view transaction status using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose the **Reserved Instances** page. The status of your purchase is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`.

Buying Reserved Instances Using the Command Line Interface or API

To buy Reserved Instances using the command line or API

1. Using the AWS CLI, see [purchase-reserved-instances-offering](#)
2. Using the AWS Tools for Windows PowerShell, see [New-EC2ReservedInstance](#)
3. Using the Amazon EC2 API, see [PurchaseReservedInstancesOffering](#)

To view transaction status using the command line or API

1. Using the AWS CLI, see [describe-reserved-instances](#)
2. Using the AWS Tools for Windows PowerShell, see [Get-EC2ReservedInstance](#)
3. Using the Amazon EC2 API, see [DescribeReservedInstances](#)

Applying Reserved Instances

Reserved Instances are automatically applied to running, or potential, On-Demand instances provided that the specifications match. You can use the AWS Management Console, a command line tool, or the Amazon EC2 API to perform any of these tasks.

Note

To purchase and modify Reserved Instances, ensure that your IAM user account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Purchase—Determine how much capacity to reserve. Specify the following criteria:

- Platform (for example, Linux).

Note

To run your Reserved Instance on a specific Linux platform, you must identify the platform when you purchase the reserved capacity. Then, when you launch your instance with the intention of using the reserved capacity you purchased, you must choose the Amazon Machine Image (AMI) that runs that specific Linux platform, along with any other specifications you identified during the purchase.

- Instance type (for example, `m1.small`).
- Availability Zone in which to run the instance.
- Term (time period) over which you want to reserve capacity.
- Tenancy. You can reserve capacity for your instance to run in single-tenant hardware (`dedicated` tenancy, as opposed to `shared`). The tenancy you select must match the tenancy of the On-Demand instance you're applying, or plan to apply, the Reserved Instance to. For more information, see [Dedicated Instances](#).
- Offering (No Upfront, Partial Upfront, All Upfront).

Use—To use your Reserved Instance, launch an On-Demand instance with the same specifications as the reservation you purchased. The pricing benefits and capacity reservations automatically apply to any matching instances you have that aren't already covered by a reservation.

For more information, see [Launch Your Instance \(p. 278\)](#).

Reserved Instance States

Reserved Instances can be in one of the following states:

- `active`—The Reserved Instance is available for use.
- `payment-pending`—AWS is processing your payment for the Reserved Instance. You can use the Reserved Instance when the state becomes **active**.
- `retired`—The Reserved Instance has been terminated for any of the following reasons:
 - AWS did not receive your payment. For example, the credit card transaction did not go through.
 - The Reserved Instance term expired.

It's important to note that status information displayed in the **State** column in the **Reserved Instance** page is different from the status information displayed in the **Listing State** in the **My Listings** tab.

If you are a seller in the Reserved Instance Marketplace the **Listing State** displays the status of a reservation that's been listed in the Reserved Instance Marketplace. For more information, see [Reserved Instance Listing States \(p. 208\)](#).

Buying in the Reserved Instance Marketplace

You can purchase Amazon EC2 Reserved Instances from AWS or you can purchase from third-party sellers who own Reserved Instances that they no longer need.

For a buyer, the Reserved Instance Marketplace provides increased selection and flexibility by allowing you to search for Reserved Instances that most closely match your preferred combination of instance type, region, and duration.

For more information about the Reserved Instance Marketplace, see [Selling in the Reserved Instance Marketplace \(p. 204\)](#).

There are a few differences between Reserved Instances purchased in the Reserved Instance Marketplace and Reserved Instances purchased directly from AWS:

- **Term**—Reserved Instances that you purchase from third-party sellers have less than a full standard term remaining. Full standard terms from AWS run for one year or three years.
- **Upfront price**—Third-party Reserved Instances can be sold at different upfront prices. The usage or recurring fees remain the same as the fees set when the Reserved Instances were originally purchased from AWS.

Basic information about you is shared with the seller, for example, your ZIP code and country information.

This information enables sellers to calculate any necessary transaction taxes that they have to remit to the government (such as sales tax or value-added tax) and is provided in the form of a disbursement report. In rare circumstances, AWS might have to provide the seller with your email address, so that they can contact you regarding questions related to the sale (for example, tax questions).

For similar reasons, AWS shares the legal entity name of the seller on the buyer's purchase invoice. If you need additional information about the seller for tax or related reasons, you can [contact AWS Support](#).

Selling in the Reserved Instance Marketplace

Selling unused reservations in the Reserved Instance Marketplace provides you with the flexibility to move to new configurations when your business needs change or if you have capacity you no longer need.

As soon as you list your Reserved Instances in the Reserved Instance Marketplace, they are available for potential buyers to find. All Reserved Instances are grouped according to the duration of the term remaining and the hourly price.

To fulfill a buyer's request, AWS first sells the Reserved Instance with the lowest upfront price in the specified grouping; then it sells the Reserved Instance with the next lowest price, until the buyer's entire order is fulfilled. AWS then processes the transactions and transfers ownership of the Reserved Instances to the buyer.

You own your Reserved Instance until it's sold. After the sale, you've given up the capacity reservation and the discounted recurring fees. If you continue to use your instance, AWS charges you the On-Demand price starting from the time that your Reserved Instance was sold. You can buy more reserved capacity, or terminate your instances when your capacity reservation is sold.

Contents

- [Getting Paid \(p. 205\)](#)
- [Registering as a Seller \(p. 205\)](#)
- [Listing Your Reserved Instances \(p. 207\)](#)
- [Pricing Your Reserved Instances \(p. 209\)](#)

Getting Paid

As soon as AWS receives funds from the buyer, a message is sent to the email address associated with the account that is registered as owner of the Reserved Instance that was sold.

AWS sends an Automated Clearing House (ACH) wire transfer to your specified bank account. Typically, this transfer occurs between one to three days after your Reserved Instance has been sold. You can view the state of this disbursement by viewing your Reserved Instance disbursement report. Disbursements take place once a day. Keep in mind that you will not be able to receive disbursements until AWS has received verification from your bank. This period can take up to two weeks.

The Reserved Instance you sold will continue to appear in the results of `DescribeReservedInstances` calls you make.

You receive a cash disbursement for your Reserved Instances through a wire transfer directly into your bank account. AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace.

Note

Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other types, such as Amazon RDS and Amazon ElastiCache Reserved Instances, cannot be sold on the Reserved Instance Marketplace.

The following are important limits to note:

- **Reserved Instances can be sold after 30 days**—Reserved Instances can only be sold when you've owned them for at least 30 days. In addition, there must be at least a month remaining in the term of the Reserved Instance you are listing.
- **Listings cannot be modified**—You cannot modify your listing in the Reserved Instance Marketplace. However, you can change your listing by first cancelling it and then creating another listing with new parameters. For information, see [Listing Your Reserved Instances \(p. 207\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Your Reserved Instances \(p. 210\)](#).
- **Discounted Reserved instances cannot be sold**—Reserved Instances purchased at a reduced cost resulting from a tiering discount cannot be sold in the Reserved Instance Marketplace. For more information, see [Reserved Instance Marketplace \(p. 197\)](#).

Registering as a Seller

To be able to sell in the Reserved Instance Marketplace, your first task is to register as a seller. During registration, you need to provide the name of your business, information about your bank, and your business's tax identification number.

After AWS receives your completed seller registration, you will receive an email confirming your registration and informing you that you can get started selling in the Reserved Instance Marketplace.

Topics

- [Bank Accounts \(p. 205\)](#)
- [Tax Information \(p. 206\)](#)
- [Sharing Information with the Buyer \(p. 207\)](#)

Bank Accounts

AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address.

To register a default bank account for disbursements

1. On the [Reserved Instance Marketplace Seller Registration](#) page, sign in. If you do not yet have an AWS account you can also create one via this page.
2. On the **Manage Bank Account** page, provide the following information about the bank through which you will receive payment:

- Bank account holder name
- Routing number
- Account number
- Bank account type

Note

If you are using a corporate bank account, you are prompted to send the information about the bank account via fax (1-206-765-3424).

After registration, the bank account provided is set as the default, pending verification with the bank. It can take up to two weeks to verify a new bank account, during which time you will not be able to receive disbursements. For an established account, it usually takes about two days for disbursements to complete.

To change the default bank account for disbursement

1. On the [Reserved Instance Marketplace Seller Registration](#) page, sign in with the account you used when you registered.
2. On the **Manage Bank Account** page, add a new bank account or modify the default bank account as needed.

Tax Information

Your sale of Reserved Instances might be subject to a transactional tax, such as sales tax or value-added tax. You should check with your business's tax, legal, finance, or accounting department to determine if transaction-based taxes are applicable. You are responsible for collecting and sending the transaction-based taxes to the appropriate tax authority.

As part of the seller registration process, you have the option of completing a tax interview. We encourage you to complete this process if any of the following apply:

- You want AWS to generate a Form 1099-K.
- You anticipate having either 50 or more transactions or \$20,000 or more in sales of Reserved Instances in a calendar year. A transaction can involve one or more Reserved Instances. If you choose to skip this step during registration, and later you reach transaction 49, you will get a message saying, "You have reached the transaction limit for pre-tax. Please complete the tax interview in the [Seller Registration Portal](#)".
- You are a non-US seller. In this case, you must electronically complete Form W-8BEN.

For more information about IRS requirements and the Form 1099-K, go to the [IRS website](#).

The tax information you enter as part of the tax interview will differ depending on whether your business is a US or non-US legal entity. As you fill out the tax interview, keep in mind the following:

- Information provided by AWS, including the information in this topic, does not constitute tax, legal, or other professional advice. To find out how the IRS reporting requirements might affect your business, or if you have other questions, please contact your tax, legal, or other professional advisor.
- To fulfill the IRS reporting requirements as efficiently as possible, answer all questions and enter all information requested during the interview.

- Check your answers. Avoid misspellings or entering incorrect tax identification numbers. They can result in an invalidated tax form.

After you complete the tax registration process, AWS files Form 1099-K. You will receive a copy of it through the US mail on or before January 31 in the year following the year that your tax account reaches the threshold levels. For example, if your tax account reaches the threshold in 2016, you will receive the form in 2017.

Sharing Information with the Buyer

When you sell in the Reserved Instance Marketplace, AWS shares your company's legal name on the buyer's statement in accordance with US regulations. In addition, if the buyer calls AWS Customer Support because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS may need to provide the buyer with your email address so that the buyer can contact you directly.

For similar reasons, the buyer's ZIP code and country information are provided to the seller in the disbursement report. As a seller, you might need this information to accompany any necessary transaction taxes that you remit to the government (such as sales tax and value-added tax).

AWS cannot offer tax advice, but if your tax specialist determines that you need specific additional information, [contact AWS Support](#).

Listing Your Reserved Instances

As a registered seller, you can choose to sell one or more of your Reserved Instances, and you can choose to sell all of them in one listing or in portions. In addition, you can list any type of Reserved Instance—including any configuration of instance type, platform, region, and Availability Zone.

If you decide to cancel your listing and a portion of that listing has already been sold, the cancellation is not effective on the portion that has been sold. Only the portion of the listing not yet sold will no longer be available in the Reserved Instance Marketplace.

Lifecycle of a Listing

Now that you have created a listing, let's walk through what happens when your listing sells.

When all the instances in your listing are matched and sold, the **My Listings** tab shows that your **Total instance count** matches the count listed under **Sold**, there are no **Available** instances left for your listing, and its **Status** is `closed`.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as `closed`.

For example, let's say you created a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with a listing count of 5.

Your **My Listings** tab in the **Reserved Instance** page of the AWS Management Console will display the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = active

Let's say that a buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. As a result of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing will look in your **My Listings** tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 2
- Available = 3
- Status = active

If you decide to cancel your listing and a portion of that listing has already sold, the cancellation is not effective on the portion that has been sold. Only the portion of the listing not yet sold will no longer be available in the Reserved Instance Marketplace.

After Your Reserved Instance Is Sold

When your Reserved Instance is sold, AWS will send you an email notification. Each day that there is any kind of activity (for example, you create a listing; you sell a listing; or AWS sends funds to your account), you will receive one email notification capturing all the activities of the day.

You can track the status of your Reserved Instance listings by looking at the **My Listings** tab of the selected Reserved Instance on the **Reserved Instance** page in the AWS Management Console. The tab contains the **Listing State** as well as information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and cancelled. You can also use the `ec2-describe-reserved-instances-listings` CLI command or the `DescribeReservedInstancesListings` API call, with the appropriate filter to obtain information about your listings.

Reserved Instance Listing States

Listing State displays the current status of your listings:

The information displayed by **Listing State** is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the **State** column in the **Reserved Instances** page. This **State** information is about your reservation. For more information, see [Reserved Instance States \(p. 203\)](#).

Listing your Reserved Instances using the AWS CLI

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS CLI

1. Get a list of your Reserved Instances by calling `aws ec2 describe-reserved-instances`.
2. Specify the ID of the Reserved Instance you want to list and call `aws ec2 create-reserved-instances-listing`. You have to specify the following required parameters:

- Reserved Instance ID
- Instance count
- MONTH:PRICE

To view your listing

- Run `aws ec2 describe-reserved-instances-listings` to get details about your listing.

To cancel and change your listing

- Run `aws ec2 cancel-reserved-instances-listings` to cancel your listing.

Listing Your Reserved Instances using the Amazon EC2 API

To list a Reserved Instance in the Reserved Instance Marketplace using the Amazon EC2 API

1. Get a list of your Reserved Instances by calling `DescribeReservedInstances`. Note the ID of the Reserved Instance to list in the Reserved Instance Marketplace.
2. Create a listing using `CreateReservedInstancesListing`.

To view your listing

1. Call `DescribeReservedInstancesListings` to get details about your listing.

To cancel your listing

1. Run `CancelReservedInstancesListing`.
2. Confirm that it's cancelled by calling `DescribeReservedInstancesListings`.

Listing Your Reserved Instance using the AWS Management Console

You can list the Reserved Instances you want to sell in the Reserved Instance Marketplace by using the AWS Management Console.

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instances to list, and choose **Sell Reserved Instances**.
4. On the **Configure Your Reserved Instance Listing** page, set the number of instances to sell and the upfront price for the remaining term in the relevant columns. You can see how the value of your reservation will change over the remainder of the term by clicking the arrow next to the **Months Remaining** column.
5. If you are an advanced user and you want to customize the pricing, you can enter different values for the subsequent months. To return to the default linear price drop, choose **Reset**.
6. Choose **Continue** when you are finished configuring your listing.
7. Confirm the details of your listing, on the **Confirm Your Reserved Instance Listing** page and if you're satisfied, choose **List Reserved Instance**.

Listing State displays the current status of your listings:

- **active**—The listing is available for purchase.
- **cancelled**—The listing is cancelled and won't be available for purchase in the Reserved Instance Marketplace.
- **closed**—The Reserved Instance is not listed. A Reserved Instance might be `closed` because the sale of the listing was completed.

Pricing Your Reserved Instances

The upfront fee is the only fee that you can specify for the Reserved Instance that you're selling. The upfront fee is the one-time fee that the buyer pays when they purchase a Reserved Instance. You

cannot specify the usage fee or the recurring fee; The buyer will pay the same usage or recurring fees that were set when the reservations were originally purchased.

The following are important limits to note:

- **You can sell up to \$50,000 in Reserved Instances per year.** If you need to sell more, complete the [Request to Raise Sales Limit on Amazon EC2 Reserved Instances](#) form.
- **The minimum price is \$0.** The minimum allowed price allowed in the Reserved Instance Marketplace is \$0.00.

You cannot modify your listing directly. However, you can change your listing by first cancelling it and then creating another listing with new parameters.

You can cancel your listing at any time, as long as it's in the `active` state. You cannot cancel the listing if it's already matched or being processed for a sale. If some of the instances in your listing are matched and you cancel the listing, only the remaining unmatched instances are removed from the listing.

Setting a Pricing Schedule

Because the value of Reserved Instances decreases over time, by default AWS can set prices to decrease in equal increments month over month. However, you can set different upfront prices based on when your reservation sells.

For example, if your Reserved Instance has nine months of its term remaining, you can specify the amount you would accept if a customer were to purchase that Reserved Instance with nine months remaining, and you could set another price with five months remaining, and yet another price with one month remaining.

Modifying Your Reserved Instances

When your computing needs change, you can modify your Reserved Instances and continue to benefit from your capacity reservation.

The following topics guide you through the modification process:

Topics

- [Requirements for Modification \(p. 211\)](#)
- [Modifying the Instance Size of Your Reservations \(p. 211\)](#)
- [Submitting Modification Requests \(p. 213\)](#)

Modification does not change the remaining term of your Reserved Instances; their end dates remain the same. There is no fee, and you do not receive any new bills or invoices. Modification is separate from purchasing and does not affect how you use, purchase, or sell Reserved Instances. You can modify your whole reservation, or just a subset, in one or more of the following ways:

- Switch Availability Zones within the same region
- Change between EC2-VPC and EC2-Classic
- Change the instance size within the same instance type

Availability Zone and network platform modifications are supported for all product platform types (Linux and Windows). Instance type modifications are supported only for the Linux platform types. However, due to licensing differences, it is not possible to change the instance type of RedHat or SUSE Linux Reserved Instances. For more information about RedHat and SUSE pricing, see [Amazon EC2 Reserved Instance Pricing](#).

After modification, the pricing benefit of the Reserved Instances is applied only to instances that match the new parameters. Instances that no longer match the new parameters are charged at the On-Demand rate unless your account has other applicable reservations.

Requirements for Modification

Amazon EC2 processes your modification request if there is sufficient capacity for your target configuration, and if the following conditions are met.

Your modified Reserved Instances must be:

- Active
- Not pending another modification request
- Not listed in the Reserved Instance Marketplace
- Terminating in the same hour (but not minutes or seconds)

Your modification request must be:

- A unique combination of Availability Zone, instance type, and network platform attributes
- A match between the instance size footprint of the active reservation and the target configuration

If your Reserved Instances are not in the active state or cannot be modified, an API call returns an error and the **Modify Reserved Instances** button in the AWS Management Console is not enabled. If you select multiple Reserved Instances for modification and one or more are for a product platform that does not allow instance type modification, the **Modify Reserved Instances** page does not show the option of changing the instance type of any of the selected Reserved Instances. For more information, see [Modifying the Instance Size of Your Reservations](#) (p. 211).

You may modify your reservations as frequently as you like; however, you cannot submit a modification request for reservations that are pending a previous modification request. Also, you cannot change or cancel a pending modification request after you submit it. After the modification has completed successfully, you can submit another modification request to roll back any changes you made. For more information, see [Determining the Status of Your Modification](#) (p. 214).

To modify Reserved Instances that are listed in the Reserved Instance Marketplace, cancel the listing, request modification, and then list them again. In addition, you cannot modify an offering before or at the same time that you purchase it. For more information, see [Reserved Instance Marketplace](#) (p. 197).

Modifying the Instance Size of Your Reservations

You can adjust the instance size of your Reserved Instances if you have Amazon Linux reservations in an instance type with multiple sizes. Keep in mind that instance size modifications are allowed only if other attributes—such as region, utilization type, tenancy, product, end date and hour—match and if capacity is available. It is not possible to modify the instance size of Windows Reserved Instances.

Note

Instances are grouped by family (based on storage, or CPU capacity); type (designed for specific use cases); and size. For example, the `c4` instance type is in the Compute optimized instance family and is available in multiple sizes. While `c3` instances are in the same family, you can't modify `c4` instances into `c3` instances because they have different hardware specifications. For more information, see [Amazon EC2 Instance Types](#).

For information about the modification process and steps, see [Submitting Modification Requests](#) (p. 213).

The following instances cannot be modified because there are no other sizes available.

- `t1.micro`
- `cc1.4xlarge`
- `cc2.8xlarge`
- `cg1.8xlarge`
- `cr1.8xlarge`
- `hi1.4xlarge`
- `hs1.8xlarge`
- `g2.2xlarge`

Your request is successful if the capacity exists and the modification does not change the instance size footprint of your Reserved Instances.

Understanding the Instance Size Footprint

Each Reserved Instance has an instance size footprint, which is determined by the normalization factor of the instance type and the number of instances in the reservation. In the Amazon EC2 console, the footprint is measured in units.

The normalization factor is based on size within the instance type (e.g., the `m1` instance family), and is only meaningful within the same instance type; instance types cannot be modified from one type to another. The following table illustrates the normalization factor that applies within an instance type.

Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
10xlarge	80

A modification request is not processed if the footprint of the target configuration does not match the size of the original configuration.

To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. For example, an `m1.medium` has a normalization factor of 2 so a reservation for four `m1.medium` instances has a footprint of 8 units.

You can allocate your reservations into different instance sizes across the same instance type as long as the instance size footprint of your reservation remains the same. For example, you can divide a reservation for one `m1.large` (1 x 4) instance into four `m1.small` (4 x 1) instances, or you can combine a reservation for four `m1.small` instances into one `m1.large` instance. However, you cannot change your reservation for two `m1.small` (2 x 1) instances into one `m1.large` (1 x 4) instance because the existing instance size footprint of your current reservation is smaller than the proposed reservation.

For more information, see [Amazon EC2 Instance Types](#).

Submitting Modification Requests

AWS provides you with several ways to view and work with modification requests: You can use the AWS Management Console, interact directly with the Amazon EC2 API, or use the command line interface.

Topics

- [AWS Management Console](#) (p. 213)
- [Command Line Interface](#) (p. 214)
- [Amazon EC2 API](#) (p. 214)
- [Determining the Status of Your Modification](#) (p. 214)

AWS Management Console

Each target configuration row on the **Modify Reserved Instances** page keeps track of the number of instances for the current instance type (**Count**) and the instance size footprint of your reservation relative to its instance type (**Units**). For more information, see [Understanding the Instance Size Footprint](#) (p. 212).

The allocated total is displayed in red if you have specified either more or fewer Reserved Instances than are available for modification. The total changes to green and you can choose **Continue** after you have specified changes for all the Reserved Instances that were available for modification.

When you modify a subset of your reservation, Amazon EC2 splits your original Reserved Instances into two or more new Reserved Instances. For example, if you have reservations for 10 instances in us-east-1a, and decide to move 5 instances to us-east-1b, the modification request results in two new reservations—one for 5 instances in us-east-1a (the original Availability Zone), and the other for 5 instances in us-east-1b.

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Modify Reserved Instances**.

Note

The first entry in the modification table is the original, unmodified reservation. To modify the attributes of all reservations, choose new specifications from the menus. To modify or split only some of your reservations, add an additional line for each change.

3. Choose **Add** for each additional attribute change and enter the number of reservations to modify in the **Count** field.
 - To change the Availability Zone, select a value in the **Availability Zone** list.
 - To change the network platform, select a value in the **Network** list.
 - To change the instance type, select a value in the **Instance Type** list.
4. To delete a specified attribute, select **X** for that row.

Note

If the **Modify Reserved Instances** page contains only one row for attribute changes, you cannot delete that row. To modify multiple Reserved Instance attributes, first add a row for the new specifications and then delete the original row.

5. Choose **Continue**.
6. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**. If you change your mind at any point, choose **Cancel** to exit the wizard.

Command Line Interface

You can complete modification tasks programmatically by using the AWS CLI ([modify-reserved-instances](#)), the AWS Tools for Windows PowerShell ([Edit-EC2ReservedInstance](#)) the Amazon EC2 API ([ModifyReservedInstances](#)), and the [AWS SDK for Java](#).

Amazon EC2 API

You can use the [ModifyReservedInstances](#) action to modify your Reserved Instances. For more information, see [Amazon EC2 API Reference](#).

Determining the Status of Your Modification

You can determine the status of your modification request by looking at the **state** of the Reserved Instances that you are modifying. The state returned shows your request as `in-progress`, `fulfilled`, or `failed`. Use the following resources to get this information:

- The **State** field in the AWS Management Console
- The [DescribeReservedInstancesModifications](#) API action
- The [describe-reserved-instances-modifications](#) AWS CLI command
- The [Get-EC2ReservedInstancesModifications](#) AWS Tools for Windows PowerShell command

The following table illustrates the possible **State** values in the AWS Management Console

State	Description
active (pending modification)	Transition state for original Reserved Instances.
retired (pending modification)	Transition state for original Reserved Instances while new Reserved Instances are being created.
retired	Reserved Instances successfully modified and replaced.
active	New Reserved Instances created from a successful modification request. -Or- Original Reserved Instances after a failed modification request.

Note

If you use the [DescribeReservedInstancesModifications](#) API action, the status of your modification request should show *processing*, *fulfilled*, or *failed*.

If your modification request succeeds:

- The modified reservation becomes effective immediately and the pricing benefit is applied to the new instances beginning at the hour of the modification request. For example, if you successfully modify your reservations at 9:15PM, the pricing benefit transfers to your new instance at 9:00PM. (You can get the `effective_date` of the modified Reserved Instances by using the [DescribeReservedInstances](#) API action or the [describe-reserved-instances](#) command (AWS CLI).
- The original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Reserved Instance. If you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.

- The modified reservation lists a \$0 fixed price and not the fixed price of the original reservation.

Note

The fixed price of the modified reservation does not affect the discount pricing tier calculations applied to your account, which are based on the fixed price of the original reservation.

If your modification request fails:

- Your Reserved Instances maintain their original configuration.
- Your Reserved Instances are immediately available for another modification request.

For more information about why some Reserved Instances cannot be modified, see [Requirements for Modification](#) (p. 211).

Troubleshooting Modification Requests

If the target configuration settings that you requested were unique, you receive a message that your request is being processed. At this point, Amazon EC2 has only determined that the parameters of your modification request are valid. Your modification request can still fail during processing due to unavailable capacity.

In some situations, you might get a message indicating incomplete or failed modification requests instead of a confirmation. Use the information in such messages as a starting point for resubmitting another modification request.

Not all selected Reserved Instances can be processed for modification

Amazon EC2 identifies and lists the Reserved Instances that cannot be modified. If you receive a message like this, go to the **Reserved Instances** page in the AWS Management Console and check the information details about these capacity reservations.

Error in processing your modification request

You submitted one or more Reserved Instances for modification and none of your requests can be processed. Depending on the number of reservations you are modifying, you can get different versions of the message.

Amazon EC2 displays the reasons why your request cannot be processed. For example, you might have specified the same target configuration—a combination of Availability Zone and platform—for one or more subsets of the Reserved Instances you are modifying. Try submitting these modification requests again, but ensure that instance details of the reservations match, and that the target configurations for all subsets being modified are unique.

Scheduled Reserved Instances

Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time that the instances are scheduled, even if you do not use them.

Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.

If you require a capacity reservation on a continuous basis, Reserved Instances might meet your needs and decrease costs. For more information, see [Reserved Instances](#) (p. 194). If you are flexible about when your instances run, Spot instances might meet your needs and decrease costs. For more information, see [Spot Instances](#) (p. 219).

Contents

- [How Scheduled Instances Work \(p. 216\)](#)
- [Purchasing a Scheduled Instance \(p. 216\)](#)
- [Launching a Scheduled Instance \(p. 217\)](#)
- [Scheduled Instance Limits \(p. 218\)](#)

How Scheduled Instances Work

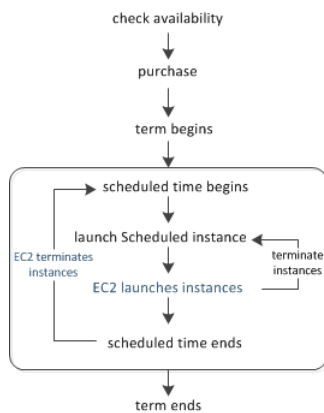
Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network (EC2-Classic or EC2-VPC).

To get started, you must search for an available schedule. You can search across multiple pools or a single pool. After you locate a suitable schedule, purchase it.

You must launch your Scheduled Instances during their scheduled time periods, using a launch configuration that matches the following attributes of the schedule that you purchased: instance type, Availability Zone, network, and platform. When you do so, Amazon EC2 launches EC2 instances on your behalf, based on the specified launch specification. Amazon EC2 must ensure that the EC2 instances have terminated by the end of the current scheduled time period so that the capacity is available for any other Scheduled Instances it is reserved for. Therefore, Amazon EC2 terminates the EC2 instances three minutes before the end of the current scheduled time period.

You can't stop or reboot Scheduled Instances, but you can terminate them manually as needed. If you terminate a Scheduled Instance before its current scheduled time period ends, you can launch it again after a few minutes. Otherwise, you must wait until the next scheduled time period.

The following diagram illustrates the lifecycle of a Scheduled Instance.



Purchasing a Scheduled Instance

To purchase a Scheduled Instance, you can use the Scheduled Reserved Instances Reservation Wizard.

Warning



After you purchase a Scheduled Instance, you can't cancel, modify, or resell your purchase.

To purchase a Scheduled Instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.

3. Choose **Purchase Scheduled Instances**.
4. On the **Find available schedules** page, do the following:
 - a. Under **Create a schedule**, select the starting date from **Starting on**, the schedule recurrence (daily, weekly, or monthly) from **Recurring**, and the minimum duration from **for duration**. Note that the console ensures that you specify a value for the minimum duration that meets the minimum required utilization for your Scheduled Instance (1,200 hours per year).

Create a schedule

Starting on  for duration  hours
 +/- 2 hours
Recurring

- b. Under **Instance details**, select the operating system and network from **Platform**. To narrow the results, select one or more instance types from **Instance type** or one or more Availability Zones from **Availability Zone**.

Instance details

Platform Instance type
Availability Zone

- c. Choose **Find schedules**.
 - d. Under **Available schedules**, select one or more schedules. For each schedule that you select, set the quantity of instances and choose **Add to Cart**.
 - e. Your cart is displayed at the bottom of the page. When you are finished adding and removing schedules from your cart, choose **Review and purchase**.
5. On the **Review and purchase** page, verify your selections and edit them as needed. When you are finished, choose **Purchase**.

To purchase a Scheduled Instance using the AWS CLI

Use the [describe-scheduled-instance-availability](#) command to list the available schedules that meet your needs, and then use the [purchase-scheduled-instances](#) command to complete the purchase.

Launching a Scheduled Instance

After you purchase a Scheduled Instance, it is available for you to launch during its scheduled time periods.

To launch a Scheduled Instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.
3. Select the Scheduled Instance and choose **Launch Scheduled Instances**.
4. On the **Configure** page, complete the launch specification for your Scheduled Instances and choose **Review**.

Important

The launch specification must match the instance type, Availability Zone, network, and platform of the schedule that you purchased.

5. On the **Review** page, verify the launch configuration and modify it as needed. When you are finished, choose **Launch**.

To launch a Scheduled Instance using the AWS CLI

Use the [describe-scheduled-instances](#) command to list your Scheduled Instances, and then use the [run-scheduled-instances](#) command to launch each Scheduled Instance during its scheduled time periods.

Scheduled Instance Limits

Scheduled Instances are subject to the following limits:

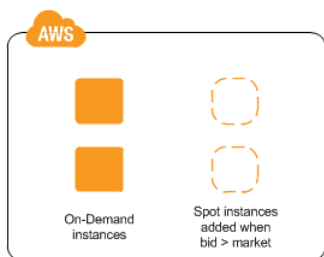
- The following are the only supported instance types: C3, C4, M4, and R3.
- The required term is 365 days (one year).
- The minimum required utilization is 1,200 hours per year.
- You can purchase a Scheduled Instance up to three months in advance.

Spot Instances

Spot instances enable you to bid on unused EC2 instances, which can lower your Amazon EC2 costs significantly. The hourly price for a Spot instance (of each instance type in each Availability Zone) is set by Amazon EC2, and fluctuates depending on the supply of and demand for Spot instances. Your Spot instance runs whenever your bid exceeds the current market price.

Spot instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see [Amazon EC2 Spot Instances](#).

The key differences between Spot instances and On-Demand instances are that Spot instances might not start immediately, the hourly price for Spot instances varies based on demand, and Amazon EC2 can terminate an individual Spot instance as the hourly price for or availability of Spot instances changes. One strategy is to launch a core group of On-Demand instances to maintain a minimum level of guaranteed compute resources for your applications, and supplement them with Spot instances when the opportunity arises.



Another strategy is to launch Spot instances with a required duration (also known as Spot blocks), which are not interrupted due to changes in the Spot price. For more information, see [Specifying a Duration for Your Spot Instances \(p. 231\)](#).

Concepts

Before you get started with Spot instances, you should be familiar with the following concepts:

- *Spot instance pool*—A set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC).
- *Spot price*—The current market price of a Spot instance per hour, which is set by Amazon EC2 based on the last fulfilled bid. You can also retrieve the Spot price history.
- *Spot instance request (or Spot bid)*—Provides the maximum price (bid price) that you are willing to pay per hour for a Spot instance. When your bid price exceeds the Spot price, Amazon EC2 fulfills your request. Note that a Spot instance request is either *one-time* or *persistent*. Amazon EC2 automatically resubmits a persistent Spot request after the Spot instance associated with the request is terminated. Your Spot instance request can optionally specify a duration for the Spot instances.
- *Spot fleet*—A set of Spot instances that is launched based on criteria that you specify. The Spot fleet selects the Spot instance pools that meet your needs and launches Spot instances to meet the target capacity for the fleet. By default Spot fleets are set to *maintain* target capacity by launching replacement instances after Spot instances in the fleet are terminated. They can also be submitted as a one-time *request* which does not persist once instances have been terminated.
- *Spot instance interruption*—Amazon EC2 terminates your Spot instance when the Spot price exceeds your bid price or there are no longer any unused EC2 instances. Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates.

- *Bid status*—Provides detailed information about the current state of your Spot bid.

How to Get Started

The first thing you need to do is get set up to use Amazon EC2. It can also be helpful to have experience launching On-Demand instances before launching Spot instances.

Get Up and Running

- [Setting Up with Amazon EC2](#) (p. 19)
- [Getting Started with Amazon EC2 Linux Instances](#) (p. 26)

Spot Basics

- [How Spot Instances Work](#) (p. 221)
- [How Spot Fleet Works](#) (p. 224)

Working with Spot Instances

- [Preparing for Interruptions](#) (p. 260)
- [Creating a Spot Instance Request](#) (p. 232)
- [Getting Bid Status Information](#) (p. 258)

Working with Spot Fleets

- [Spot Fleet Prerequisites](#) (p. 238)
- [Creating a Spot Fleet Request](#) (p. 240)

Related Services

You can provision Spot instances directly using Amazon EC2. You can also provision Spot instances using other services in AWS. For more information, see the following documentation.

Auto Scaling and Spot instances

You can create launch configurations with a bid price so that Auto Scaling can launch Spot instances. For more information, see [Launching Spot instances in Your Auto Scaling Group](#) in the *Auto Scaling User Guide*.

Amazon EMR and Spot instances

There are scenarios where it can be useful to run Spot instances in an Amazon EMR cluster. For more information, see [Lower Costs with Spot Instances](#) in the *Amazon EMR Developer Guide*.

AWS CloudFormation Templates

AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include a Spot price. For more information, see [EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration](#).

AWS SDK for Java

You can use the Java programming language to manage your Spot instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#) and [Tutorial: Advanced Amazon EC2 Spot Request Management](#).

AWS SDK for .NET

You can use the .NET programming environment to manage your Spot instances. For more information, see [Tutorial: Amazon EC2 Spot instances](#).

Pricing

You pay the Spot price for Spot instances, which is set by Amazon EC2 and fluctuates periodically depending on the supply of and demand for Spot instances. If your bid price exceeds the current Spot price, Amazon EC2 fulfills your request and your Spot instances run until either you terminate them or the Spot price increases above your bid price.

Everyone pays that same Spot price for that period, regardless of whether their bid price was higher. You never pay more than your bid price per hour, and often pay less per hour. For example, if you bid \$0.25 per hour, and the Spot price is \$0.20 per hour, you only pay \$0.20 per hour. If the Spot price drops, you pay the new, lower price. If the Spot price rises, you pay the new price if it is equal to or less than your bid price. If the Spot price rises above your bid price, then your Spot instance is interrupted.

At the start of each instance hour, you are charged based on the Spot price. If your Spot instance is interrupted in the middle of an instance hour because the Spot price exceeded your bid, you are not charged for the hour of use that was interrupted. However, if you terminate your Spot instance in the middle of an instance hour, you are charged for the hour.

Note that Spot instances with a predefined duration use a fixed hourly price that remains in effect for the Spot instance while it runs.

View Prices

To view the current (updated every five minutes) lowest Spot price per region and instance type, see the [Spot Instances Pricing](#) page.

To view the Spot price history for the past three months, use the Amazon EC2 console or the [describe-spot-price-history](#) command (AWS CLI). For more information, see [Spot Instance Pricing History](#) (p. 229).

Note that we independently map Availability Zones to codes for each AWS account. Therefore, you can get different results for the same Availability Zone code (for example, `us-west-2a`) between different accounts.

View Billing

To review your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill. For more information, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

How Spot Instances Work

To use Spot instances, create a *Spot instance request* or a *Spot fleet request*. The request includes the maximum price that you are willing to pay per hour per instance (your bid price), and other constraints such as the instance type and Availability Zone. If your bid price is greater than the current Spot price for the specified instance, and the specified instance is available, your request is fulfilled immediately. Otherwise, the request is fulfilled whenever the Spot price falls below your bid price or the specified instance becomes available. Spot instances run until you terminate them or until Amazon EC2 must terminate them (also known as a *Spot instance interruption*).

When you use Spot instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot instance when the Spot price rises above your bid price, when the demand for Spot instances rises, or when the supply of Spot instances decreases. When Amazon EC2 marks a Spot instance for termination, it provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates. Note that you can't enable termination protection for Spot instances. For more information, see [Spot Instance Interruptions](#) (p. 260).

Note that you can't stop and start an Amazon EBS-backed instance if it is a Spot instance, but you can reboot or terminate it.

Shutting down a Spot instance on OS-level results in the Spot instance being terminated. It is not possible to change this behavior.

Contents

- [Supply and Demand in the Spot Market \(p. 222\)](#)
- [Launching Spot Instances in a Launch Group \(p. 223\)](#)
- [Launching Spot Instances in an Availability Zone Group \(p. 223\)](#)
- [Launching Spot Instances in a VPC \(p. 224\)](#)

Supply and Demand in the Spot Market

AWS continuously evaluates how many Spot instances are available in each Spot instance pool, monitors the bids that have been made for each pool, and provisions the available Spot instances to the highest bidders. The Spot price for a pool is set to the lowest fulfilled bid for that pool. Therefore, the Spot price is the price above which you must bid to fulfill a Spot request for a single Spot instance immediately.

For example, suppose that you create a Spot instance request, and that the corresponding Spot instance pool has only five Spot instances for sale. Your bid price is \$0.10, which is also the current Spot price. The following table shows the current bids, ranked in descending order. Bids 1-5 are fulfilled. Bid 5, being the last fulfilled bid, sets the Spot price at \$0.10. Bid 6 is unfulfilled. Bids 3-5, which share the same bid price of \$0.10, are ranked in random order.

Bid	Bid price	Current Spot price	Notes
1	\$1.00	\$0.10	
2	\$1.00	\$0.10	
3	\$0.10	\$0.10	
4	\$0.10	\$0.10	Your bid
5	\$0.10	\$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the same Spot price for the period.
---	---		Spot capacity cutoff
6	\$0.05		

Now, let's say that the size of this pool drops to 3. Bids 1-3 are fulfilled. Bid 3, the last fulfilled bid, sets the Spot price at \$0.10. Bids 4-5, which also are \$0.10, are unfulfilled. As you can see, even though the Spot price didn't change, two of the bids, including your bid, are no longer fulfilled because the Spot supply decreased.

Bid	Bid price	Current Spot price	Notes
1	\$1.00	\$0.10	
2	\$1.00	\$0.10	
3	\$0.10	\$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Spot Instances**

Bid	Bid price	Current Spot price	Notes
			same Spot price for the period.
— — —	— — —		Spot capacity cutoff
4	\$0.10		Your bid
5	\$0.10		
6	\$0.05		

To fulfill a Spot request for a single instance from this pool, you must bid above the current Spot price of \$0.10. If you bid \$0.101, your request will be fulfilled, the Spot instance for bid 3 would be interrupted, and the Spot price would become \$0.101. If you bid \$2.00, the Spot instance for bid 3 would be interrupted and the Spot price would become \$1.00 (the price for bid 2).

Keep in mind that no matter how high you bid, you can never get more than the available number of Spot instances in a Spot instance pool. If the size of the pool drops to zero, then all the Spot instances from that pool would be interrupted.

Launching Spot Instances in a Launch Group

Specify a launch group in your Spot instance request to tell Amazon EC2 to launch a set of Spot instances only if it can launch them all. In addition, if the Spot service must terminate one of the instances in a launch group (for example, if the Spot price rises above your bid price), it must terminate them all. However, if you terminate one or more of the instances in a launch group, Amazon EC2 does not terminate the remaining instances in the launch group.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled. It can also increase the chance that your Spot instances will be terminated.

If you create another successful Spot instance request that specifies the same (existing) launch group as an earlier successful request, then the new instances are added to the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group are terminated, which includes instances launched by the first and second requests.

Launching Spot Instances in an Availability Zone Group

Specify an Availability Zone group in your Spot instance request to tell the Spot service to launch a set of Spot instances in the same Availability Zone. Note that Amazon EC2 need not terminate all instances in an Availability Zone group at the same time. If Amazon EC2 must terminate one of the instances in an Availability Zone group, the others remain running.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled.

If you specify an Availability Zone group but don't specify an Availability Zone in the Spot instance request, the result depends on whether you specified the EC2-Classical network, a default VPC, or a nondefault VPC. For more information about EC2-Classical and EC2-VPC, see [Supported Platforms](#) (p. 549).

EC2-Classical

Amazon EC2 finds the lowest-priced Availability Zone in the region and launches your Spot instances in that Availability Zone if the lowest bid for the group is higher than the current Spot price in that Availability Zone. Amazon EC2 waits until there is enough capacity to launch your Spot instances together, as long as the Spot price remains lower than the lowest bid for the group.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet, or if you don't specify a subnet, it selects an Availability Zone and its default subnet, but it might not be the lowest-priced Availability Zone. If you deleted the default subnet for an Availability Zone, then you must specify a different subnet.

Nondefault VPC

Amazon EC2 uses the Availability Zone for the specified subnet.

Launching Spot Instances in a VPC

To take advantage of the features of EC2-VPC when you use Spot instances, specify in your Spot request that your Spot instances are to be launched in a VPC. You specify a subnet for your Spot instances the same way that you specify a subnet for your On-Demand instances.

The process for making a Spot instance request that launches Spot instances in a VPC is the same as the process for making a Spot instance request that launches Spot instances in EC2-Classic—except for the following differences:

- You should base your bid on the Spot price history of Spot instances in a VPC.
- [Default VPC] If you want your Spot instance launched in a specific low-priced Availability Zone, you must specify the corresponding subnet in your Spot instance request. If you do not specify a subnet, Amazon EC2 selects one for you, and the Availability Zone for this subnet might not have the lowest Spot price.
- [Nondefault VPC] You must specify the subnet for your Spot instance.

How Spot Fleet Works

A *Spot fleet* is a collection, or fleet, of Spot instances. The Spot fleet attempts to launch the number of Spot instances that are required to meet the target capacity that you specified in the Spot fleet request. The Spot fleet also attempts to maintain its target capacity fleet if your Spot instances are interrupted due to a change in Spot prices or available capacity.

A *Spot instance pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC). When you make a Spot fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot fleet selects the Spot instance pools that are used to fulfill the request, based on the launch specifications included in your Spot fleet request, and the configuration of the Spot fleet request. The Spot instances come from the selected pools.

Contents

- [Spot Fleet Allocation Strategy \(p. 224\)](#)
- [Spot Price Overrides \(p. 225\)](#)
- [Spot Fleet Instance Weighting \(p. 225\)](#)
- [Walkthrough: Using Spot Fleet with Instance Weighting \(p. 226\)](#)

Spot Fleet Allocation Strategy

The allocation strategy for your Spot fleet determines how it fulfills your Spot fleet request from the possible Spot instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot fleet request:

`lowestPrice`

The Spot instances come from the pool with the lowest price. This is the default strategy.

`diversified`

The Spot instances are distributed across all pools.

Choosing an Allocation Strategy

You can optimize your Spot fleets based on your use case.

If your fleet is small or runs for a short time, the probability that your Spot instances will be interrupted is low, even with all the instances in a single Spot instance pool. Therefore, the `lowestPrice` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot instances across multiple pools. For example, if your Spot fleet request specifies 10 pools and a target capacity of 100 instances, the Spot fleet launches 10 Spot instances in each pool. If the Spot price for one pool increases above your bid price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

Note that with the `diversified` strategy, the Spot fleet does not launch Spot instances into any pools with a Spot price that is higher than the [On-Demand price](#).

Maintaining Target Capacity

After Spot instances are terminated due to a change in the Spot price or available capacity of a Spot instance pool, the Spot fleet launches replacement Spot instances. If the allocation strategy is `lowestPrice`, the Spot fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is `diversified`, the Spot fleet distributes the replacement Spot instances across the remaining pools.

Spot Price Overrides

Each Spot fleet request must include a global Spot price. By default, the Spot fleet uses this price as the bid price for each of its launch specifications.

You can optionally specify a Spot price in one or more launch specifications. This bid price is specific to the launch specification. If a launch specification includes a specific Spot price, the Spot fleet uses this price as the bid price for that launch specification, overriding the global Spot price. Note that any other launch specifications that do not include a specific Spot price still use the global Spot price.

Spot Fleet Instance Weighting

When you request a fleet of Spot instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your bid price for each Spot instance pool accordingly using *instance weighting*.

By default, the Spot price that you specify represents your bid price *per instance hour*. When you use the instance weighting feature, the Spot price that you specify represents your bid price *per unit hour*. You can calculate your bid price per unit hour by dividing your bid price for an instance type by the number of units that it represents. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

The following table includes examples of calculations to determine the bid price per unit for a Spot fleet request with a target capacity of 10.

Instance type	Instance weight	Spot price per instance hour	Spot price per unit hour	Number of instances launched
r3.xlarge	2	\$0.05	.025	5

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Spot Instances**

Instance type	Instance weight	Spot price per instance hour	Spot price per unit hour	Number of instances launched
			(.05 divided by 2)	(10 divided by 2)
r3.xlarge	8	\$0.10	.0125 (.10 divided by 8)	2 (10 divided by 8, result rounded up)

Use Spot fleet instance weighting as follows to provision the target capacity you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your Spot fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the bid price per unit.
3. For each launch configuration, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance Weighting Example

Consider a Spot fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest Spot price per unit (Spot price for `r3.2xlarge` per instance hour divided by 6), the Spot fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest Spot price per unit (Spot price for `c3.xlarge` per instance hour divided by 5), the Spot fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance Weighting and Allocation Strategy

Consider a Spot fleet request with the following configuration:

- A target capacity of 30
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The Spot fleet would launch four instances (30 divided by 8, result rounded up). With the `lowestPrice` strategy, all four instances come from the pool that provides the lowest Spot price per unit. With the `diversified` strategy, the Spot fleet launches 1 instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest Spot price per unit.

Walkthrough: Using Spot Fleet with Instance Weighting

This walkthrough uses a fictitious company called Example Corp to illustrate the process of bidding for a Spot fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the following requirements for their Spot fleet.

Instance Types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target Capacity in Units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as 1 unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their Spot fleet request to 20.

Instance Weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their Spot fleet request.

Bid Price Per Unit Hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their bid price. They could also use recent Spot prices, or a combination of the two. To calculate bid price per unit hour, they divide their starting bid price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could enter a global bid price per unit hour of \$0.7 and be competitive for all three instance types. They could also enter a global bid price per unit hour of \$0.7 and a specific bid price per unit hour of \$0.9 in the `r3.8xlarge` launch specification. Depending on the strategy for provisioning their Spot fleet, Example Corp could bid lower to further reduce costs, or bid higher to reduce the probability of interruption.

Verifying Permissions

Before creating a Spot fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see [Spot Fleet Prerequisites](#) (p. 238).

Creating the Request

Example Corp creates a file, `config.json`, with the following configuration for its Spot fleet request:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 1
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.4xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.8xlarge",
      "SubnetId": "subnet-482e4972",
      "SpotPrice": "0.90",
      "WeightedCapacity": 4
    }
  ]
}
```

Example Corp creates the Spot fleet request using the following `request-spot-fleet` command:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see [Spot Fleet Requests](#) (p. 237).

Fulfillment

The allocation strategy determines which Spot instance pools your Spot instances come from.

With the `lowestPrice` strategy (which is the default strategy), the Spot instances come from the pool with the lowest Spot price per unit at the time of fulfillment. To provide 20 units of capacity, the Spot fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the `diversified` strategy, the Spot instances would come from all three pools. The Spot fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge`

instances (which provide 6 units), and 2 `r3.xlarge` instances (which provide 8 units), for a total of 20 units.

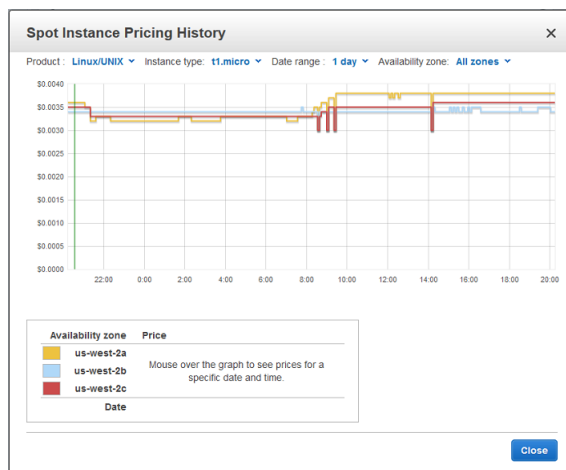
Spot Instance Pricing History

The Spot price represents the price above which you have to bid to guarantee that a single Spot request is fulfilled. When your bid price is above the Spot price, Amazon EC2 launches your Spot instance, and when the Spot price rises above your bid price, Amazon EC2 terminates your Spot instance. You can bid above the current Spot price so that your Spot request is fulfilled quickly. However, before you specify a bid price for your Spot instance, we recommend that you review the Spot price history. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

Using the Spot price history as a guide, you can select a bid price that would have met your needs in the past. For example, you can determine which bid price that would have provided 75 percent uptime in the time range you viewed. However, keep in mind that the historical trends are not a guarantee of future results. Spot prices vary based on real-time supply and demand, and the conditions that generated certain patterns in the Spot price might not occur in the future.

To view the Spot price history using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. If you are new to Spot instances, you see a welcome page; choose **Get started**, scroll to the bottom of the screen, and then choose **Cancel**.
4. Choose **Pricing History**. By default, the page displays a graph of the data for Linux `t1.micro` instances in all Availability Zones over the past day. Move your mouse over the graph to display the prices at specific times in the table below the graph.



5. (Optional) To review the Spot price history for a specific Availability Zone, select an Availability Zone from the list. You can also select a different product, instance type, or date range.

To view the Spot price history using the command line

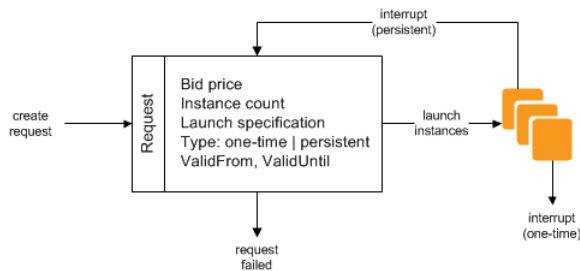
You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-spot-price-history` (AWS CLI)
- `Get-EC2SpotPriceHistory` (AWS Tools for Windows PowerShell)

Spot Instance Requests

To use Spot instances, you create a Spot instance request that includes the number of instances, the instance type, the Availability Zone, and the maximum price that you are willing to pay per instance hour (your bid). If your bid exceeds the current Spot price, Amazon EC2 fulfills your request immediately. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

The following illustration shows how Spot requests work. Notice that the action taken for a Spot instance interruption depends on the request type (one-time or persistent). If the request is a persistent request, the request is opened again after your Spot instance is terminated.



Contents

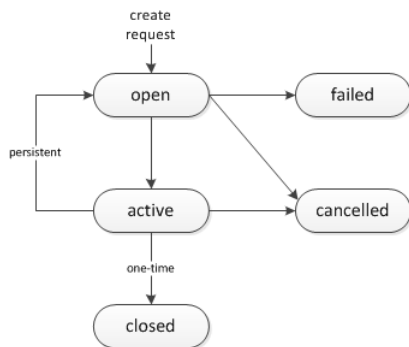
- [Spot Instance Request States](#) (p. 230)
- [Specifying a Duration for Your Spot Instances](#) (p. 231)
- [Creating a Spot Instance Request](#) (p. 232)
- [Finding Running Spot Instances](#) (p. 234)
- [Tagging Spot Instance Requests](#) (p. 234)
- [Cancelling a Spot Instance Request](#) (p. 235)
- [Spot Request Example Launch Specifications](#) (p. 235)

Spot Instance Request States

A Spot instance request can be in one of the following states:

- `open`—The request is waiting to be fulfilled.
- `active`—The request is fulfilled and has an associated Spot instance.
- `failed`—The request has one or more bad parameters.
- `closed`—The Spot instance was interrupted or terminated.
- `cancelled`—You cancelled the request, or the request expired.

The following illustration represents the transitions between the request states. Notice that the transitions depend on the request type (one-time or persistent).



A one-time Spot instance request remains active until Amazon EC2 launches the Spot instance, the request expires, or you cancel the request. If the Spot price rises above your bid price, your Spot instance is terminated and the Spot instance request is closed.

A persistent Spot instance request remains active until it expires or you cancel it, even if the request is fulfilled. For example, if you create a persistent Spot instance request for one instance when the Spot price is \$0.25, Amazon EC2 launches your Spot instance if your bid price is above \$0.25. If the Spot price rises above your bid price, your Spot instance is terminated; however, the Spot instance request is open again and Amazon EC2 launches a new Spot instance when the Spot price falls below your bid price.

You can track the status of your Spot instance requests, as well as the status of the Spot instances launched, through the bid status. For more information, see [Spot Bid Status \(p. 255\)](#).

Specifying a Duration for Your Spot Instances

Amazon EC2 does not terminate Spot instances with a specified duration (also known as Spot blocks) when the Spot price changes. This makes them ideal for jobs that take a finite time to complete, such as batch processing, encoding and rendering, modeling and analysis, and continuous integration.

You can specify a duration of 1, 2, 3, 4, 5, or 6 hours. The price that you pay depends on the specified duration. To view the current prices for a 1 hour duration or a 6 hour duration, see [Spot Instance Prices](#). You can use these prices to estimate the cost of the 2, 3, 4, and 5 hour durations. When a request with a duration is fulfilled, the price for your Spot instance is fixed, and this price remains in effect until the instance terminates.

When you specify a duration in your Spot request, the duration period for each Spot instance starts as soon as the instance receives its instance ID. The Spot instance runs until you terminate it or the duration period ends. At the end of the duration period, Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates.

To launch Spot instances with a specified duration using the console

Select the appropriate request type. For more information, see [Creating a Spot Instance Request \(p. 232\)](#).

To launch Spot instances with a specified duration using the AWS CLI

To specify a duration for your Spot instances, include the `--block-duration-minutes` option with the `request-spot-instances` command. For example, the following command creates a Spot request that launches Spot instances that run for two hours:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5  
--block-duration-minutes 120 --type "one-time" --launch-specification  
file://specification.json
```

To retrieve the cost for Spot instances with a specified duration using the AWS CLI

Use the `describe-spot-instance-requests` command to retrieve the fixed cost for your Spot instances with a specified duration. The information is in the `actualBlockHourlyPrice` field.

Creating a Spot Instance Request

The process for requesting a Spot instance is similar to the process for launching an On-Demand instance. Note that you can't change the parameters of your Spot request, including the bid price, after you've submitted the request.

If you request multiple Spot instances at one time, Amazon EC2 creates separate Spot instance requests so that you can track the status of each request separately. For more information about tracking Spot requests, see [Spot Bid Status \(p. 255\)](#).

Prerequisites

Before you begin, decide on your bid price, how many Spot instances you'd like, and what instance type to use. To review Spot price trends, see [Spot Instance Pricing History \(p. 229\)](#).

To create a Spot instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. If you are new to Spot instances, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
4. On the **Find instance types** page, do the following:
 - a. For **Request type**, the default is a one-time Spot request created using a Spot fleet. For more information, see [Spot Fleet Requests \(p. 237\)](#). To use Spot blocks instead, select **Reserve for duration**.
 - b. For **Target capacity**, enter the number of units to request. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage.
 - c. [Spot block] For **Reserved duration**, select the number of hours for the job to complete.
 - d. For **AMI**, choose one of the basic Amazon Machine Images (AMI) provided by AWS, or choose **Use custom AMI** to specify your own AMI.
 - e. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).
 - f. [Spot fleet] For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 224\)](#).
 - g. For **Network**, your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 549\)](#).
 - [Existing VPC] Select the VPC.
 - [New VPC] Select **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - [EC2-Classic] Select **EC2-Classic**.
 - h. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot instances. If you prefer specific Availability Zones, do the following:
 - [EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- [EC2-Classical] Select **Select specific zone/subnet**, and then select one or more Availability Zones.
 - i. [Spot fleet] For **Maximum price**, you can use automated bidding or specify a bid price. Your Spot instances are not launched if your bid price is lower than the Spot price for the instance types that you selected.
 - j. Choose **Next**.
5. On the **Configure** page, do the following:
 - a. (Optional) If you need to connect to your instances, specify your key pair using **Key pair name**.
 - b. (Optional) If you need to launch your Spot instances with an IAM role, specify the role using **IAM instance profile**.
 - c. (Optional) If you have any start-up scripts to run, specify them using **User data**.
 - d. For **Security groups**, choose one or more security groups.
 - e. [EC2-VPC] If you need to connect to your instances in a VPC, select **auto-assign at launch** for **Public IP**.
 - f. By default, the request remains in effect until it is fulfilled or you cancel it. To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid to**.
 - g. [Spot fleet] By default, we terminate your Spot instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.
 - h. Choose **Review**.
 6. On the **Review** page, verify the launch configuration. To make changes, choose **Previous**. To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**. When you are ready, choose **Launch**.
 7. On the confirmation page, choose **OK**.

[Spot fleet] The request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

[Spot block] The request type is `block` and the initial state is `open`. When the request is fulfilled, the state is `active` and the status is `fulfilled`.

To create a Spot instance request using the AWS CLI

Use the following [request-spot-instances](#) command to create a one-time request:

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Use the following [request-spot-instances](#) command to create a persistent request:

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "persistent" --launch-specification file://specification.json
```

For example launch specification files, see [Spot Request Example Launch Specifications](#) (p. 235).

Amazon EC2 launches your Spot instance when the Spot price is below your bid. The Spot instance runs until either it is interrupted, or you terminate it yourself. Use the following [describe-spot-instance-requests](#) command to monitor your Spot instance request:

```
aws ec2 describe-spot-instance-requests --spot-instance-request-ids sir-08b93456
```


Finding Running Spot Instances

Amazon EC2 launches a Spot instance when the Spot price is below your bid. A Spot instance runs until either its bid price is no longer higher than the Spot price, or you terminate it yourself. (If your bid price is exactly equal to the Spot price, there is a chance that your Spot instance will remain running, depending on demand.)

To find running Spot instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.

You can see both Spot instance requests and Spot fleet requests. If a Spot instance request has been fulfilled, **Capacity** is the ID of the Spot instance. For a Spot fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot fleet, choose the expand arrow, or select the fleet and then select the **Instances** tab.

3. Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the **Show/Hide** icon, and then select **Lifecycle**. For each instance, **Lifecycle** is either `normal`, `spot`, or `scheduled`.

To find running Spot instances using the AWS CLI

To enumerate your Spot instances, use the `describe-spot-instance-requests` command with the `--query` option as follows:

```
aws ec2 describe-spot-instance-requests --query SpotInstanceRequests[*].  
{ID:InstanceId}
```

The following is example output:

```
[  
  {  
    "ID": "i-1234567890abcdef0"  
  },  
  {  
    "ID": "i-0598c7d356eba48d7"  
  }  
]
```

Alternatively, you can enumerate your Spot instances using the `describe-instances` command with the `--filters` option as follows:

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

Tagging Spot Instance Requests

To help categorize and manage your Spot instance requests, you can tag them with metadata of your choice. You tag your Spot instance requests in the same way that you tag other any other Amazon EC2 resource. For more information, see [Tagging Your Amazon EC2 Resources \(p. 751\)](#).

You can assign a tag to the request after you create it.

The tags that you create for your Spot instance requests only apply to the requests. These tags are not added automatically to the Spot instance that the Spot service launches to fulfill the request. You must add tags to a Spot instance yourself after the Spot instance is launched.

To add a tag to your Spot instance request or Spot instance using the AWS CLI

Use the following [create-tags](#) command to tag your resources:

```
aws ec2 create-tags --resources sir-08b93456 i-1234567890abcdef0 --tags  
Key=purpose,Value=test
```

Canceling a Spot Instance Request

If you no longer want your Spot request, you can cancel it. You can only cancel Spot instance requests that are `open` or `active`. Your Spot request is `open` when your request has not yet been fulfilled and no instances have been launched. Your Spot request is `active` when your request has been fulfilled, and Spot instances have launched as a result. If your Spot request is `active` and has an associated running Spot instance, cancelling the request does not terminate the instance; you must terminate the running Spot instance manually.

If the Spot request is a persistent Spot request, it returns to the `open` state so that a new Spot instance can be launched. To cancel a persistent Spot request and terminate its Spot instances, you must cancel the Spot request first and then terminate the Spot instances. Otherwise, the Spot request can launch a new instance.

To cancel a Spot instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. (Optional) If you are finished with the associated Spot instances, you can terminate them. In the navigation pane, choose **Instances**, select the instance, choose **Actions**, choose **Instance State**, and then choose **Terminate**.

To cancel a Spot instance request using the AWS CLI

Use the following [cancel-spot-instance-requests](#) command to cancel the specified Spot request:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-  
ids sir-08b93456
```

If you are finished with the associated Spot instances, you can terminate them manually using the following [terminate-instances](#) command:

```
aws ec2 terminate-instances --instance-  
ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

Spot Request Example Launch Specifications

The following examples show launch configurations that you can use with the [request-spot-instances](#) command to create a Spot instance request. For more information, see [Creating a Spot Instance Request](#) (p. 232).

1. [Launch Spot instances](#) (p. 235)
2. [Launch Spot instances in the specified Availability Zone](#) (p. 236)
3. [Launch Spot instances in the specified subnet](#) (p. 236)

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. If your account supports EC2-VPC only, Amazon EC2 launches the instances

Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

in the default subnet of the selected Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in the selected Availability Zone.

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Note that you can specify security groups for EC2-Classic either by ID or by name (using the `SecurityGroups` field). You must specify security groups for EC2-VPC by ID.

Example 2: Launch Spot Instances in the Specified Availability Zone

The following example includes an Availability Zone. If your account supports EC2-VPC only, Amazon EC2 launches the instances in the default subnet of the specified Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in the specified Availability Zone.

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Example 3: Launch Spot Instances in the Specified Subnet

The following example includes a subnet. Amazon EC2 launches the instances in the specified subnet. If the VPC is a nondefault VPC, the instance does not receive a public IP address by default.

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

To assign a public IP address to an instance in a nondefault VPC, specify the `AssociatePublicIpAddress` field as shown in the following example. Note that when you specify a network interface, you must include the subnet ID and security group ID using the network interface, rather than using the `SubnetId` and `SecurityGroupIds` fields shown in example 3.

```
{
```

```
"ImageId": "ami-1a2b3c4d",
"KeyName": "my-key-pair",
"InstanceType": "m3.medium",
"NetworkInterfaces": [
  {
    "DeviceIndex": 0,
    "SubnetId": "subnet-1a2b3c4d",
    "Groups": [ "sg-1a2b3c4d" ],
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}
```

Spot Fleet Requests

To use a Spot fleet, you create a Spot fleet request that includes the target capacity, one or more launch specifications for the instances, and the bid price that you are willing to pay. Amazon EC2 attempts to maintain your Spot fleet's target capacity as Spot prices change. For more information, see [How Spot Fleet Works \(p. 224\)](#).

You can create a Spot fleet to submit a one-time `request` for your desired capacity, or require it to `maintain` a target capacity over time. Both types of requests benefit from Spot fleet's allocation strategy.

When you `request` a target capacity, Spot fleet places the required bids but will not attempt to replenish Spot instances if capacity is diminished. If capacity is not available, Spot fleet will not submit bids in alternative Spot pools.

When you want to `maintain` a target capacity, Spot fleet will place the required bids to meet this target capacity and automatically replenish any interrupted instances. By default, Spot fleets are set to `maintain` the requested target capacity.

It is not possible to modify the target capacity of a one-time `request` once it's been submitted. To change the target capacity, cancel the request and submit a new one.

A Spot fleet request remains active until it expires or you cancel it. When you cancel a Spot fleet request, you may specify whether cancelling your Spot fleet request terminates the Spot instances in your Spot fleet.

Each launch specification includes the information that Amazon EC2 needs to launch an instance—such as an AMI, an instance type, a subnet or Availability Zone, and one or more security groups.

Contents

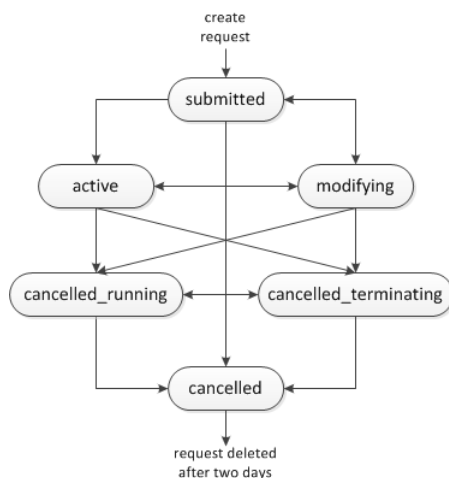
- [Spot Fleet Request States \(p. 238\)](#)
- [Spot Fleet Prerequisites \(p. 238\)](#)
- [Spot Fleet and IAM Users \(p. 239\)](#)
- [Planning a Spot Fleet Request \(p. 240\)](#)
- [Creating a Spot Fleet Request \(p. 240\)](#)
- [Monitoring Your Spot Fleet \(p. 241\)](#)
- [Modifying a Spot Fleet Request \(p. 242\)](#)
- [Cancelling a Spot Fleet Request \(p. 243\)](#)
- [Spot Fleet Example Configurations \(p. 244\)](#)

Spot Fleet Request States

A Spot fleet request can be in one of the following states:

- `submitted`—The Spot fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of Spot instances.
- `active`—The Spot fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot instances. The request remains in this state until it is modified or cancelled.
- `modifying`—The Spot fleet request is being modified. The request remains in this state until the modification is fully processed or the Spot fleet is cancelled. A one-time `request` cannot be modified, and this state does not apply to such Spot requests.
- `cancelled_running`—The Spot fleet is cancelled and will not launch additional Spot instances, but its existing Spot instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- `cancelled_terminating`—The Spot fleet is cancelled and its Spot instances are terminating. The request remains in this state until all instances are terminated.
- `cancelled`—The Spot fleet is cancelled and has no running Spot instances. The Spot fleet request is deleted two days after its instances were terminated.

The following illustration represents the transitions between the request states. Note that if you exceed your Spot fleet limits, the request is cancelled immediately.



Spot Fleet Prerequisites

If you use the AWS Management Console to create a Spot fleet, it creates a role named `aws-ec2-spot-fleet-role` that grants the Spot fleet permission to bid on, launch, and terminate instances on your behalf, and specifies it in your Spot fleet request. If you create a Spot fleet using the AWS CLI or an API, you can use this role if it exists, or manually create your own role for this purpose as follows.

To manually create an IAM role with the `AmazonEC2SpotFleetRole` policy

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create New Role**.
4. On the **Set Role Name** page, type a name for the role and then choose **Next Step**.
5. On the **Select Role Type** page, choose **Select** next to **Amazon EC2 Spot Fleet Role**.

6. On the **Attach Policy** page, select the `AmazonEC2SpotFleetRole` policy, and then choose **Next Step**.
7. On the **Review** page, choose **Create Role**.

Spot Fleet and IAM Users

If IAM users will be creating or managing Spot fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for Spot fleet

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**, and then choose **Create Policy**.
3. On the **Create Policy** page, choose **Select** next to **Create Your Own Policy**.
4. On the **Review Policy** page, enter a policy name and copy the following text into the **Policy Document** section.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole",
        "iam:ListRoles",
        "iam:ListInstanceProfiles"
      ],
      "Resource": "*"
    }
  ]
}
```

The `ec2:*` enables an IAM user to call all Amazon EC2 API actions. To limit the user to specific API actions, specify those actions instead.

The `iam:PassRole` action enables the user to specify the Spot fleet role in a Spot fleet request. The `iam:ListRoles` action enables the user to enumerate existing roles. The `iam:ListInstanceProfiles` action enables the user to enumerate existing instance profiles. The Amazon EC2 console uses `iam:ListRoles` to populate the **IAM role** list and `iam:ListInstanceProfiles` to populate the **IAM instance profile** list. To enable the user to create roles or instance profiles using the console, you must add the following actions: `iam:CreateRole`, `iam:CreateInstanceProfile`, and `iam:AddRoleToInstanceProfile`.

5. Choose **Create Policy**.
6. In the navigation pane, choose **Users**, and then choose the user who will submit the Spot fleet request.
7. On the **Permissions** tab, choose **Attach Policy**.
8. On the **Attach Policy** page, select the policy you created above, and choose **Attach Policy**.

Planning a Spot Fleet Request

Before you create a Spot fleet request, review [Spot Best Practices](#). Use these best practices when you plan your Spot fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine whether you want to create a Spot fleet that submits a one-time `request` for the desired target capacity, or one that will `maintain` a target capacity over time.
- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot fleet request. You can set target capacity in instances or in custom units. For more information, see [Spot Fleet Instance Weighting \(p. 225\)](#).
- Determine your bid price per instance hour. Bidding lower can further reduce costs, while bidding higher can reduce the probability of interruption.
- Determine your bid price per unit, if you are using instance weighting. To calculate the bid price per unit, divide the bid price per instance hour by the number of units (or weight) that this instance represents. (If you are not using instance weighting, the default bid price per unit is the bid price per instance hour.)
- Review the possible options for your Spot fleet request. For more information, see the [request-spot-fleet](#) command in the *AWS Command Line Interface Reference*. For additional examples, see [Spot Fleet Example Configurations \(p. 244\)](#).

Creating a Spot Fleet Request

When you create a Spot fleet request, you must specify information about the Spot instances to launch, such as the instance type and the Spot price.

To create a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. On the **Find instance types** page, do the following:
 - a. For **Request type**, select either **Request** or **Request and Maintain**.
 - b. For **Target capacity**, enter the number of units to request. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage.
 - c. For **AMI**, choose one of the basic Amazon Machine Images (AMI) provided by AWS, or choose **Use custom AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.
 - d. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).
 - e. For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 224\)](#).
 - f. For **Network**, your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 549\)](#).
 - [Existing VPC] Select the VPC.
 - [New VPC] Select **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - [EC2-Classic] Select **EC2-Classic**.
 - g. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot instances. If you prefer specific Availability Zones, do the following:

Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

- [EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - [EC2-Classic] Select **Select specific zone/subnet**, and then select one or more Availability Zones.
- h. For **Maximum price**, you can use automated bidding or specify a bid price. Your Spot instances are not launched if your bid price is lower than the Spot price for the instance types that you selected.
 - i. Choose **Next**.
4. On the **Configure** page, do the following:
 - a. (Optional) If you need to connect to your instances, specify your key pair using **Key pair name**.
 - b. (Optional) If you need to launch your Spot instances with an IAM role, specify the role using **IAM instance profile**.
 - c. (Optional) If you have any start-up scripts to run, specify them using **User data**.
 - d. For **Security groups**, choose one or more security groups.
 - e. [EC2-VPC] If you need to connect to your instances in a VPC, select **auto-assign at launch** for **Public IP**.
 - f. By default, the request remains in effect until it is fulfilled or you cancel it. To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid to**.
 - g. (Optional) By default, we terminate your Spot instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.
 - h. Choose **Review**.
 5. On the **Review** page, verify the launch configuration. To make changes, choose **Previous**. To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**. When you are ready, choose **Launch**.
 6. On the confirmation page, choose **OK**. The request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

To create a Spot fleet request using the AWS CLI

Use the following `request-spot-fleet` command to create a Spot fleet request:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For example configuration files, see [Spot Fleet Example Configurations \(p. 244\)](#).

The following is example output:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Monitoring Your Spot Fleet

The Spot fleet launches Spot instances when the Spot price is below your bid. The Spot instances run until either the bid price is no longer higher than the Spot price, or you terminate them yourself.

To monitor your Spot fleet using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot fleet request. The configuration details are available in the **Description** tab.
4. To list the Spot instances for the Spot fleet, choose the **Instances** tab.
5. To view the history for the Spot fleet, choose the **History** tab.

To monitor your Spot fleet using the AWS CLI

Use the following [describe-spot-fleet-requests](#) command to describe your Spot fleet requests:

```
aws ec2 describe-spot-fleet-requests
```

Use the following [describe-spot-fleet-instances](#) command to describe the Spot instances for the specified Spot fleet:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the following [describe-spot-fleet-request-history](#) command to describe the history for the specified Spot fleet request:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-18T00:00:00Z
```

Modifying a Spot Fleet Request

You can modify an active Spot fleet request to complete the following tasks:

- Increase the target capacity
- Decrease the target capacity

Note

It is not possible to modify a one-time Spot fleet request.

When you increase the target capacity, the Spot fleet launches the additional Spot instances according to the allocation strategy for its Spot fleet request. If the allocation strategy is `lowestPrice`, the Spot fleet launches the instances from the lowest-priced Spot instance pool in the Spot fleet request. If the allocation strategy is `diversified`, the Spot fleet distributes the instances across the pools in the Spot fleet request.

When you decrease the target capacity, the Spot fleet cancels any open bids that exceed the new target capacity. You can request that the Spot fleet terminate Spot instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `lowestPrice`, the Spot fleet terminates the instances with the highest price per unit. If the allocation strategy is `diversified`, the Spot fleet terminates instances across the pools. Alternatively, you can request that the Spot fleet keep the fleet at its current size, but not replace any Spot instances that are interrupted or that you terminate manually.

To modify a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot fleet request.
3. Choose **Actions**, and then choose **Modify target capacity**.
4. In **Modify target capacity**, do the following:

- a. Enter the new target capacity.
- b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, deselect **Terminate instances**.
- c. Choose **Submit**.

To modify a Spot fleet request using the AWS CLI

Use the following [modify-spot-fleet-request](#) command to update the target capacity of the specified Spot fleet request:

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 20
```

You can modify the previous command as follows to decrease the target capacity of the specified Spot fleet without terminating any Spot instances as a result:

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 10 --excess-capacity-termination-policy NoTermination
```

Canceling a Spot Fleet Request

When you are finished using your Spot fleet, you can cancel the Spot fleet request. This cancels all Spot requests associated with the Spot fleet, so that no new Spot instances are launched for your Spot fleet. You must specify whether the Spot fleet should terminate its Spot instances. If you terminate the instances, the Spot fleet request enters the `cancelled_terminating` state. Otherwise, the Spot fleet request enters the `cancelled_running` state and the instances continue to run until they are interrupted or you terminate them manually.

To cancel a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot fleet request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. In **Cancel spot request**, verify that you want to cancel the Spot fleet. To keep the fleet at its current size, deselect **Terminate instances**. When you are ready, choose **Confirm**.

To cancel a Spot fleet request using the AWS CLI

Use the following [cancel-spot-fleet-requests](#) command to cancel the specified Spot fleet request and terminate the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

The following is example output:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_terminating",
      "PreviousSpotFleetRequestState": "active"
    }
  ]
}
```

```
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

You can modify the previous command as follows to cancel the specified Spot fleet request without terminating the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

The following is example output:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_running",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

Spot Fleet Example Configurations

The following examples show launch configurations that you can use with the `request-spot-fleet` command to create a Spot fleet request. For more information, see [Creating a Spot Fleet Request](#) (p. 240).

1. [Launch Spot instances using the lowest-priced Availability Zone or subnet in the region](#) (p. 244)
2. [Launch Spot instances using the lowest-priced Availability Zone or subnet in a specified list](#) (p. 245)
3. [Launch Spot instances using the lowest-priced instance type in a specified list](#) (p. 246)
4. [Override the Spot price for the request](#) (p. 248)
5. [Launch a Spot fleet using the diversified allocation strategy](#) (p. 249)
6. [Launch a Spot fleet using instance weighting](#) (p. 250)

Example 1: Launch Spot Instances Using the Lowest-priced Availability Zone or Subnet in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. If your account supports EC2-VPC only, the Spot fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. If your account supports EC2-Classic, the Spot fleet launches the instances in EC2-Classic in the lowest-priced Availability Zone. Note that the price you pay will not exceed the specified Spot price for the request.

```
{  
  "SpotPrice": "0.07",  
  "TargetCapacity": 20,  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "KeyName": "my-key-pair",  
    }  
  ]  
}
```

```
    "SecurityGroups": [
      {
        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "InstanceType": "m3.medium",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
  }
]
```

Example 2: Launch Spot Instances Using the Lowest-priced Availability Zone or Subnet in a Specified List

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

If your account supports EC2-VPC only, the Spot fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified. If your account supports EC2-Classic, the Spot fleet launches the instances in the lowest-priced Availability Zone that you specified.

```
{
  "SpotPrice": "0.07",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "Placement": {
        "AvailabilityZone": "us-west-2a, us-west-2b"
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Subnets

You can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

Note that you can't specify different subnets from the same Availability Zone in a Spot fleet request.

```
{
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

```
"SpotPrice": "0.07",
"TargetCapacity": 20,
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
"LaunchSpecifications": [
  {
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroups": [
      {
        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "InstanceType": "m3.medium",
    "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
  }
]
}
```

If the instances are launched in a default VPC, they receive a public IP address by default. If the instances are launched in a nondefault VPC, they do not receive a public IP address by default. Use a network interface in the launch specification to assign a public IP address to instances launched in a nondefault VPC. Note that when you specify a network interface, you must include the subnet ID and security group ID using the network interface.

```
...
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
  }
}
...
```

Example 3: Launch Spot Instances Using the Lowest-priced Instance Type in a Specified List

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
  "SpotPrice": "2.80",
  "TargetCapacity": 20,
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Spot Instances**

```
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
"LaunchSpecifications": [
  {
    "ImageId": "ami-1a2b3c4d",
    "SecurityGroups": [
      {
        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "InstanceType": "cc2.8xlarge",
    "Placement": {
      "AvailabilityZone": "us-west-2b"
    }
  },
  {
    "ImageId": "ami-1a2b3c4d",
    "SecurityGroups": [
      {
        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "InstanceType": "r3.8xlarge",
    "Placement": {
      "AvailabilityZone": "us-west-2b"
    }
  }
]
}
```

Subnet

```
{
  "SpotPrice": "2.80",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "cc2.8xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "r3.8xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    }
  ]
}
```

Example 4. Override the Spot Price for the Request

The ability to specify Spot prices for individual launch specifications provides you with additional control over the bidding process. The following examples override the Spot price for the request (0.070) with individual Spot prices for two of the three launch specifications. Note that the Spot price for the request is used for any launch specification that does not specify an individual Spot price. The Spot fleet launches the instances using the instance type with the lowest price.

Availability Zone

```
{
  "SpotPrice": "1.68",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
      "SpotPrice": "0.04"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.4xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
      "SpotPrice": "0.06"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.8xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    }
  ]
}
```

Subnet

```
{
  "SpotPrice": "1.68",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d",
      "SpotPrice": "0.04"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.4xlarge",
      "SubnetId": "subnet-1a2b3c4d",

```

```
    "SpotPrice": "0.06"  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "c3.8xlarge",  
    "SubnetId": "subnet-1a2b3c4d"  
  }  
]  
}
```

Example 5: Launch a Spot Fleet Using the Diversified Allocation Strategy

The following example uses the `diversified` allocation strategy. The launch specifications have different instance types but the same AMI and Availability Zone or subnet. The Spot fleet distributes the 30 instances across the 3 launch specifications, such that there are 10 instances of each type. For more information, see [Spot Fleet Allocation Strategy \(p. 224\)](#).

Availability Zone

```
{  
  "SpotPrice": "0.70",  
  "TargetCapacity": 30,  
  "AllocationStrategy": "diversified",  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c4.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "m3.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "r3.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    }  
  ]  
}
```

Subnet

```
{  
  "SpotPrice": "0.70",  
  "TargetCapacity": 30,  
  "AllocationStrategy": "diversified",  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c4.2xlarge",  
      "SubnetId": "subnet-1a2b3c4d"  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "m3.2xlarge",  
      "SubnetId": "subnet-1a2b3c4d"  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "r3.2xlarge",  
      "SubnetId": "subnet-1a2b3c4d"  
    }  
  ]  
}
```


Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

```
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "c4.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  },
  {
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "m3.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  },
  {
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "r3.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  }
]
```

Example 6: Launch a Spot Fleet Using Instance Weighting

The following examples use instance weighting, which means that the bid price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot fleet selects the instance type with the lowest price per unit hour. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` bid is successful, Spot provisions 4 of these instances. (Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.)

If the `c3.xlarge` bid is successful, Spot provisions 7 of these instances. (Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.)

For more information, see [Spot Fleet Instance Weighting](#) (p. 225).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
      "WeightedCapacity": 6
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
      "WeightedCapacity": 3
    }
  ]
}
```

Subnet

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d",
      "WeightedCapacity": 6
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.xlarge",
      "SubnetId": "subnet-1a2b3c4d",
      "WeightedCapacity": 3
    }
  ]
}
```

Priority

You can also use instance weighting to give priority to an Availability Zone or subnet. For example, the following launch specifications are nearly identical, except that they specify different subnets and weights. The Spot fleet finds the specification with the highest value for `WeightedCapacity`, and attempts to provision the request in the least expensive Spot instance pool in that subnet. (Note that the second launch specification does not include a weight, so it defaults to 1.)

```
{
  "SpotPrice": "0.42",
  "TargetCapacity": 40,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "SubnetId": "subnet-bb3337d"
    }
  ]
}
```

CloudWatch Metrics for Spot Fleet

Amazon EC2 provides Amazon CloudWatch metrics that you can use to monitor your Spot fleet.

Important

To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see [Enable or Disable Detailed Monitoring for Your Instances](#) (p. 419).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitoring Your Instances Using CloudWatch \(p. 419\)](#).

Spot Fleet Metrics

The `AWS/EC2Spot` namespace includes the following metrics, plus the CloudWatch metrics for the Spot instances in your fleet. For more information, see [Instance Metrics \(p. 421\)](#).

The `AWS/EC2Spot` namespace includes the following metrics.

Metric	Description
<code>AvailableInstancePoolsCount</code>	The Spot Instance pools specified in the Spot Fleet request. Units: Count
<code>BidsSubmittedForCapacity</code>	The capacity for which Amazon EC2 has submitted bids. Units: Count
<code>EligibleInstancePoolCount</code>	The Spot Instance pools specified in the Spot Fleet request where Amazon EC2 can fulfill bids. Amazon EC2 will not fulfill bids in pools where your bid price is less than the Spot price or the Spot price is greater than the price for On-Demand instances. Units: Count
<code>FulfilledCapacity</code>	The capacity that Amazon EC2 has fulfilled. Units: Count
<code>MaxPercentCapacityAllocation</code>	The maximum value of <code>PercentCapacityAllocation</code> across all Spot Instance pools specified in the Spot Fleet request. Units: Percent
<code>PendingCapacity</code>	The difference between <code>TargetCapacity</code> and <code>FulfilledCapacity</code> . Units: Count
<code>PercentCapacityAllocation</code>	The capacity allocated for the Spot Instance pool for the specified dimensions. To get the maximum value recorded across all Spot Instance pools, use <code>MaxPercentCapacityAllocation</code> . Units: Percent
<code>TargetCapacity</code>	The target capacity of the Spot Fleet request. Units: Count
<code>TerminatingCapacity</code>	The capacity that is being terminated due to Spot Instance interruptions. Units: Count

If the unit of measure for a metric is `Count`, the most useful statistic is `Average`.

Spot Fleet Dimensions

To filter the data for your Spot fleet, you can use the following dimensions.

Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

Dimensions	Description
AvailabilityZone	Filter the data by Availability Zone.
FleetRequestId	Filter the data by Spot Fleet request.
InstanceType	Filter the data by instance type.

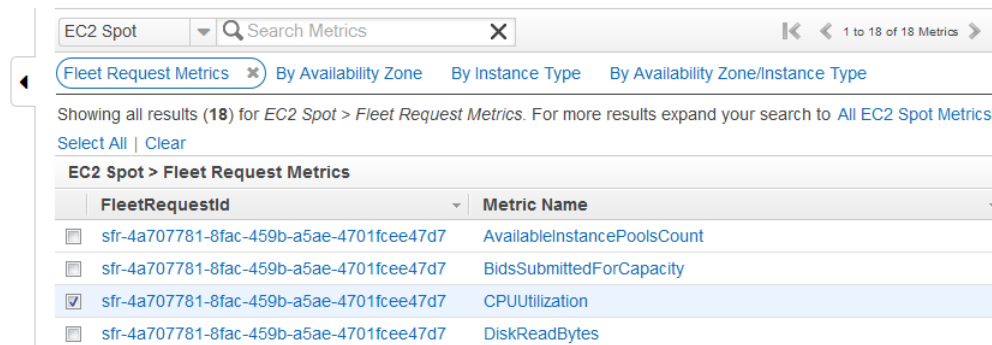
View the CloudWatch Metrics for Your Spot Fleet

You can view the CloudWatch metrics for your Spot fleet using the Amazon CloudWatch console. These metrics are displayed as monitoring graphs. These graphs show data points if the Spot fleet is active.

Metrics are grouped first by namespace, and then by the various combinations of dimensions within each namespace. For example, you can view all Spot fleet metrics, or Spot fleet metrics groups by Spot fleet request ID, instance type, or Availability Zone.

To view Spot fleet metrics

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, under **Metrics**, choose the **EC2 Spot** namespace.
3. (Optional) To filter the metrics by dimension, select one of the following:
 - **Fleet Request Metrics** — Group by Spot fleet request
 - **By Availability Zone** — Group by Spot fleet request and Availability Zone
 - **By Instance Type** — Group by Spot fleet request and instance type
 - **By Availability Zone/Instance Type** — Group by Spot fleet request, Availability Zone, and instance type
4. To view the data for a metric, select the check box next to the metric.



Automatic Scaling for Spot Fleet

Automatic scaling is the ability to increase or decrease the target capacity of your Spot fleet automatically based on demand. A Spot fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies. We recommend that you create two policies, one for scaling out and one for scaling in.

A *scaling policy* uses CloudWatch alarms to trigger the scaling process. For example, if you want to scale out when CPU utilization reaches a certain level, create an alarm using the `CPUUtilization` metric provided by Amazon EC2.

When you create a scaling policy, you must specify one of the following scaling adjustment types:

- **Add** — Increase the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Remove** — Decrease the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Set to** — Set the target capacity of the fleet to the specified number of capacity units.

You can also configure the cooldown period for a scaling policy. After Auto Scaling starts a scaling activity, it waits for the cooldown period to complete before resuming scaling activities. This ensures that Auto Scaling doesn't launch or terminate additional Spot instances before the previous launch or terminate activity takes effect.

Limits

- The Spot fleet request must have a request type of `maintain`. Automatic scaling is not supported for one-time requests or Spot blocks.

Prerequisites

- Consider which CloudWatch metrics are important to your application. You can create CloudWatch alarms based on metrics provided by AWS or your own custom metrics.
- For the AWS metrics that you will use in your scaling policies, enable CloudWatch metrics collection if the service that provides the metrics does not enable it by default.
- If you use the AWS Management Console to enable automatic scaling for your Spot fleet, it creates a role named `aws-ec2-spot-fleet-autoscale-role` that grants Auto Scaling permission to describe the alarms for your policies, monitor the current capacity of the fleet, and modify the capacity of the fleet. If you configure automatic scaling using the AWS CLI or an API, you can use this role if it exists, or manually create your own role for this purpose as follows.
 1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
 2. In the navigation pane, choose **Roles**.
 3. Choose **Create New Role**.
 4. On the **Set Role Name** page, type a name for the role and then choose **Next Step**.
 5. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.
 6. On the **Attach Policy** page, select the `AmazonEC2SpotFleetAutoscaleRole` policy and then choose **Next Step**.
 7. On the **Review** page, choose **Create Role**.
 8. Select the role that you just created.
 9. On the **Trust Relationships** tab, choose **Edit Trust Relationship**.
 10. Change `ec2.amazonaws.com` to `application-autoscaling.amazonaws.com` and then choose **Update Trust Policy**.

To create a CloudWatch alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Choose **Create Alarm**.
4. For **CloudWatch Metrics by Category**, choose a category. For example, choose **EC2 Spot Metrics, Fleet Request Metrics**.
5. Select a metric, and then choose **Next**.
6. For **Alarm Threshold**, type a name and description for the alarm, and set the threshold value and number of time periods for the alarm.

- (Optional) To receive notification of a scaling event, for **Actions**, choose **New list** and type your email address. Otherwise, you can delete the notification now and add one later if needed.
- Choose **Create Alarm**.

To configure automatic scaling for your Spot fleet using the console

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot fleet request, and then choose the **Auto Scaling** tab.
- If automatic scaling is not configured, choose **Configure**.
- Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling will not scale your fleet below the minimum capacity or above the maximum capacity.
- Initially, **Scaling policies** contains policies named ScaleUp and ScaleDown. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy** to add a policy.
- To define a policy, do the following:
 - For **Policy name**, type a name for the policy.
 - For **Policy trigger**, select an existing alarm or choose **Create new alarm** to open the Amazon CloudWatch console and create an alarm.
 - For **Modify capacity**, select a scaling adjustment type, select a number, and select a unit.
 - (Optional) To perform step scaling, choose **Define steps**. By default, an add policy has a lower bound of -infinity and an upper bound of the alarm threshold. By default, a remove policy has a lower bound of the alarm threshold and an upper bound of +infinity. To add another step, choose **Add step**.
 - (Optional) To modify the default value for the cooldown period, select a number from **Cooldown period**.
- Choose **Save**.

To configure automatic scaling for your Spot fleet using the AWS CLI

- Register the Spot fleet request as a scalable target using the [register-scalable-target](#) command.
- Create a scaling policy using the [put-scaling-policy](#) command.
- Create an alarm that will trigger the scaling policy using the [put-metric-alarm](#) command.

Spot Bid Status

To help you track your Spot instance requests, plan your use of Spot instances, and bid strategically, Amazon EC2 provides a *bid status*. For example, a bid status can tell you the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

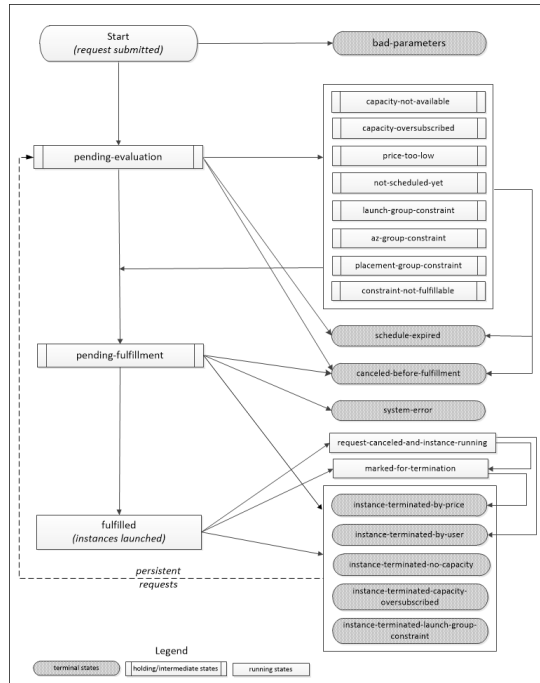
At each step of the process—also called the Spot request *life cycle*, specific events determine successive request states.

Contents

- [Life Cycle of a Spot Request \(p. 256\)](#)
- [Getting Bid Status Information \(p. 258\)](#)
- [Spot Bid Status Codes \(p. 259\)](#)

Life Cycle of a Spot Request

The following diagram shows you the paths that your Spot request can follow throughout its life cycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot instance.



Pending evaluation

As soon as you make a Spot instance request, it goes into the `pending-evaluation` state unless one or more request parameters is not valid (`bad-parameters`).

Status Code	Request State	Instance State
<code>pending-evaluation</code>	<code>open</code>	<code>n/a</code>
<code>bad-parameters</code>	<code>closed</code>	<code>n/a</code>

Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a bid price below the current Spot price, your request stays in a holding state until the Spot price goes below your bid price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

Status Code	Request State	Instance State
<code>capacity-not-available</code>	<code>open</code>	<code>n/a</code>
<code>capacity-oversubscribed</code>	<code>open</code>	<code>n/a</code>
<code>price-too-low</code>	<code>open</code>	<code>n/a</code>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Spot Instances**

Status Code	Request State	Instance State
not-scheduled-yet	open	n/a
launch-group-constraint	open	n/a
az-group-constraint	open	n/a
placement-group-constraint	open	n/a
constraint-not-fulfillable	open	n/a

Pending evaluation/fulfillment-terminal

Your Spot instance request can go to a terminal state if you create a request that is valid only during a specific time period and this time period expires before your request reaches the pending fulfillment phase, you cancel the request, or a system error occurs.

Status Code	Request State	Instance State
schedule-expired	closed	n/a
cancel-before-fulfillment*	cancelled	n/a
bad-parameters	failed	n/a
system-error	closed	n/a

* If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met and your bid price is equal to or higher than the current Spot price, your Spot request goes into the `pending-fulfillment` state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was cancelled by the user before a Spot instance was launched, or because an unexpected system error occurred.

Status Code	Request State	Instance State
pending-fulfillment	open	n/a

Fulfilled

When all the specifications for your Spot instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot instances, which can take a few minutes.

Status Code	Request State	Instance State
fulfilled	active	pending → running

Fulfilled-terminal

Amazon Elastic Compute Cloud User Guide for Linux Instances Spot Instances

Your Spot instances continue to run as long as your bid price is at or above the Spot price, there is spare Spot capacity for your instance type, and you don't terminate the instance. If a change in Spot price or available capacity requires Amazon EC2 to terminate your Spot instances, the Spot request goes into a terminal state. For example, if your bid equals the Spot price but Spot instances are oversubscribed at that price, the status code is `instance-terminated-capacity-oversubscribed`. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot instances.

Status Code	Request State	Instance State
<code>request-canceled-and-instance-running</code>	<code>cancelled</code>	<code>running</code>
<code>marked-for-termination</code>	<code>closed</code>	<code>running</code>
<code>instance-terminated-by-price</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>
<code>instance-terminated-by-user</code>	<code>closed or cancelled *</code>	<code>terminated</code>
<code>instance-terminated-no-capacity</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>
<code>instance-terminated-capacity-oversubscribed</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>
<code>instance-terminated-launch-group-constraint</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>

* The request state is `closed` if you terminate the instance but do not cancel the bid. The request state is `cancelled` if you terminate the instance and cancel the bid. Note that even if you terminate a Spot instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot instance was terminated. In this case, the request state can either be `closed` or `cancelled`.

Persistent requests

When your Spot instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the `pending-evaluation` state and then Amazon EC2 can launch a new Spot instance when the constraints are met.

Getting Bid Status Information

You can get bid status information using the AWS Management Console or a command line tool.

To get bid status information using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Check the value of **Status** in the **Description** tab.

To get bid status information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-spot-instance-requests` (AWS CLI)

- [Get-EC2SpotInstanceRequest](#) (AWS Tools for Windows PowerShell)

Spot Bid Status Codes

Spot bid status information is composed of a bid status code, the update time, and a status message. Together, they help you determine the disposition of your Spot request.

The following list describes the Spot bid status codes:

`az-group-constraint`

Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

`bad-parameters`

One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The bid status message indicates which parameter is not valid.

`cancelled-before-fulfillment`

The user cancelled the Spot request before it was fulfilled.

`capacity-not-available`

There is not enough capacity available for the instances that you requested.

`capacity-oversubscribed`

The number of Spot requests with bid prices equal to or higher than your bid price exceeds the available capacity in this Spot instance pool.

`constraint-not-fulfillable`

The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The bid status message indicates which constraint is not valid.

`fulfilled`

The Spot request is `active`, and Amazon EC2 is launching your Spot instances.

`instance-terminated-by-price`

The Spot price rose above your bid price. If your request is a persistent bid, the process restarts, so your bid is pending evaluation.

`instance-terminated-by-user` or `spot-instance-terminated-by-user`

You terminated a Spot instance that had been fulfilled, so the bid state is `closed` (unless it's a persistent bid) and the instance state is `terminated`.

`instance-terminated-capacity-oversubscribed`

Your instance is terminated because the number of Spot requests with bid prices equal to or higher than your bid price exceeded the available capacity in this Spot instance pool. (Note that the Spot price might not have changed.) The Spot service randomly selects instances to be terminated.

`instance-terminated-launch-group-constraint`

One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

`instance-terminated-no-capacity`

There is no longer enough Spot capacity available for the instance.

`launch-group-constraint`

Amazon EC2 cannot launch all the instances that you requested at the same time. All instances in a launch group are started and terminated together.

`marked-for-termination`

The Spot instance is marked for termination.

`not-scheduled-yet`

The Spot request will not be evaluated until the scheduled date.

`pending-evaluation`

After you make a Spot instance request, it goes into the `pending-evaluation` state while the system evaluates the parameters of your request.

`pending-fulfillment`

Amazon EC2 is trying to provision your Spot instances.

`placement-group-constraint`

The Spot request can't be fulfilled yet because a Spot instance can't be added to the placement group at this time.

`price-too-low`

The bid request can't be fulfilled yet because the bid price is below the Spot price. In this case, no instance is launched and your bid remains `open`.

`request-cancelled-and-instance-running`

You canceled the Spot request while the Spot instances are still running. The request is cancelled, but the instances remain `running`.

`schedule-expired`

The Spot request expired because it was not fulfilled before the specified date.

`system-error`

There was an unexpected system error. If this is a recurring issue, please contact customer support for assistance.

Spot Instance Interruptions

Demand for Spot instances can vary significantly from moment to moment, and the availability of Spot instances can also vary significantly depending on how many unused EC2 instances are available. In addition, no matter how high you bid, it is still possible that your Spot instance will be interrupted. Therefore, you must ensure that your application is prepared for a Spot instance interruption. We strongly recommend that you do not use Spot instances for applications that can't be interrupted.

The following are the possible reasons that Amazon EC2 will terminate your Spot instances:

- **Price**—The Spot price is greater than your bid price.
- **Capacity**—If there are not enough unused EC2 instances to meet the demand for Spot instances, Amazon EC2 terminates Spot instances, starting with those instances with the lowest bid prices. If there are several Spot instances with the same bid price, the order in which the instances are terminated is determined at random.
- **Constraints**—If your request includes a constraint such as a launch group or an Availability Zone group, these Spot instances are terminated as a group when the constraint can no longer be met.

Preparing for Interruptions

Here are some best practices to follow when you use Spot instances:

- Choose a reasonable bid price. Your bid price should be high enough to make it likely that your request will be fulfilled, but not higher than you are willing to pay. This is important because if the supply is low for an extended period of time, the Spot price can remain high during that period because it is based on the highest bid prices. We strongly recommend against bidding above the price for On-Demand instances.
- Ensure that your instance is ready to go as soon as the request is fulfilled by using an Amazon Machine Image (AMI) that contains the required software configuration. You can also use user data to run commands at start-up.
- Store important data regularly in a place that won't be affected when the Spot instance terminates. For example, you can use Amazon S3, Amazon EBS, or DynamoDB.
- Divide the work into small tasks (using a Grid, Hadoop, or queue-based architecture) or use checkpoints so that you can save your work frequently.
- Use Spot instance termination notices to monitor the status of your Spot instances.

- Test your application to ensure that it handles an unexpected instance termination gracefully. You can do so by running the application using an On-Demand instance and then terminating the On-Demand instance yourself.

Spot Instance Termination Notices

The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of *Spot instance termination notices*, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance.

This warning is made available to the applications on your Spot instance using an item in the instance metadata. For example, you can check for this warning in the instance metadata periodically (we recommend every 5 seconds) using the following query:

```
$ if curl -s http://169.254.169.254/latest/meta-data/spot/termination-time |  
  grep -q .*T.*Z; then echo terminated; fi
```

For information about other ways to retrieve instance metadata, see [Retrieving Instance Metadata](#) (p. 338).

If your Spot instance is marked for termination by Amazon EC2, the `termination-time` item is present and it specifies the approximate time in UTC when the instance will receive the shutdown signal. For example:

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot instance yourself, the `termination-time` item is either not present (so you receive an HTTP 404 error) or contains a value that is not a time value.

Note that while we make every effort to provide this warning the moment that your Spot instance is marked for termination by Amazon EC2, it is possible that your Spot instance will be terminated before Amazon EC2 can make the warning available. Therefore, you must ensure that your application is prepared to handle an unexpected Spot instance interruption even if you are checking for Spot instance termination notices.

If Amazon EC2 fails to terminate the instance, the Spot bid status is set to `fulfilled`. Note that `termination-time` remains in the instance metadata with the original approximate time, which is now in the past.

Spot Instance Data Feed

To help you understand the charges for your Spot instances, Amazon EC2 provides a data feed that describes your Spot instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are very large (for example, when file contents for the hour exceed 50 MB before compression).

Note

If you don't have a Spot instance running during a certain hour, you won't receive a data feed file for that hour.

Contents

- [Data Feed File Name and Format](#) (p. 262)
- [Amazon S3 Bucket Requirements](#) (p. 262)
- [Subscribing to Your Spot instance Data Feed](#) (p. 263)
- [Deleting Your Spot Instance Data Feed](#) (p. 263)

Data Feed File Name and Format

The Spot instance data feed file name uses the following format (with the date and hour in UTC):

```
bucket-name.s3.amazonaws.com/{optional prefix}/aws-account-id.YYYY-MM-DD-  
HH.n.unique-id.gz
```

For example, if your bucket name is `myawsbucket` and your prefix is `myprefix`, your file names are similar to the following:

```
myawsbucket.s3.amazonaws.com/myprefix/  
111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

The Spot instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

Field	Description
Timestamp	The timestamp used to determine the price charged for this instance hour.
UsageType	The type of usage and instance type being charged for. For <code>m1.small</code> Spot instances, this field is set to <code>SpotUsage</code> . For all other instance types, this field is set to <code>SpotUsage:{instance-type}</code> . For example, <code>SpotUsage:c1.medium</code> .
Operation	The product being charged for. For Linux Spot instances, this field is set to <code>RunInstances</code> . For Windows Spot instances, this field is set to <code>RunInstances:0002</code> . Spot usage is grouped according to Availability Zone.
InstanceID	The ID of the Spot instance that generated this instance hour.
MyBidID	The ID for the Spot instance request that generated this instance hour.
MyMaxPrice	The maximum price specified for this Spot instance request.
MarketPrice	The Spot price at the time specified in the <code>Timestamp</code> field.
Charge	The price charged for this instance hour.
Version	The version included in the data feed file name for this record.

Amazon S3 Bucket Requirements

When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files. Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must use a bucket from the US East (N. Virginia) (`us-east-1`) region.
- You must have `FULL_CONTROL` permission to the bucket.

If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission.

- When you create your data feed subscription, Amazon S3 updates the ACL of the specified bucket to allow the AWS data feed account read and write permissions.
- Removing the permissions for the data feed account does not disable the data feed. If you remove those permissions but don't disable the data feed, we restore those permissions the next time that the data feed account needs to write to the bucket.
- Each data feed file has its own ACL (separate from the ACL for the bucket). The bucket owner has `FULL_CONTROL` permission to the data files. The data feed account has read and write permissions.
- If you delete your data feed subscription, Amazon EC2 doesn't remove the read and write permissions for the data feed account on either the bucket or the data files. You must remove these permissions yourself.

Subscribing to Your Spot instance Data Feed

To subscribe to your data feed, use the following [create-spot-datafeed-subscription](#) command:

```
$ aws ec2 create-spot-datafeed-subscription --bucket myawsbucket [--  
prefix myprefix]
```

The following is example output:

```
{  
  "SpotDatafeedSubscription": {  
    "OwnerId": "111122223333",  
    "Prefix": "myprefix",  
    "Bucket": "myawsbucket",  
    "State": "Active"  
  }  
}
```

Deleting Your Spot Instance Data Feed

To delete your data feed, use the following [delete-spot-datafeed-subscription](#) command:

```
$ aws ec2 delete-spot-datafeed-subscription
```

Spot Instance Limits

Spot instance requests are subject to the following limits:

Limits

- [Unsupported Instance Types](#) (p. 263)
- [Spot Request Limits](#) (p. 264)
- [Spot Bid Price Limit](#) (p. 264)
- [Spot Fleet Limits](#) (p. 264)
- [Amazon EBS Encryption Unsupported](#) (p. 264)

Unsupported Instance Types

The following instance types are not supported for Spot:

- T2

- HS1

Some Spot instance types aren't available in every region. To view the supported instance types for a region, go to [Spot Instance Pricing](#) and select the region.

Spot Request Limits

By default, there is an account limit of 20 Spot instances per region. If you terminate your Spot instance but do not cancel the request, the request counts against this limit until Amazon EC2 detects the termination and closes the request.

Spot instance limits are dynamic. When your account is new, your limit might be lower than 20 to start, but increase over time. In addition, your account might have limits on specific Spot instance types. If you submit a Spot instance request and you receive the error `Max spot instance count exceeded`, you can go to [AWS Support Center](#) and submit a limit increase request form. For **Use Case Description**, indicate that you need an increase in your limits for Spot instance requests.

Spot Bid Price Limit

The bid price limit for Spot instances is ten times on On-Demand price. This limit is designed to protect you from incurring unexpected charges.

Spot Fleet Limits

The usual Amazon EC2 limits apply to instances launched by a Spot fleet, such as Spot bid price limits, instance limits, and volume limits. In addition, the following limits apply:

- The number of active Spot fleets per region: 1,000
- The number of launch specifications per fleet: 50
- The size of the user data in a launch specification: 16 KB
- The target capacity per Spot fleet: 3,000
- The target capacity across all Spot fleets in a region: 5,000
- A Spot fleet request can't span regions.
- A Spot fleet request can't span different subnets from the same Availability Zone.

Amazon EBS Encryption Unsupported

You can specify encrypted EBS volumes in the launch specification for your Spot instances, but these volumes are not encrypted.

Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, Linux Enterprise Server, and so on.

Contents

- [Differences between Dedicated Hosts and Dedicated Instances \(p. 265\)](#)
- [Pricing and Billing \(p. 265\)](#)
- [Dedicated Hosts Limitations and Restrictions \(p. 266\)](#)
- [Dedicated Host Configurations \(p. 267\)](#)
- [Using Dedicated Hosts \(p. 267\)](#)
- [Monitoring Dedicated Hosts \(p. 274\)](#)

Differences between Dedicated Hosts and Dedicated Instances

Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server.

When you use Dedicated Hosts, you have control over instance placement on the host using the Host Affinity and Instance Auto-placement settings. With Dedicated Instances, you don't have control over which host your instance launches and runs on. If your organization wants to use AWS, but has an existing software license with hardware compliance requirements, this allows visibility into the host's hardware so you can meet those requirements.

For more information about the differences between Dedicated Hosts and Dedicated Instances, see [Amazon EC2 Dedicated Hosts](#).

For more information about working with Dedicated Hosts and Dedicated Instances, see [Modifying Instance Tenancies](#) (p. 271).

Pricing and Billing

On-Demand Dedicated Hosts

On-Demand billing is automatically activated when you allocate a Dedicated Host to your account.

You are billed an hourly On-Demand rate. Rates vary based on the instance type that the Dedicated Host supports and the region in which the Dedicated Host is running. The instance type size or the number of instances that are running on the Dedicated Host do not have an impact on the cost of the host.

To terminate On-Demand billing, you must first stop instances running on the Dedicated Host and then release it. For more information, see [Managing and Releasing Dedicated Hosts](#) (p. 271).

Dedicated Host Reservations

Dedicated Host Reservations provide a billing discount compared to running On-Demand Dedicated Hosts. Reservations are available in three payment options:

- **No Upfront**—No Upfront Reservations provide you with a discount on your Dedicated Host usage over a term and do not require an upfront payment. Available for a one-year term only.
- **Partial Upfront**—A portion of the reservation must be paid upfront and the remaining hours in the term are billed at a discounted rate. Available in one-year and three-year terms.
- **All Upfront**—Provides the lowest effective price. Available in one-year and three-year terms and covers the entire cost of the term upfront, with no additional charges going forward.

You must have active Dedicated Hosts in your account before you can purchase reservations. Each reservation covers a single, specific Dedicated Host in your account. Reservations are applied to the instance family on the host, not the instance size. If you have three Dedicated Hosts with different instance sizes (`m4.xlarge`, `m4.medium`, and `m4.large`) you can associate a single `m4` reservation with all those Dedicated Hosts. The instance family and region of the reservation must match that of the Dedicated Hosts you want to associate it with.

Note

When a reservation is associated with a Dedicated Host, the Dedicated Host can't be released until the reservation's term is over.

Purchasing Dedicated Host Reservations

You can purchase Dedicated Host Reservations using the console or the API.

To purchase Dedicated Host Reservations using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page choose **Dedicated Host Reservations**.
3. Choose **Purchase Dedicated Host Reservation**.
4. On the **Purchase Dedicated Host Reservation** screen, you can search for offerings using the default settings or you can specify a configuration for the offering.
 - **Host instance family**—The options listed correspond with the Dedicated Hosts in your account that are not assigned to a reservation.
 - **Availability Zone**—The Availability Zone of the Dedicated Hosts in your account that aren't assigned to a reservation.
 - **Payment Option**—The payment option for the offering.
 - **Term**—The term of the reservation. Can be one or three years.
5. Choose **Find offering**.
6. Select an offering.
7. Choose the Dedicated Hosts to associate with the Dedicated Host Reservation.
8. Choose **Review**.
9. Review your order and choose **Purchase** to complete the transaction.

Viewing Dedicated Host Reservations

You can view information about the Dedicated Hosts associated with your reservation, the term of the reservation, the payment option selected, and the start and end dates of the reservation.

View details of Dedicated Host Reservations

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the Dedicated Hosts page, choose **Dedicated Host Reservations**.
3. Choose the reservation from the list provided.
4. Select **Details** for information about the reservation.
5. Select **Hosts** for information about the Dedicated Hosts the reservation is associated with.

Dedicated Hosts Limitations and Restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions.

- Only BYOL RHEL, SUSE Linux, or Windows AMIs offered by AWS or on the AWS Marketplace can be used with Dedicated Hosts.
- Up to two On-Demand Dedicated Hosts per instance family, per region can be allocated. It is possible to request a limit increase: [Request to Raise Allocation Limit on Amazon EC2 Dedicated Hosts](#).
- The instances that run on a Dedicated Host can only be launched in a VPC.
- Host limits are independent from instance limits. Instances that you are running on Dedicated Hosts do not count towards your instance limits.
- Auto Scaling groups are not supported.
- Amazon RDS instances are not supported.

- The AWS Free Usage tier is not available for Dedicated Hosts.
- Instance placement control refers to managing instance launches onto Dedicated Hosts. Placement groups are not supported for Dedicated Hosts.

Dedicated Host Configurations

Dedicated Hosts are configured to support a single instance type and size capacity. The number of instances you can launch onto a Dedicated Host depends on the instance type that the Dedicated Host is configured to support. For example, if you allocated a `c3.xlarge` Dedicated Host, you'd have the right to launch up to 8 `c3.xlarge` instances on the Dedicated Host. To determine the number of instance type sizes that you can run on a particular Dedicated Host, see [Amazon EC2 Dedicated Hosts Pricing](#).

Using Dedicated Hosts

To use a Dedicated Host, you first *allocate* hosts for use in your account. You then *launch* instances onto the hosts by specifying `host` tenancy for the instance. The *instance auto-placement* setting allows you to control whether an instance can launch onto a particular host. When an instance is stopped and restarted, the *Host affinity* setting determines whether it's restarted on the same, or a different, host. If you no longer need an On-Demand host, you can stop the instances running on the host, direct them to launch on a different host, and then *release* the Dedicated Host.

Contents

- [Bring Your Own License \(p. 267\)](#)
- [Allocating Dedicated Hosts \(p. 268\)](#)
- [Launching Instances onto Dedicated Hosts \(p. 268\)](#)
- [Understanding Instance Placement and Host Affinity \(p. 270\)](#)
- [Modifying Instance Tenancies \(p. 271\)](#)
- [Managing and Releasing Dedicated Hosts \(p. 271\)](#)
- [API and CLI Command Overview \(p. 272\)](#)
- [Tracking Configuration Changes with AWS Config \(p. 273\)](#)

Bring Your Own License

You can use your own software licenses on Dedicated Hosts. These are the general steps you need to follow in order to bring your own volume licensed machine image into Amazon EC2.

1. Verify that the license terms controlling the use of your machine images (AMIs) allow the usage of a machine image in a virtualized cloud environment. For more information about Microsoft Licensing, see [Amazon Web Services and Microsoft Licensing](#).
2. After you have verified that your machine image can be used within Amazon EC2, import your machine images using the `ImportImage` API operation made available by the VM Import/Export tools. For information about restrictions and limitations, see [VM Import/Export Prerequisites](#). For information about how to import your VM using `ImportImage`, see [Importing a VM into Amazon EC2 Using ImportImage](#).
3. If you need a mechanism to track how your images were used in AWS, enable host recording in the AWS Config service. You can use AWS Config to record configuration changes to a Dedicated Host and use the output as a data source for license reporting. For more information, see [Tracking Configuration Changes with AWS Config \(p. 273\)](#).
4. After you've imported your machine image, you can launch instances from this image onto active Dedicated Hosts in your account.
5. When you run these instances, depending on the operating system, you may be required to activate these instances against your own KMS server (for example, Windows Server or Windows SQL).

Server). You cannot activate your imported Windows AMI against the Amazon Windows KMS server.

Allocating Dedicated Hosts

To begin using Dedicated Hosts, they need to be allocated to your account. You can use the AWS Management Console, interact directly with the API, or use the command line interface to perform these tasks. Follow these steps every time you allocate a Dedicated Host.

To allocate Dedicated Hosts to your account

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Allocate Dedicated Host**.
3. Configure your host using the options provided:
 - a. **Instance type**—Instance type that will be available on the Dedicated Host.
 - b. **Availability Zone**—The Availability Zone for the Dedicated Host.
 - c. **Allow instance auto-placement**—The default setting is **Off**. The Dedicated Host accepts `host` tenancy instance launches only (provided capacity is available). When instance auto-placement is **On**, any instances with the tenancy of `host`, and matching the Dedicated Host's configuration, can be launched onto the host.
 - d. **Quantity**—The number of hosts to allocate with these settings.
4. Choose **Allocate host**.

The Dedicated Host capacity is made available in your account immediately.

If you launch instances with tenancy `host` but do not have any active Dedicated Hosts in your account, you receive an error and the instance launch fails.

Launching Instances onto Dedicated Hosts

After you have allocated a Dedicated Host, you can launch instances onto it. Instances with the tenancy `host` can be launched onto a specific Dedicated Host or Amazon EC2 can select the appropriate Dedicated Hosts for you (auto-placement). You cannot launch instances with the tenancy `host` if you do not have active Dedicated Hosts in your account with available capacity matching the instance type configuration of the instances you are launching.

Note

The instances launched onto Dedicated Hosts can only be launched in a VPC. For more information, see [Introduction to VPC](#).

Before you launch your instances, take note of the limitations. For more information, see [Dedicated Hosts Limitations and Restrictions \(p. 266\)](#).

Launching instances onto a Dedicated Host from the Dedicated Hosts page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select a host, choose **Actions** and then choose **Launch Instance(s) onto Host**.
3. Select the AMI to use. If you have imported your own AMI, choose **My AMIs** on the left sidebar and select the relevant AMI.
4. Choose the instance type for the Dedicated Host; this is the only instance type you can launch onto the host.
5. On the **Configure Instance Details** page, the **Tenancy** and **Host** options are pre-selected. You can toggle the **Affinity** setting to **On** or **Off**.
 - **On**—If stopped, the instance always restarts on that specific host.

- **Off**—The instance launches onto the specified Dedicated Host, but is not guaranteed to restart on it if stopped.
6. Complete the rest of the steps and choose **Launch Instances**.

The instance is automatically launched onto the Dedicated Host that you specified. To view the instances on a Dedicated Host, go to the **Dedicated Hosts** page, and select the Dedicated Host that you specified when you launched the instance.

Launching instances onto a specific Dedicated Host from the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, choose **Launch Instance**.
3. Select an AMI from the list. If you have imported your own AMI, choose **My AMIs** and select the imported image. Not all AMIs can be used with Dedicated Hosts.
4. Select the type of instance to launch.
5. On the **Configure Instance Details** page, the Dedicated Host settings are:
 - **Tenancy—Dedicated host — Launch this instance on a Dedicated host**. If you're not able to choose this, check whether you have selected an incompatible AMI or instance type.
 - **Host**—Select a host. If you are unable to select a Dedicated Host, check:
 - Whether the selected subnet is in a different Availability Zone to the host.
 - That the instance type you've selected matches the instance type that the Dedicated Host supports. If you don't have matching, running hosts, the only option available is **Use auto-placement** but the instance launch fails unless there is available, matching Dedicated Host capacity in your account.
 - **Affinity**—The default setting for this is **Off**. The instance launches onto the specified Dedicated Host, but is not guaranteed to restart on it if stopped.

Note

If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.

6. Complete the rest of the configuration steps. Choose **Review and Launch**.
7. Choose **Launch** to launch your instance.
8. Select an existing key pair, or create a new one. Choose **Launch Instances**.

Launching instances onto any Dedicated Host from the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, choose **Launch Instance**.
3. Select an AMI from the list. If you have imported your own AMI, choose **My AMIs** and select the imported image. Not all AMIs can be used with Dedicated Hosts.
4. Select the type of instance to launch.
5. On the **Configure Instance Details** page, the Dedicated Host settings are:
 - **Tenancy—Dedicated host — Launch this instance on a Dedicated host** If you're not able to choose this, check whether you have selected an incompatible AMI or instance type.
 - **Host**—For this type of launch, keep the setting as **Use auto-placement**.
 - **Affinity**—The default setting for this is **Off**. The instance launches onto any available Dedicated Host in your account, but is not guaranteed to restart on that host if stopped.

If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.

6. Complete the rest of the configuration steps. Choose **Review and Launch**.
7. Choose **Launch** to launch your instance.
8. Select an existing key pair, or create a new one. Choose **Launch Instances**.

Understanding Instance Placement and Host Affinity

Placement control happens on both the instance level and host level.

Contents

- [Instance Auto-Placement \(p. 270\)](#)
- [Host Affinity \(p. 270\)](#)
- [Modifying Instance Auto-Placement and Host Affinity \(p. 270\)](#)
- [Modifying Instance Host Affinity \(p. 271\)](#)

Instance Auto-Placement

Auto-placement allows you to manage whether instances that you launch are launched onto a specific host, or onto any host that has matching configurations. The default setting for this is **Off**. This means that the Dedicated Host you are allocating only accepts `host` tenancy instances launches that specify the unique host ID. Instances launched without a host ID specified are not able to launch onto a host that have instance auto-placement set to **Off**.

Host Affinity

Host Affinity establishes a launch relationship between an instance and a Dedicated Host. When affinity is set to `host`, an instance launched onto a specific host always restarts on the same host if stopped. This applies to both targeted and untargeted launches.

If affinity is set to `default`, and you stop and restart the instance, it can be restarted on any available host but tries to launch back onto the last Dedicated Host it ran on (on a best-effort basis).

You can modify the relationship between an instance and a Dedicated Host by changing the affinity from `host` to `default` and vice-versa. For more information, see [Modifying Instance Tenancies \(p. 271\)](#).

Modifying Instance Auto-Placement and Host Affinity

You can manage instance placement controls using the Amazon EC2 console, the API, or CLI.

To modify the instance placement settings of your instances, first stop the instances and then edit the instance placement settings.

Note

If the instance is stopped and restarted, it is not guaranteed to restart on the same Dedicated Host.

To edit an instance's placement settings (any available hosts)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, select the instance to edit.
3. Choose **Actions, Instance State**, and **Stop**.
4. Choose **Actions, Instance Settings**, and **Modify Instance Placement**.
5. Change the instance tenancy to **Launch this instance on a Dedicated host**.
6. Choose **This instance can run on any one of my Hosts**. The instance launches onto any Dedicated Host that has auto-placement enabled.
7. Choose **Save** to continue.
8. Open the context (right-click) menu on the instance and choose **Instance State, Start**.

To edit an instance's placement settings (specific Dedicated Host)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, select the instance to edit.
3. Choose **Actions**, **Instance State**, and **Stop**.
4. Choose **Actions**, **Instance Settings**, and **Modify Instance Placement**.
5. Change the instance tenancy to **Launch this instance on a Dedicated host**.
6. Choose **This instance can only run on the selected Host**. Then select a value for **Target Host** and choose whether you want the instance to be placed on any available host, or a specific host.
7. Choose **Save** to continue.
8. Open the context (right-click) menu on the instance and choose **Instance State**, **Start**.

Modifying Instance Host Affinity

If you no longer want an instance to have affinity with a host, you can stop the instance and change its affinity to `default`. This removes the persistence between the instance and the host. However, when you restart the instance, it may launch back onto the same Dedicated Host (depending on Dedicated Host availability in your account, and on a best-effort basis). However, if it is stopped again, it will not restart on the same host.

Modifying Instance Tenancies

You can modify the tenancy of a Dedicated Instance from `dedicated` to `host`, and vice-versa if it is not using a Windows, SUSE, or RHEL AMI provided by Amazon EC2. You need to stop your Dedicated Instance in order to do this. Instances with `shared` tenancy cannot be modified to `host` tenancy.

Modify instance tenancy from `dedicated` to `host`

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances**, then select the Dedicated Instances to modify.
3. Choose **Actions**, **Instance State**, and **Stop**.
4. Open the context (right-click) menu on the instance and choose **Instance Settings**, **Modify Instance Placement**.
5. On the **Modify Instance Placement** page, do the following:
 - **Tenancy**—Choose **Launch this instance on a Dedicated host**.
 - **Affinity**—Choose either **This instance can run on any one of my Hosts** or **This instance can only run on the selected Host**.

If you choose **This instance can run on any one of my Hosts**, the instance launches onto any available, compatible Dedicated Hosts in your account.

If you choose **This instance can only run on the selected Host**, select a value for **Target Host**. If no target host is listed, you may not have available, compatible Dedicated Hosts in your account.

6. Choose **Save**.
7. When you restart your instance Amazon EC2 places your instance on an available Dedicated Host in your account, provided it supports the instance type that you're launching.

Managing and Releasing Dedicated Hosts

You can use the console, interact directly with the API, or use the command line interface to view details about individual instances on a host and release an On-Demand Dedicated Host.

To view details of instances on a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select the host to view more information about.
3. Choose the **Description** tab for information about the host. Choose the **Instances** tab for information about instances running on your host.

To release a Dedicated Host

Any running instances on the Dedicated Host need to be stopped before you can release the host. These instances can be migrated to other Dedicated Hosts in your account so that you can continue to use them. For more information, see [Modifying Instance Auto-Placement and Host Affinity \(p. 270\)](#). These steps apply only to On-Demand Dedicated Hosts.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select the Dedicated Host to release.
3. Choose **Actions, Release Hosts**.
4. Confirm your choice by choosing **Release**.

After you release a Dedicated Host, you cannot reuse the same host or host ID again.

When the Dedicated Host is released you are no longer charged On-Demand billing rates for it. The Dedicated Host status is changed to `released` and you are not able to launch any instances onto that host.

If you've recently released Dedicated Hosts, it may take some time for them to stop counting towards your limit. During this time, you may experience `LimitExceeded` errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their `host` tenancy setting.

API and CLI Command Overview

You can perform the tasks described in this section using an API or the command line.

To allocate Dedicated Hosts to your account

- [allocate-hosts](#) (AWS CLI)
- [AllocateHosts](#) (Amazon EC2 Query API)
- [New-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To describe your Dedicated Hosts

- [describe-hosts](#) (AWS CLI)
- [DescribeHosts](#) (Amazon EC2 Query API)
- [Get-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To modify your Dedicated Hosts

- [modify-hosts](#) (AWS CLI)
- [ModifyHosts](#) (Amazon EC2 Query API)
- [Edit-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To modify instance auto-placement

- [modify-instance-placement](#) (AWS CLI)
- [ModifyInstancePlacement](#) (Amazon EC2 Query API)
- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

To release your Dedicated Hosts

- [release-hosts](#) (AWS CLI)
- [ReleaseHosts](#) (Amazon EC2 Query API)
- [Remove-EC2Hosts](#) (AWS Tools for Windows PowerShell)

Tracking Configuration Changes with AWS Config

You can use AWS Config to record configuration changes for Dedicated Hosts, and instances that are launched, stopped, or terminated on them. You can then use the information captured by AWS Config as a data source for license reporting.

AWS Config records configuration information for Dedicated Hosts and instances individually and pairs this information through relationships. There are three reporting conditions.

- **AWS Config recording status**—When **On**, AWS Config is recording one or more AWS resource types, which can include Dedicated Hosts and Dedicated Instances. To capture the information required for license reporting, verify that hosts and instances are being recorded with the following fields.
- **Host recording status**—When **Enabled**, the configuration information for Dedicated Hosts is recorded.
- **Instance recording status**—When **Enabled**, the configuration information for Dedicated Instances is recorded.

If any of these three conditions are disabled, the icon in the **Edit Config Recording** button is red. To derive the full benefit of this tool, ensure that all three recording methods are enabled. When all three are enabled, the icon is green. To edit the settings, choose **Edit Config Recording**. You are directed to the **Set up AWS Config** page in the AWS Config console, where you can set up AWS Config and start recording for your hosts, instances, and other supported resource types. For more information, see [Setting up AWS Config using the Console](#) in the *AWS Config Developer Guide*.

Note

AWS Config records your resources after it discovers them, which might take several minutes.

After AWS Config starts recording configuration changes to your hosts and instances, you can get the configuration history of any host that you have allocated or released and any instance that you have launched, stopped, or terminated. For example, at any point in the configuration history of a Dedicated Host, you can look up how many instances are launched on that host alongside the number of sockets and cores on the host. For any of those instances, you can also look up the ID of its Amazon Machine Image (AMI). You can use this information to report on licensing for your own server-bound software that is licensed per-socket or per-core.

You can view configuration histories in any of the following ways.

- By using the AWS Config console. For each recorded resource, you can view a timeline page, which provides a history of configuration details. To view this page, choose the grey icon in the **Config Timeline** column of the **Dedicated Hosts** page. For more information, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- By running AWS CLI commands. First, you can use the [list-discovered-resources](#) command to get a list of all hosts and instances. Then, you can use the [get-resource-config-history](#) command to get the

Amazon Elastic Compute Cloud User Guide for Linux Instances Dedicated Hosts

configuration details of a host or instance for a specific time interval. For more information, see [View Configuration Details Using the CLI](#) in the *AWS Config Developer Guide*.

- By using the AWS Config API in your applications. First, you can use the [ListDiscoveredResources](#) action to get a list of all hosts and instances. Then, you can use the [GetResourceConfigHistory](#) action to get the configuration details of a host or instance for a specific time interval.

For example, to get a list of all of your Dedicated Hosts from AWS Config, run a CLI command such as the following:

```
aws configservice list-discovered-resources --resource-type  
AWS::EC2::Host
```

To obtain the configuration history of a Dedicated Host from AWS Config, run a CLI command such as the following:

```
aws configservice get-resource-config-history --resource type  
AWS::EC2::Instance --resource-id i-36a47fdf
```

To manage AWS Config settings using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Edit Config Recording**.
3. In the AWS Config console, follow the steps provided to turn on recording. For more information, see [Setting up AWS Config using the Console](#).

For more information, see [Viewing Configuration Details in the AWS Config Console](#).

To activate AWS Config using the command line or API

- Using the AWS CLI, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- Using the Amazon EC2 API, see [GetResourceConfigHistory](#).

Monitoring Dedicated Hosts

Amazon EC2 constantly monitors the state of your Dedicated Hosts; updates are communicated on the Amazon EC2 console. You can also obtain information about your Dedicated Hosts using the API or CLI.

The following table illustrates the possible **State** values in the console.

State	Description
available	AWS hasn't detected an issue with the Dedicated Host; no maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.
released	The Dedicated Host has been released. The host ID is no longer in use. Released hosts cannot be reused.
under-assessment	AWS is exploring a possible issue with the Dedicated Host. If action needs to be taken,

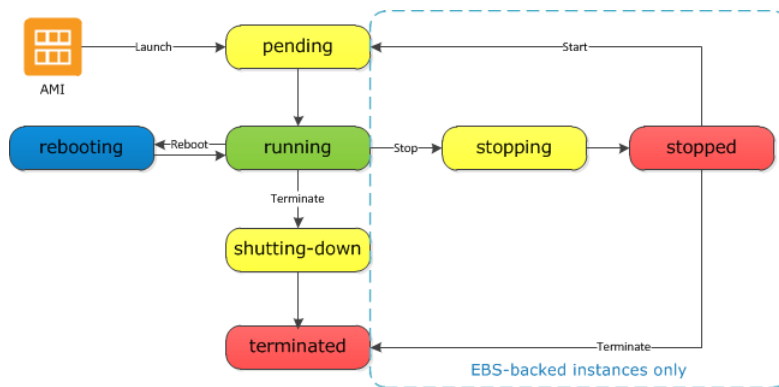
**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Lifecycle**

State	Description
	you will be notified via the AWS Management Console or email. Instances cannot be launched onto a Dedicated Host in this state.
permanent-failure	An unrecoverable failure has been detected. You will receive an eviction notice through your instances and by email. Your instances may continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. Instances cannot be launched onto Dedicated Hosts in this state.
released-permanent-failure	AWS permanently releases Dedicated Hosts that have failed and no longer have running instances on them. The Dedicated Host ID is no longer available for use.

Instance Lifecycle

By working with Amazon EC2 to manage your instances from the moment you launch them through their termination, you ensure that your customers have the best possible experience with the applications or sites that you host on your instances.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see [Storage for the Root Device \(p. 71\)](#).



Instance Launch

When you launch an instance, it enters the `pending` state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the `running` state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the `running` state, you're billed for each hour or partial hour that you keep the instance running; even if the instance remains idle and you don't connect to it.

For more information, see [Launch Your Instance \(p. 278\)](#) and [Connect to Your Linux Instance \(p. 289\)](#).

Instance Stop and Start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge hourly usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the `stopped` state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the `pending` state, and in most cases, we move the instance to a new host computer. (Your instance may stay on the same host computer if there are no problems with the host computer.) When you stop and start your instance, you'll lose any data on the instance store volumes on the previous host computer.

If your instance is running in EC2-Classic, it receives a new private IP address, which means that an Elastic IP address (EIP) associated with the private IP address is no longer associated with your instance. If your instance is running in EC2-VPC, it retains its private IP address, which means that an EIP associated with the private IP address or network interface is still associated with your instance.

Each time you transition an instance from `stopped` to `running`, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

For more information, see [Stop and Start Your Instance \(p. 299\)](#).

Instance Reboot

You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system; the instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing hour.

For more information, see [Reboot Your Instance \(p. 302\)](#).

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see [Instance Retirement \(p. 303\)](#).

Instance Termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Differences Between Reboot, Stop, and Terminate**

Note that if you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is deleted. You can also describe a terminated instance using the CLI and API. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate a shutdown from within the instance itself (for example, by using the **shutdown** command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see [Terminate Your Instance \(p. 305\)](#).

Differences Between Reboot, Stop, and Terminate

The following table summarizes the key differences between rebooting, stopping, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	The instance runs on a new host computer	None
Private and public IP addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IP addresses EC2-VPC: The instance keeps its private IP address. The instance gets a new public IP address, unless it has an Elastic IP address (EIP), which doesn't change during a stop/start.	None
Elastic IP addresses (EIP)	The EIP remains associated with the instance	EC2-Classic: The EIP is disassociated from the instance EC2-VPC: The EIP remains associated with the instance	The EIP is disassociated from the instance
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Launch**

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to <code>stopping</code> . Each time an instance transitions from <code>stopped</code> to <code>running</code> , we start a new instance billing hour.	You stop incurring charges for an instance as soon as its state changes to <code>shutting-down</code> .

Note that operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see [Changing the Instance Initiated Shutdown Behavior](#) (p. 307).

Launch Your Instance

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). You can either leverage the free tier to launch and use a micro instance for free for 12 months. If you launch an instance that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see the [Amazon EC2 Pricing](#).

You can launch an instance using the following methods.

Method	Documentation
Use the Amazon EC2 console with an AMI that you select	Launching an Instance (p. 279)
Use the Amazon EC2 console to launch an instance using an existing instance as a template	Launching an Instance Using an Existing Instance as a Template (p. 285)
Use the Amazon EC2 console with an Amazon EBS snapshot that you created	Launching a Linux Instance from a Backup (p. 286)
Use the Amazon EC2 console with an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 287)
Use the AWS CLI with an AMI that you select	Using Amazon EC2 through the AWS CLI
Use the AWS Tools for Windows PowerShell with an AMI that you select	Amazon EC2 from the AWS Tools for Windows PowerShell

After you launch your instance, you can connect to it and use it. To begin, the instance state is `pending`. When the instance state is `running`, the instance has started booting. There might be a short time before you can connect to the instance. The instance receives a public DNS name that you can use to contact the instance from the Internet. The instance also receives a private DNS name that other instances within the same Amazon EC2 network (EC2-Classic or EC2-VPC) can use to contact the instance. For more information about connecting to your instance, see [Connect to Your Linux Instance](#) (p. 289).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Your Instance](#) (p. 305).

Launching an Instance

Before you launch your instance, be sure that you are set up. For more information, see [Setting Up with Amazon EC2](#) (p. 19).

Your AWS account might support both the EC2-Classic and EC2-VPC platforms, depending on when you created your account and which regions you've used. To find out which platform your account supports, see [Supported Platforms](#) (p. 549). If your account supports EC2-Classic, you can launch an instance into either platform. If your account supports EC2-VPC only, you can launch an instance into a VPC only.

Important

When you launch an instance that's not within the [AWS Free Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Launching Your Instance from an AMI

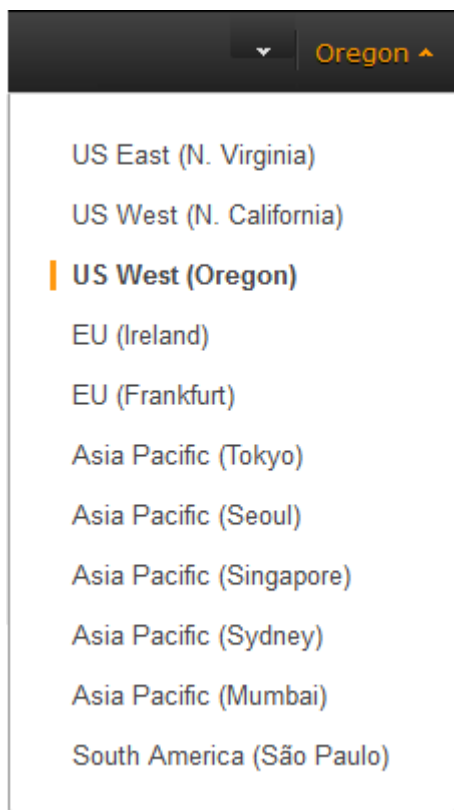
When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server: for example, Linux, Apache, and your web site.

Tip

To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations](#) (p. 743).



3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI as follows:
 - a. Select the type of AMI to use in the left pane:

Quick Start

A selection of popular AMIs to help you get started quickly. To ensure that you select an AMI that is eligible for the free tier, choose **Free tier only** in the left pane. (Notice that these AMIs are marked **Free tier eligible**.)

My AMIs

The private AMIs that you own, or private AMIs that have been shared with you.

AWS Marketplace

An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see [Launching an AWS Marketplace Instance \(p. 287\)](#).

Community AMIs

The AMIs that AWS community member have made available for others to use. To filter the list of AMIs by operating system, choose the appropriate check box under **Operating system**. You can also filter by architecture and root device type.

- b. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see [Storage for the Root Device \(p. 71\)](#).
- c. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM. For more information, see [Linux AMI Virtualization Types \(p. 74\)](#).
- d. Choose an AMI that meets your needs, and then choose **Select**.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see [Instance Types \(p. 156\)](#).

To remain eligible for the free tier, choose the **t2.micro** instance type. For more information, see [T2 Instances \(p. 160\)](#).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, choose **All generations** from the filter list.

Note

If you are new to AWS and would like to set up an instance quickly for testing purposes, you can choose **Review and Launch** at this point to accept default configuration settings, and launch your instance. Otherwise, to configure your instance further, choose **Next: Configure Instance Details**.

6. On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then choose **Next: Add Storage**:

- **Number of instances:** Enter the number of instances to launch.

Note

To help ensure that you maintain the correct number of instances to handle your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Auto Scaling User Guide](#).

- **Purchasing option:** Select **Request Spot instances** to launch a Spot instance. For more information, see [Spot Instances \(p. 219\)](#).
- Your account may support the EC2-Classic and EC2-VPC platforms, or EC2-VPC only. To find out which platform your account supports, see [Supported Platforms \(p. 549\)](#). If your account supports EC2-VPC only, you can launch your instance into your default VPC or a nondefault VPC. Otherwise, you can launch your instance into EC2-Classic or a nondefault VPC.

Note

Some instance types must be launched into a VPC. If you don't have a VPC, you can let the wizard create one for you.

To launch into EC2-Classic:

- **Network:** Select **Launch into EC2-Classic**.
- **Availability Zone:** Select the Availability Zone to use. To let AWS choose an Availability Zone for you, select **No preference**.

To launch into a VPC:

- **Network:** Select the VPC, or to create a new VPC, choose **Create new VPC** to go the Amazon VPC console. When you have finished, return to the wizard and choose **Refresh** to load your VPC in the list.
- **Subnet:** Select the subnet into which to launch your instance. If your account is EC2-VPC only, select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.
- **Auto-assign Public IP:** Specify whether your instance receives a public IP address. By default, instances in a default subnet receive a public IP address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see [Public IP Addresses and External DNS Hostnames \(p. 572\)](#).
- **IAM role:** Select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see [IAM Roles for Amazon EC2 \(p. 536\)](#).

- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 307\)](#).
 - **Enable termination protection:** Select this check box to prevent accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 306\)](#).
 - **Monitoring:** Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 419\)](#).
 - **EBS-Optimized instance:** An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see [Amazon EBS–Optimized Instances \(p. 692\)](#).
 - **Tenancy:** If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (**Dedicated**) or on a Dedicated host (**Dedicated host**). Additional charges may apply. For more information, see [Dedicated Instances](#) in the *Amazon VPC User Guide* and [Dedicated Hosts \(p. 264\)](#).
 - **Network interfaces:** If you selected a specific subnet, you can specify up to two network interfaces for your instance:
 - For **Network Interface**, select **New network interface** to let AWS create a new interface, or select an existing, available network interface.
 - For **Primary IP**, enter a private IP address from the range of your subnet, or leave **Auto-assign** to let AWS choose a private IP address for you.
 - Choose **Add IP** to assign more than one private IP address to the selected network interface.
 - Choose **Add Device** to add a secondary network interface. A secondary network interface can reside in a different subnet of the VPC, provided it's in the same Availability Zone as your instance.
For more information, see [Elastic Network Interfaces \(ENI\) \(p. 588\)](#). If you specify more than one network interface, your instance cannot receive a public IP address. Additionally, you cannot override the subnet's public IP setting using **Auto-assign Public IP** if you specify an existing network interface for eth0. For more information, see [Assigning a Public IP Address \(p. 575\)](#).
 - **Kernel ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.
 - **RAM disk ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
 - **Placement group:** A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see [Placement Groups \(p. 602\)](#).
 - **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.
7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then choose **Next: Tag Instance** when you have finished:
- **Type:** Select instance store or Amazon EBS volumes to associate with your instance. The type of volume available in the list depends on the instance type you've chosen. For more information, see [Amazon EC2 Instance Store \(p. 715\)](#) and [Amazon EBS Volumes \(p. 636\)](#).
 - **Device:** Select from the list of available device names for the volume.
 - **Snapshot:** Enter the name or ID of the snapshot from which to restore a volume. You can also search for public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.

- **Size:** For Amazon EBS-backed volumes, you can specify a storage size. Note that even if you have selected an AMI and instance that are eligible for the free tier, you need to keep under 30 GiB of total storage to stay within the free tier.

Note

Linux AMIs require GPT partition tables and GRUB 2 for boot volumes 2 TiB (2048 GiB) or larger. Many Linux AMIs today use the MBR partitioning scheme, which only supports up to 2047 GiB boot volumes. If your instance does not boot with a boot volume that is 2 TiB or larger, the AMI you are using may be limited to a 2047 GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances.

Note

If you increase the size of your root volume at this point (or any other volume created from a snapshot), you need to extend the file system on that volume in order to use the extra space. For more information about extending your file system after your instance has launched, see [Expanding the Storage Space of an EBS Volume on Linux \(p. 671\)](#).

- **Volume Type:** For Amazon EBS volumes, select either a General Purpose SSD, Provisioned IOPS SSD, or Magnetic volume. For more information, see [Amazon EBS Volume Types \(p. 638\)](#).

Note

If you select a Magnetic boot volume, you'll be prompted when you complete the wizard to make General Purpose SSD volumes the default boot volume for this instance and future console launches. (This preference persists in the browser session, and does not affect AMIs with Provisioned IOPS SSD boot volumes.) We recommended that you make General Purpose SSD volumes the default because they provide a much faster boot experience and they are the optimal volume type for most workloads. For more information, see [Amazon EBS Volume Types \(p. 638\)](#).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (io1) volumes. If you are unable to create an io1 volume (or launch an instance with an io1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports io1 volumes by creating a 4 GiB io1 volume in that zone.

- **IOPS:** If you have selected a Provisioned IOPS SSD volume type, then you can enter the number of I/O operations per second (IOPS) that the volume can support.
 - **Delete on Termination:** For Amazon EBS volumes, select this check box to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS Volumes on Instance Termination \(p. 308\)](#).
 - **Encrypted:** Select this check box to encrypt new Amazon EBS volumes. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes may only be attached to [supported instance types \(p. 697\)](#).
8. On the **Tag Instance** page, specify [tags \(p. 751\)](#) for the instance by providing key and value combinations. Choose **Create Tag** to add more than one tag to your resource. Choose **Next: Configure Security Group** when you are done.
 9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#).) Select or create a security group as follows, and then choose **Review and Launch**.

To select an existing security group:

1. Choose **Select an existing security group**. Your security groups are displayed. (If you are launching into EC2-Classic, these are security groups for EC2-Classic. If you are launching into a VPC, these are security group for that VPC.)
2. Select a security group from the list.
3. (Optional) You can't edit the rules of an existing security group, but you can copy them to a new group by choosing **Copy to new**. Then you can add rules as described in the next procedure.

To create a new security group:

1. Choose **Create a new security group**. The wizard automatically defines the launch-wizard-x security group.
2. (Optional) You can edit the name and description of the security group.
3. The wizard automatically defines an inbound rule to allow to you connect to your instance over SSH (port 22) for Linux or RDP (port 3389) for Windows.

Caution

This rule enables all IP addresses (0.0.0.0/0) to access your instance over the specified port. This is acceptable for this short exercise, but it's unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

4. You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow Internet traffic.

To add a rule, choose **Add Rule**, select the protocol to open to network traffic, and then specify the source. Choose **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.

When you are ready, choose **Launch**.

11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up.

To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

12. (Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose **Create status check alarms** and follow the directions. For more information, see [Creating and Editing Status Check Alarms \(p. 414\)](#).
13. If the instance state immediately goes to `terminated` instead of `running`, you can get information about why the instance didn't launch. For more information, see [What To Do If An Instance Immediately Terminates \(p. 773\)](#).

Launching an Instance Using an Existing Instance as a Template

The Amazon EC2 console provides a **Launch More Like This** wizard option that enables you to use a current instance as a template for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

Note

The **Launch More Like This** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

The following configuration details are copied from the selected instance into the launch wizard:

- AMI ID
- Instance type
- Availability Zone, or the VPC and subnet in which the selected instance is located
- Public IP address. If the selected instance currently has a public IP address, the new instance receives a public IP address - regardless of the selected instance's default public IP address setting. For more information about public IP addresses, see [Public IP Addresses and External DNS Hostnames](#) (p. 572).
- Placement group, if applicable
- IAM role associated with the instance, if applicable
- Shutdown behavior setting (stop or terminate)
- Termination protection setting (true or false)
- CloudWatch monitoring (enabled or disabled)
- Amazon EBS-optimization setting (true or false)
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Kernel ID and RAM disk ID, if applicable
- User data, if specified
- Tags associated with the instance, if applicable
- Security groups associated with the instance

The following configuration details are not copied from your selected instance; instead, the wizard applies their default settings or behavior:

- (VPC only) Number of network interfaces: The default is one network interface, which is the primary network interface (eth0).
- Storage: The default storage configuration is determined by the AMI and the instance type.

To use your current instance as a template

1. On the Instances page, select the instance you want to use.
2. Choose **Actions**, and then **Launch More Like This**.
3. The launch wizard opens on the **Review Instance Launch** page. You can check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

When you are ready, choose **Launch** to select a key pair and launch your instance.

Launching a Linux Instance from a Backup

With an Amazon EBS-backed Linux instance, you can back up the root device volume of the instance by creating a snapshot. When you have a snapshot of the root device volume of an instance, you can terminate that instance and then later launch a new instance from the snapshot. This can be useful if you don't have the original AMI that you launched an instance from, but you need to be able to launch an instance using the same image.

Important

Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the EC2 `billingProduct` code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot does not maintain this billing code, and subsequent instances launched from such an AMI will not be able to connect to package update infrastructure.

Similarly, although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from the AMI.

To create Windows AMIs or AMIs for Linux operating systems that must retain AMI billing codes to work properly, see [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#) or [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#).

Use the following procedure to create an AMI from the root volume of your instance using the console. If you prefer, you can use the [register-image](#) (AWS CLI) or [Register-EC2Image](#) (AWS Tools for Windows PowerShell) command instead.

To create an AMI from your root volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots**.
3. Choose **Create Snapshot**.
4. In the **Volumes** field, start typing the name or ID of the root volume, and then select it from the list of options.
5. Choose the snapshot that you just created, and then choose **Create Image** from the **Actions** list.
6. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, then choose **Create**. If you're re-creating a parent instance, then choose the same options as the parent instance.
 - **Architecture**: Choose **i386** for 32-bit or **x86_64** for 64-bit.
 - **Root device name**: Enter the appropriate name for the root volume. For more information, see [Device Naming on Linux Instances \(p. 729\)](#).
 - **Virtualization type**: Choose whether instances launched from this AMI use paravirtual (PV) or hardware virtual machine (HVM) virtualization. For more information, see [Linux AMI Virtualization Types \(p. 74\)](#).
 - (PV virtualization type only) **Kernel ID** and **RAM disk ID**: Choose the AKI and ARI from the lists. If you choose the default AKI or don't choose an AKI, you'll be required to specify an AKI every time you launch an instance using this AMI. In addition, your instance may fail the health checks if the default AKI is incompatible with the instance.
 - (Optional) **Block Device Mappings**: Add volumes or expand the default size of the root volume for the AMI. For more information about resizing the file system on your instance for a larger volume, see [Extending a Linux File System \(p. 673\)](#).
7. In the navigation pane, choose **AMIs**.
8. Choose the AMI that you just created, and then choose **Launch**. Follow the wizard to launch your instance. For more information about how to configure each step in the wizard, see [Launching an Instance \(p. 279\)](#).

Launching an AWS Marketplace Instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see [Paid AMIs \(p. 85\)](#). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see [Managing Your AWS Marketplace Subscriptions \(p. 88\)](#).

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Choose **Select** to choose your product.
4. A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, choose **Continue**.

Note

You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, choose **Next: Configure Instance Details**.
6. On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see [Launching an Instance \(p. 279\)](#). Choose **Next** until you reach the **Configure Security Group** page.

The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IP addresses (0.0.0.0/0) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

When you are ready, choose **Review and Launch**.

7. On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, choose **Launch** to select or create a key pair, and launch your instance.
8. Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose **View Instances** to go to the Instances page.

Note

You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the **running** state, you can connect to it. To do this, select your instance in the list and choose **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see [Connect to Your Linux Instance \(p. 289\)](#).

Important

Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see [Managing Your AWS Marketplace Subscriptions \(p. 88\)](#).

Launching an AWS Marketplace AMI Instance Using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

Connect to Your Linux Instance

Learn how to connect to the Linux instances that you launched and transfer files between your local computer and your instance.

If you need to connect to a Windows instance, see [Connecting to Your Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Your Computer	Topic
Linux	Connecting to Your Linux Instance Using SSH (p. 289)
Windows	Connecting to Your Linux Instance from Windows Using PuTTY (p. 293)
All	Connecting to Your Linux Instance Using MindTerm (p. 298)

After you connect to your instance, you can try one of our tutorials, such as [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 33\)](#) or [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 42\)](#).

Connecting to Your Linux Instance Using SSH

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

Note

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page.

The following instructions explain how to connect to your instance using an SSH client. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites:

- **Install an SSH client**
Your Linux computer most likely includes an SSH client by default. You can check for an SSH client by typing **ssh** at the command line. If your computer doesn't recognize the command, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, see <http://www.openssh.com>.
- **Install the AWS CLI Tools**
(Optional) If you're using a public AMI from a third party, you can use the command line tools to verify the fingerprint. For more information about installing the AWS CLI, see [Getting Set Up](#) in the *AWS Command Line Interface User Guide*.
- **Get the ID of the instance**
You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.
- **Get the public DNS name of the instance**
You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

- **Locate the private key**

You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see [Authorizing Network Access to Your Instances](#).

Important

Your default security group does not allow incoming SSH traffic by default.

Connecting to Your Linux Instance

Use the following procedure to connect to your Linux instance using an SSH client. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

To connect to your instance using SSH

1. (Optional) You can verify the RSA key fingerprint on your running instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.

- [get-console-output](#) (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

Note

Ensure that the instance is in the `running` state, not the `pending` state. The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. In a command line shell, change directories to the location of the private key file that you created when you launched the instance.
3. Use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

```
chmod 400 /path/my-key-pair.pem
```

4. Use the `ssh` command to connect to the instance. You'll specify the private key (`.pem`) file and `user_name@public_dns_name`. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is either `root` or `ec2-user`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

```
ssh -i /path/my-key-pair.pem ec2-  
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

You'll see a response like the following.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com  
(10.254.142.33)'  
can't be established.  
RSA key fingerprint is  
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
```

```
Are you sure you want to continue connecting (yes/no)?
```

- (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
- Enter `yes`.

You'll see a response like the following.

```
Warning: Permanently added  
'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

Transferring Files to Linux Instances from Linux Using SCP

One way to transfer files between your local computer and a Linux instance is to use Secure Copy (SCP). This section describes how to transfer files with SCP. The procedure is very similar to the procedure for connecting to an instance with SSH.

Prerequisites

- **Install an SCP client**

Most Linux, Unix, and Apple computers include an SCP client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools, including an SCP client. For more information, go to <http://www.openssh.org>.

- **Get the ID of the instance**

You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

- **Get the public DNS name of the instance**

You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

- **Locate the private key**

You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see [Authorizing Network Access to Your Instances](#).

Important

Your default security group does not allow incoming SSH traffic by default.

The following procedure steps you through using SCP to transfer a file. If you've already connected to the instance with SSH and have verified its fingerprints, you can start with the step that contains the SCP command (step 4).

To use SCP to transfer a file

- (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example,

1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

- `get-console-output` (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

Note

The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. In a command shell, change directories to the location of the private key file that you specified when you launched the instance.
3. Use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

```
chmod 400 /path/my-key-pair.pem
```

4. Transfer a file to your instance using the instance's public DNS name. For example, if the name of the private key file is `my-key-pair`, the file to transfer is `SampleFile.txt`, and the public DNS name of the instance is `ec2-198-51-100-1.compute-1.amazonaws.com`, use the following command to copy the file to the `ec2-user` home directory.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

Tip

For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is either `root` or `ec2-user`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

You'll see a response like the following.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

5. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
6. Enter `yes`.

You'll see a response like the following.

```
Warning: Permanently added
'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
Sending file modes: C0644 20 SampleFile.txt
Sink: C0644 20 SampleFile.txt
SampleFile.txt                               100%   20    0.0KB/s
00:00
```

Note

If you receive a "bash: scp: command not found" error, you must first install **scp** on your Linux instance. For some operating systems, this is located in the `openssh-clients` package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install **scp**.

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

To transfer files in the other direction (from your Amazon EC2 instance to your local computer), simply reverse the order of the host parameters. For example, to transfer the `SampleFile.txt` file from your EC2 instance back to the home directory on your local computer as `SampleFile2.txt`, use the following command on your local computer.

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/SampleFile2.txt
```

Connecting to Your Linux Instance from Windows Using PuTTY

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

Note

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page.

The following instructions explain how to connect to your instance using PuTTY, a free SSH client for Windows. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Prerequisites

Before you connect to your Linux instance using PuTTY, complete the following prerequisites:

- **Install PuTTY**
Download and install PuTTY from the [PuTTY download page](#). If you already have an older version of PuTTY installed, we recommend that you download the latest version. Be sure to install the entire suite.
- **Get the ID of the instance**
You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.
- **Get the public DNS name of the instance**
You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.
- **Locate the private key**
You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.
- **Enable inbound SSH traffic from your IP address to your instance**

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see [Authorizing Network Access to Your Instances](#).

Important

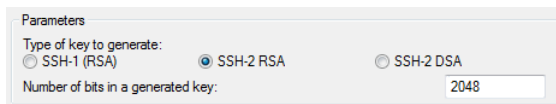
Your default security group does not allow incoming SSH traffic by default.

Converting Your Private Key Using PuTTYgen

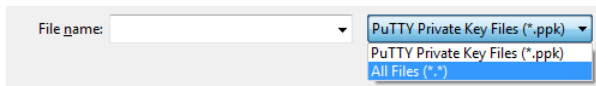
PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY.

To convert your private key

1. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
2. Under **Type of key to generate**, select **SSH-2 RSA**.



3. Click **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.



4. Select your .pem file for the key pair that you specified when you launch your instance, and then click **Open**. Click **OK** to dismiss the confirmation dialog box.
5. Click **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.

Note

A passphrase on a private key is an extra layer of protection, so even if your private key is discovered, it can't be used without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or copy files to an instance.

6. Specify the same name for the key that you used for the key pair (for example, `my-key-pair`). PuTTY automatically adds the .ppk file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

Starting a PuTTY Session

Use the following procedure to connect to your Linux instance using PuTTY. You'll need the .ppk file that you created for your private key. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

To start a PuTTY session

1. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example,

Amazon Elastic Compute Cloud User Guide for Linux Instances Connect

1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

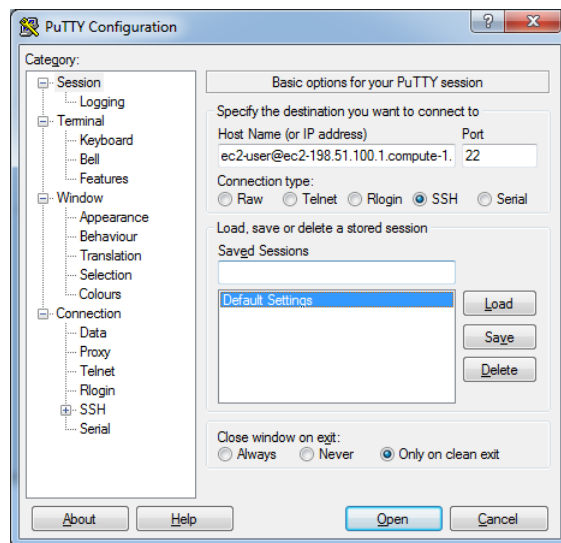
- `get-console-output` (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

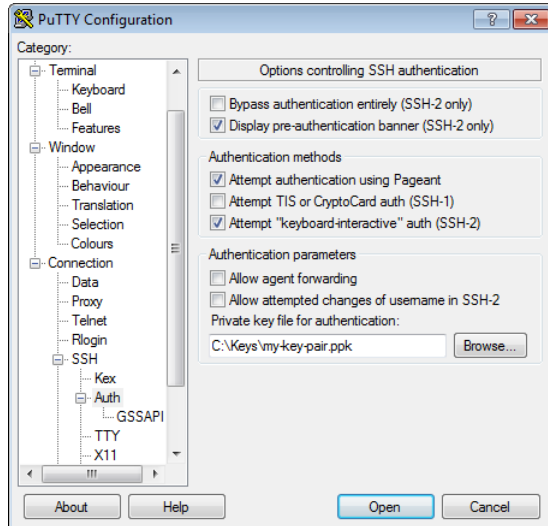
Note

The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. Start PuTTY (from the **Start** menu, click **All Programs > PuTTY > PuTTY**).
3. In the Category pane, select **Session** and complete the following fields:
 - a. In the **Host Name** box, enter `user_name@public_dns_name`. Be sure to specify the appropriate user name for your AMI. For example:
 - For an Amazon Linux AMI, the user name is `ec2-user`.
 - For a RHEL5 AMI, the user name is either `root` or `ec2-user`.
 - For an Ubuntu AMI, the user name is `ubuntu`.
 - For a Fedora AMI, the user name is either `fedora` or `ec2-user`.
 - For SUSE Linux, the user name is either `root` or `ec2-user`.
 - Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.
 - b. Under **Connection type**, select **SSH**.
 - c. Ensure that **Port** is 22.



4. In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**. Complete the following:
 - a. Click **Browse**.
 - b. Select the `.ppk` file that you generated for your key pair, and then click **Open**.
 - c. (Optional) If you plan to start this session again later, you can save the session information for future use. Select **Session** in the **Category** tree, enter a name for the session in **Saved Sessions**, and then click **Save**.
 - d. Click **Open** to start the PuTTY session.



5. If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host you are connecting to.
6. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
7. Click **Yes**. A window opens and you are connected to your instance.

Note

If you specified a passphrase when you converted your private key to PuTTY's format, you must provide that passphrase when you log in to the instance.

If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Transferring Files to Your Linux Instance Using the PuTTY Secure Copy Client

The PuTTY Secure Copy client (PSCP) is a command-line tool that you can use to transfer files between your Windows computer and your Linux instance. If you prefer a graphical user interface (GUI), you can use an open source GUI tool named WinSCP. For more information, see [Transferring Files to Your Linux Instance Using WinSCP](#) (p. 296).

To use PSCP, you'll need the private key you generated in [Converting Your Private Key Using PuTTYgen](#) (p. 294). You'll also need the public DNS address of your Linux instance.

The following example transfers the file `Sample_file.txt` from the `C:\` drive on a Windows computer to the `/usr/local` directory on a Linux instance:

```
C:\> pscp -i C:\Keys\my-key-pair.ppk C:\
\Sample_file.txt user_name@public_dns:/usr/local/Sample_file.txt
```

Transferring Files to Your Linux Instance Using WinSCP

WinSCP is a GUI-based file manager for Windows that allows you to upload and transfer files to a remote computer using the SFTP, SCP, FTP, and FTPS protocols. WinSCP allows you to drag and drop files from your Windows machine to your Linux instance or synchronize entire directory structures between the two systems.

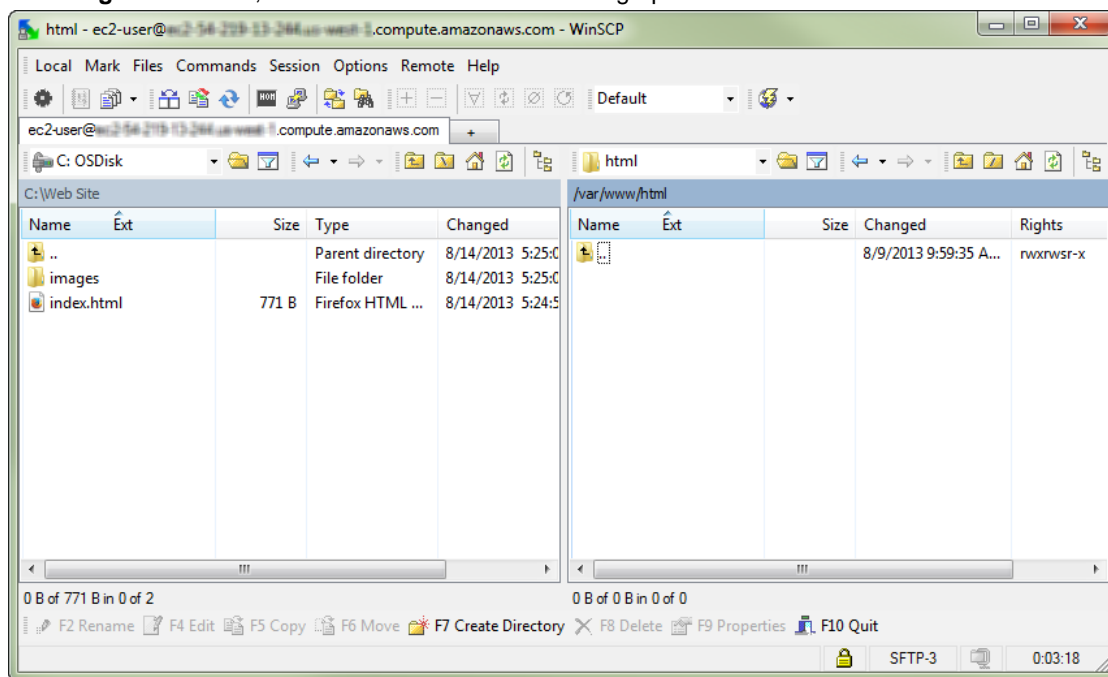
To use WinSCP, you'll need the private key you generated in [Converting Your Private Key Using PuTTYgen](#) (p. 294). You'll also need the public DNS address of your Linux instance.

1. Download and install WinSCP from <http://winscp.net/eng/download.php>. For most users, the default installation options are OK.
2. Start WinSCP.
3. At the **WinSCP login** screen, for **Host name**, enter the public DNS address for your instance.
4. For **User name**, enter the default user name for your AMI. For Amazon Linux AMIs, the user name is `ec2-user`. For Red Hat AMIs the user name is `root`, and for Ubuntu AMIs the user name is `ubuntu`.
5. Specify the private key for your instance. For **Private key**, enter the path to your private key, or click the `...` button to browse for the file. For newer versions of WinSCP, you need to click **Advanced** to open the advanced site settings and then under **SSH**, click **Authentication** to find the **Private key file** setting.

Note

WinSCP requires a PuTTY private key file (`.ppk`). You can convert a `.pem` security key file to the `.ppk` format using PuTTYgen. For more information, see [Converting Your Private Key Using PuTTYgen](#) (p. 294).

6. (Optional) In the left panel, click **Directories**, and then, for **Remote directory**, enter the path for the directory you want to add files to. For newer versions of WinSCP, you need to click **Advanced** to open the advanced site settings and then under **Environment**, click **Directories** to find the **Remote directory** setting.
7. Click **Login** to connect, and click **Yes** to add the host fingerprint to the host cache.



8. After the connection is established, in the connection window your Linux instance is on the right and your local machine is on the left. You can drag and drop files directly into the remote file system from your local machine. For more information on WinSCP, see the project documentation at <http://winscp.net/eng/docs/start>.

Note

If you receive a "Cannot execute SCP to start transfer" error, you must first install `scp` on your Linux instance. For some operating systems, this is located in the `openssh-clients` package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install `scp`.


```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

Connecting to Your Linux Instance Using MindTerm

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

Note

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page.

The following instructions explain how to connect to your instance using MindTerm through the Amazon EC2 console. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Important

The Chrome browser does not support the NPAPI plugin, and therefore cannot run the MindTerm client. For more information, go to the Chromium [NPAPI deprecation article](#). You can use Firefox, Safari, or Internet Explorer 9 or higher instead.

Prerequisites

- **Install Java**
Your Linux computer most likely includes Java. If not, see [How do I enable Java in my web browser?](#) On a Windows or Mac client, you must run your browser using administrator credentials. For Linux, additional steps may be required if you are not logged in as `root`.
- **Enable Java in your browser**
For instructions, see https://java.com/en/download/help/enable_browser.xml.
- **Locate the private key**
You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.
- **Enable inbound SSH traffic from your IP address to your instance**
Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see [Authorizing Network Access to Your Instances](#).

Important

Your default security group does not allow incoming SSH traffic by default.

Starting MindTerm

To connect to your instance using a web browser with MindTerm

1. In the Amazon EC2 console, click **Instances** in the navigation pane.
2. Select the instance, and then click **Connect**.
3. Click **A Java SSH client directly from my browser (Java required)**.
4. Amazon EC2 automatically detects the public DNS name of your instance and the name of the populates **Public DNS** for you. It also detects name of the key pair that you specified when you launched the instance. Complete the following, and then click **Launch SSH Client**.
 - a. In **User name**, enter the user name to log in to your instance.

Tip

For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user

name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is either `root` or `ec2-user`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

- b. In **Private key path**, enter the fully-qualified path to your private key (`.pem`) file, including the key pair name; for example:

```
C:\KeyPairs\my-key-pair.pem
```

- c. (Optional) Click **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.
5. If necessary, click **Yes** to trust the certificate, and click **Run** to run the MindTerm client.
6. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, to confirm setup for your home directory, and to confirm setup of the known hosts directory. Confirm these settings.
7. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, click **No**.
8. A window opens and you are connected to your instance.

Note

If you clicked **No** in the previous step, you'll see the following message, which is expected:

```
Verification of server key disabled in this session.
```

Stop and Start Your Instance

You can stop and restart your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the Overview section.

When you stop an instance, we shut it down. We don't charge hourly usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a stopped instance we charge a full instance hour, even if you make this transition multiple times within a single hour.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, we stop charging for that instance. For more information, see [Terminate Your Instance](#) (p. 305).

Contents

- [Overview](#) (p. 299)
- [Stopping and Starting Your Instances](#) (p. 300)
- [Modifying a Stopped Instance](#) (p. 301)
- [Troubleshooting](#) (p. 302)

Overview

You can only stop an Amazon EBS-backed instance. To verify the root device type of your instance, describe the instance and check whether the device type of its root volume is `ebs` (Amazon EBS-

backed instance) or `instance_store` (instance store-backed instance). For more information, see [Determining the Root Device Type of Your AMI \(p. 72\)](#).

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to `stopping` and then `stopped`.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- The instance is migrated to a new underlying host computer when it's started.
- EC2-Classic: We release the public and private IP addresses for the instance when you stop the instance, and assign new ones when you restart it.

EC2-VPC: The instance retains its private IP addresses when stopped and restarted. We release the public IP address and assign a new one when you restart it.

- EC2-Classic: We disassociate any Elastic IP address that's associated with the instance. You're charged for Elastic IP addresses that aren't associated with an instance. When you restart the instance, you must associate the Elastic IP address with the instance; we don't do this automatically.

EC2-VPC: The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance.

- When you stop and start a Windows instance, the EC2Config service performs tasks on the instance such as changing the drive letters for any attached Amazon EBS volumes. For more information about these defaults and how you can change them, see [Configuring a Windows Instance Using the EC2Config Service](#) in the *Amazon EC2 User Guide for Windows Instances*.
- If you've registered the instance with a load balancer, it's likely that the load balancer won't be able to route traffic to your instance after you've stopped and restarted it. You must de-register the instance from the load balancer after stopping the instance, and then re-register after starting the instance. For more information, see [Register or Deregister EC2 Instances for Your Classic Load Balancer](#) in the *Classic Load Balancer Guide*.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Auto Scaling User Guide*.
- When you stop a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after restarting it. For more information about ClassicLink, see [ClassicLink \(p. 550\)](#).

For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 277\)](#).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

Stopping and Starting Your Instances

You can start and stop your Amazon EBS-backed instance using the console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the **shutdown**, **halt**, or **poweroff** command), the instance stops. You can change this behavior so that it terminates instead. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 307\)](#).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, choose **Instances**, and select the instance.
2. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
3. Choose **Actions**, select **Instance State**, and then choose **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

5. While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a Stopped Instance \(p. 301\)](#).
6. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
7. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

[EC2-Classic] When the instance state becomes `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.

8. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions**, and then select **Associate Address**.
 - d. Select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

To stop and start an Amazon EBS-backed instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `stop-instances` and `start-instances` (AWS CLI)
- `Stop-EC2Instance` and `Start-EC2Instance` (AWS Tools for Windows PowerShell)

Modifying a Stopped Instance

You can change the instance type, user data, and EBS-optimization attributes of a stopped instance using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the `DeleteOnTermination`, `kernel`, or `RAM disk` attributes.

To modify an instance attribute

- To change the instance type, see [Resizing Your Instance \(p. 189\)](#).
- To change the user data for your instance, see [Configuring Instances with User Data \(p. 340\)](#).
- To enable or disable EBS–optimization for your instance, see [Modifying EBS–Optimization \(p. 695\)](#).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Updating the Block Device Mapping of a Running Instance \(p. 737\)](#).

To modify an instance attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `modify-instance-attribute` (AWS CLI)
- `Edit-EC2InstanceAttribute` (AWS Tools for Windows PowerShell)

Troubleshooting

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, you can forcibly stop it. For more information, see [Troubleshooting Stopping Your Instance \(p. 779\)](#).

Reboot Your Instance

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see [Scheduled Events for Your Instances \(p. 415\)](#).

We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance. If you use Amazon EC2 to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within four minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, select **Instance State**, and then select **Reboot**.
4. Choose **Yes, Reboot** when prompted for confirmation.

To reboot an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [reboot-instances](#) (AWS CLI)
- [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell)

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

Topics

- [Identifying Instances Scheduled for Retirement](#) (p. 303)
- [Working with Instances Scheduled for Retirement](#) (p. 304)

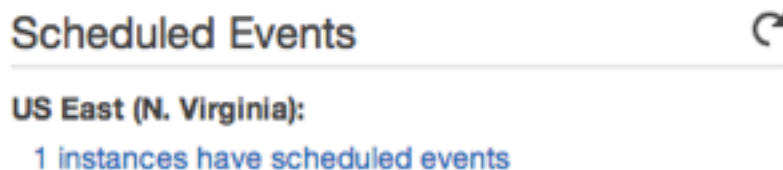
For more information about types of instance events, see [Scheduled Events for Your Instances](#) (p. 415).

Identifying Instances Scheduled for Retirement

If your instance is scheduled for retirement, you'll receive an email prior to the event with the instance ID and retirement date. This email is sent to the address that's associated with your account; the same email address that you use to log in to the AWS Management Console. If you use an email account that you do not check regularly, then you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement. To update the contact information for your account, go to the [Account Settings](#) page.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **EC2 Dashboard**. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes, organized by region.



3. If you have an instance with a scheduled event listed, select its link below the region name to go to the **Events** page.
4. The **Events** page lists all resources with events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [describe-instance-status](#) (AWS CLI)
- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)

Working with Instances Scheduled for Retirement

There are a number of actions available to you when your instance is scheduled for retirement. The action you take depends on whether your instance root device is an Amazon EBS volume, or an instance store volume. If you do not know what your instance root device type is, you can find out using the Amazon EC2 console or the command line.

Determining Your Instance Root Device Type

To determine your instance root device type using the console

1. In the navigation pane, select **Events**. Use the filter lists to identify retiring instances, as demonstrated in the procedure above, [Identifying instances scheduled for retirement \(p. 303\)](#).
2. In the **Resource ID** column, select the instance ID to go to the **Instances** page.
3. Select the instance and locate the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed.

To determine your instance root device type using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Managing Instances Scheduled for Retirement

You can perform one of the actions listed below in order to preserve the data on your retiring instance. It's important that you take this action before the instance retirement date, to prevent unforeseen downtime and data loss.

Warning

If your instance store-backed instance passes its retirement date, it's terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if they are attached to an EBS-backed instance.

Instance Root Device Type	Action
EBS	Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private and Elastic IP addresses associated with your instance, see Stop and Start Your Instance (p. 299) .
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 89) .

Instance Root Device Type	Action
Instance store	Create an instance store-backed AMI from your instance using the AMI tools, and launch a replacement instance. For more information, see Creating an Instance Store-Backed Linux AMI (p. 92) .
Instance store	Convert your instance to an EBS-backed instance by transferring your data to an EBS volume, taking a snapshot of the volume, and then creating an AMI from the snapshot. You can launch a replacement instance from your new AMI. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 129) .

Terminate Your Instance

When you've decided that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or restart an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and restart your instance, see [Stop and Start Your Instance \(p. 299\)](#). For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 277\)](#).

Topics

- [Instance Termination \(p. 305\)](#)
- [Terminating an Instance \(p. 306\)](#)
- [Enabling Termination Protection for an Instance \(p. 306\)](#)
- [Changing the Instance Initiated Shutdown Behavior \(p. 307\)](#)
- [Preserving Amazon EBS Volumes on Instance Termination \(p. 308\)](#)
- [Troubleshooting \(p. 310\)](#)

Instance Termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You cannot delete the terminated instance entry yourself.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, by default, any additional EBS volumes that you attach at launch, or any EBS volumes that you attach to an existing instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see [Preserving Amazon EBS Volumes on Instance Termination \(p. 308\)](#).

You can prevent an instance from being terminated accidentally by someone using the AWS Management Console, the CLI, and the API. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute with the default value of `false` (the instance can be terminated through Amazon EC2). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances). For more information, see [Enabling Termination Protection for an Instance \(p. 306\)](#).

You can control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 307\)](#).

If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

Terminating an Instance

You can terminate an instance using the AWS Management Console or the command line.

To terminate an instance using the console

1. Before you terminate the instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, select **Instances**.
4. Select the instance, choose **Actions**, select **Instance State**, and then select **Terminate**.
5. Select **Yes, Terminate** when prompted for confirmation.

To terminate an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `terminate-instances` (AWS CLI)
- `Stop-EC2Instance` (AWS Tools for Windows PowerShell)

Enabling Termination Protection for an Instance

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. If you want to prevent your instance from being accidentally terminated using Amazon EC2, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The `DisableApiTermination` attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 307\)](#).

You can't prevent instances that are part of an Auto Scaling group from terminating using termination protection. However, you can specify which instances should terminate first. For more information, see [Choosing a Termination Policy](#) in the *Auto Scaling User Guide*.

You can't enable termination protection for Spot instances — a Spot instance is terminated when the Spot price exceeds your bid price. However, you can prepare your application to handle Spot instance interruptions. For more information, see [Spot Instance Interruptions \(p. 260\)](#).

You can enable or disable termination protection using the AWS Management Console or the command line.

To enable termination protection for an instance at launch time

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance** and follow the directions in the wizard.
3. On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1. Select the instance, choose **Actions, Instance Settings**, and then choose **Change Termination Protection**.
2. Select **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, select **Actions**, select **Instance Settings**, and then choose **Change Termination Protection**.
2. Select **Yes, Disable**.

To enable or disable termination protection using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `modify-instance-attribute` (AWS CLI)
- `Edit-EC2InstanceAttribute` (AWS Tools for Windows PowerShell)

Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown**, **halt**, or **poweroff**), the instance stops. You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line. The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself; it does not apply when you stop an instance using the `StopInstances` API or the Amazon EC2 console.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, select **Actions, Instance Settings**, and then choose **Change Shutdown Behavior**. The current behavior is already selected.
4. To change the behavior, select an option from the **Shutdown behavior** list, and then select **Apply**.



To change the shutdown behavior of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Preserving Amazon EBS Volumes on Instance Termination

When an instance terminates, Amazon EC2 uses the value of the `DeleteOnTermination` attribute for each attached Amazon EBS volume to determine whether to preserve or delete the volume.

By default, the `DeleteOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of an instance when the instance terminates.

By default, when you attach an EBS volume to an instance, its `DeleteOnTermination` attribute is set to `false`. Therefore, the default is to preserve these volumes. After the instance terminates, you can take a snapshot of the preserved volume or attach it to another instance.

To verify the value of the `DeleteOnTermination` attribute for an EBS volume that is in-use, look at the instance's block device mapping. For more information, see [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 738\)](#).

You can change value of the `DeleteOnTermination` attribute for a volume when you launch the instance or while the instance is running.

Examples

- [Changing the Root Volume to Persist at Launch Using the Console \(p. 308\)](#)
- [Changing the Root Volume to Persist at Launch Using the Command Line \(p. 309\)](#)
- [Changing the Root Volume of a Running Instance to Persist Using the Command Line \(p. 309\)](#)

Changing the Root Volume to Persist at Launch Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, select **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then choose **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume to Persist at Launch Using the Command Line

When you launch an instance, you can use one of the following commands to change the root device volume to persist. The root device is typically `/dev/sda1`. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

For example, add the following option to your `run-instances` command:

```
--block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`:

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false,
      "SnapshotId": "snap-1234567890abcdef0",
      "VolumeType": "gp2"
    }
  }
]
```

Changing the Root Volume of a Running Instance to Persist Using the Command Line

You can use one of the following commands to change the root device volume of a running instance to persist. The root device is typically `/dev/sda1`. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

For example, use the following command:

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --  
block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`:

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

Troubleshooting

If your instance is in the `shutting-down` state for longer than usual, it will eventually be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see [Troubleshooting Terminating \(Shutting Down\) Your Instance \(p. 781\)](#).

Recover Your Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata. For more information about using Amazon CloudWatch alarms to recover an instance, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 450\)](#). To troubleshoot issues with instance recovery failures, see [Troubleshooting Instance Recovery Failures](#) in the *Amazon EC2 User Guide for Linux Instances*.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important

The recover action is only supported on:

- C3, C4, M3, M4, R3, T2, and X1 instance types.
- Instances in a VPC

Note

If your instance has a public IP address, it retains the same public IP address after recovery.

- Instances with shared tenancy (where the tenancy attribute of the instance is set to `default`)
- Instances that use Amazon EBS storage exclusively

Currently, the recover action is not supported on:

- EC2-Classic instances
- Dedicated tenancy instances
- Instances running on dedicated hosts
- Instances with encrypted EBS volumes
- Instances that use any instance store volumes, including instances launched with block device mappings for instance store volumes

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Configuring Your Amazon Linux Instance

After you have successfully launched and logged into your Amazon Linux instance, you can make changes to it. There are many different ways you can configure an instance to meet the needs of a specific application. The following are some common tasks to help get you started.

Contents

- [Common Configuration Scenarios \(p. 311\)](#)
- [Managing Software on Your Linux Instance \(p. 312\)](#)
- [Managing User Accounts on Your Linux Instance \(p. 320\)](#)
- [Processor State Control for Your EC2 Instance \(p. 322\)](#)
- [Setting the Time for Your Linux Instance \(p. 326\)](#)
- [Changing the Hostname of Your Linux Instance \(p. 330\)](#)
- [Setting Up Dynamic DNS on Your Linux Instance \(p. 332\)](#)
- [Running Commands on Your Linux Instance at Launch \(p. 334\)](#)
- [Instance Metadata and User Data \(p. 337\)](#)

Common Configuration Scenarios

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code. For more

information on installing and building software from these locations, see [Managing Software on Your Linux Instance \(p. 312\)](#).

Amazon Linux instances come pre-configured with an `ec2-user` account, but you may want to add other user accounts that do not have super-user privileges. For more information on adding and removing user accounts, see [Managing User Accounts on Your Linux Instance \(p. 320\)](#).

The default time configuration for Amazon Linux instances uses Network Time Protocol to set the system time on an instance. The default time zone is UTC. For more information on setting the time zone for an instance or using your own time server, see [Setting the Time for Your Linux Instance \(p. 326\)](#).

If you have your own network with a domain name registered to it, you can change the hostname of an instance to identify itself as part of that domain. You can also change the system prompt to show a more meaningful name without changing the hostname settings. For more information, see [Changing the Hostname of Your Linux Instance \(p. 330\)](#). You can configure an instance to use a dynamic DNS service provider. For more information, see [Setting Up Dynamic DNS on Your Linux Instance \(p. 332\)](#).

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2, `cloud-init` directives, and shell scripts. For more information, see [Running Commands on Your Linux Instance at Launch \(p. 334\)](#).

Managing Software on Your Linux Instance

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code.

Contents

- [Updating Instance Software \(p. 313\)](#)
- [Adding Repositories \(p. 316\)](#)
- [Finding Software Packages \(p. 317\)](#)
- [Installing Software Packages \(p. 318\)](#)
- [Preparing to Compile Software \(p. 319\)](#)

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. For more information, see [Updating Instance Software \(p. 313\)](#).

By default, Amazon Linux instances launch with two repositories enabled: `amzn-main` and `amzn-updates`. While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository. For more information, see [Adding Repositories \(p. 316\)](#). For help finding packages in enabled repositories, see [Finding Software Packages \(p. 317\)](#). For information about installing software on an Amazon Linux instance, see [Installing Software Packages \(p. 318\)](#).

Not all software is available in software packages stored in repositories; some software must be compiled on an instance from its source code. For more information, see [Preparing to Compile Software \(p. 319\)](#).

Amazon Linux instances manage their software using the `yum` package manager. The `yum` package manager can install, remove, and update software, as well as manage all of the dependencies for each package. Debian-based Linux distributions, like Ubuntu, use the `apt-get` command and `dpkg` package manager, so the `yum` examples in the following sections do not work for those distributions.

Updating Instance Software

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. When you first launch and connect to an Amazon Linux instance, you may see a message asking you to update software packages for security purposes. This section shows how to update an entire system, or just a single package.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

```
  _|  _|_ )  
 _| (  /   Amazon Linux AMI  
__|\__|__|
```

```
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/  
There are 12 security update(s) out of 25 total update(s) available  
Run "sudo yum update" to apply all updates.  
[ec2-user ~]$
```

To update all packages on an Amazon Linux instance

- (Optional) Start a **screen** session in your shell window. Sometimes you may experience a network interruption that can disconnect the SSH connection to your instance. If this happens during a long software update, it can leave the instance in a recoverable, although confused state. A **screen** session allows you to continue running the update even if your connection is interrupted, and you can reconnect to the session later without problems.
 - Execute the **screen** command to begin the session.

```
[ec2-user ~]$ screen
```

- If your session is disconnected, log back into your instance and list the available screens.

```
[ec2-user ~]$ screen -ls  
There is a screen on:  
 17793.pts-0.ip-12-34-56-78 (Detached)  
1 Socket in /var/run/screen/S-ec2-user.
```

- Reconnect to the screen using the **screen -r** command and the process ID from the previous command.

```
[ec2-user ~]$ screen -r 17793
```

- When you are finished using **screen**, use the **exit** command to close the session.

```
[ec2-user ~]$ exit  
[screen is terminating]
```

- Run the **yum update** command. Optionally, you can add the `--security` flag to apply only security updates.

```
[ec2-user ~]$ sudo yum update  
Loaded plugins: priorities, security, update-motd, upgrade-helper  
amzn-main | 2.1 kB  
00:00
```


**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Managing Software**

```

amzn-updates | 2.3 kB
00:00
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package aws-apitools-ec2.noarch 0:1.6.8.1-1.0.amzn1 will be updated
---> Package aws-apitools-ec2.noarch 0:1.6.10.0-1.0.amzn1 will be an
update
---> Package gnupg2.x86_64 0:2.0.18-1.16.amzn1 will be updated
---> Package gnupg2.x86_64 0:2.0.19-8.21.amzn1 will be an update
---> Package libgcrypt.i686 0:1.4.5-9.10.amzn1 will be updated
---> Package libgcrypt.x86_64 0:1.4.5-9.10.amzn1 will be updated
---> Package libgcrypt.i686 0:1.4.5-9.12.amzn1 will be an update
---> Package libgcrypt.x86_64 0:1.4.5-9.12.amzn1 will be an update
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
---> Package python-boto.noarch 0:2.9.9-1.0.amzn1 will be updated
---> Package python-boto.noarch 0:2.13.3-1.0.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version                Repository
Size
=====
Updating:
aws-apitools-ec2  noarch   1.6.10.0-1.0.amzn1    amzn-updates
14 M
gnupg2            x86_64   2.0.19-8.21.amzn1    amzn-updates
2.4 M
libgcrypt         i686     1.4.5-9.12.amzn1     amzn-updates
248 k
libgcrypt         x86_64   1.4.5-9.12.amzn1     amzn-updates
262 k
openssl           x86_64   1:1.0.1e-4.54.amzn1  amzn-updates
1.7 M
python-boto       noarch   2.13.3-1.0.amzn1     amzn-updates
1.6 M

Transaction Summary
=====
Upgrade           6 Package(s)

Total download size: 20 M
Is this ok [y/N]:

```

- Review the packages listed, and type **y** and **Enter** to accept the updates. Updating all of the packages on a system can take several minutes. The **yum** output shows the status of the update while it is running.

```

Downloading Packages:
(1/6): aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch.rpm | 14 MB
00:00
(2/6): gnupg2-2.0.19-8.21.amzn1.x86_64.rpm | 2.4 MB
00:00
(3/6): libgcrypt-1.4.5-9.12.amzn1.i686.rpm | 248 kB
00:00

```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Managing Software**

```
(4/6): libgcrypt-1.4.5-9.12.amzn1.x86_64.rpm | 262 kB
00:00
(5/6): openssl-1.0.1e-4.54.amzn1.x86_64.rpm | 1.7 MB
00:00
(6/6): python-boto-2.13.3-1.0.amzn1.noarch.rpm | 1.6 MB
00:00
-----
Total                28 MB/s | 20 MB
00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : libgcrypt-1.4.5-9.12.amzn1.x86_64
1/12
  Updating   : gnupg2-2.0.19-8.21.amzn1.x86_64
2/12
  Updating   : aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch
3/12
  Updating   : 1:openssl-1.0.1e-4.54.amzn1.x86_64
4/12
  ...
Complete!
```

4. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any `glibc` libraries should also be followed by a reboot. For updates to packages that control services, it may be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

To update a single package on an Amazon Linux instance

Use this procedure to update a single package (and its dependencies) and not the entire system.

1. Run the `yum update` command with the name of the package you would like to update.

```
[ec2-user ~]$ sudo yum update openssl
Loaded plugins: priorities, security, update-motd, upgrade-helper
amzn-main | 2.1 kB
00:00
amzn-updates | 2.3 kB
00:00
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository
Size
=====
Updating:
```

```
openssl          x86_64          1:1.0.1e-4.54.amzn1      amzn-updates
1.7 M

Transaction Summary
=====
Upgrade          1 Package(s)

Total download size: 1.7 M
Is this ok [y/N]:
```

2. Review the package information listed, and type **y** and **Enter** to accept the update or updates. Sometimes there will be more than one package listed if there are package dependencies that must be resolved. The **yum** output shows the status of the update while it is running.

```
Downloading Packages:
openssl-1.0.1e-4.54.amzn1.x86_64.rpm | 1.7 MB
00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Cleanup    : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2
  Verifying  : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Verifying  : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2

Updated:
  openssl.x86_64 1:1.0.1e-4.54.amzn1

Complete!
```

3. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any `glibc` libraries should also be followed by a reboot. For updates to packages that control services, it may be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

Adding Repositories

By default, Amazon Linux instances launch with two repositories enabled: `amzn-main` and `amzn-updates`. While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

To install a package from a different repository with **yum**, you need to add the repository information to the `/etc/yum.conf` file or to its own `repository.repo` file in the `/etc/yum.repos.d` directory. You can do this manually, but most yum repositories provide their own `repository.repo` file at their repository URL.

To add a yum repository to `/etc/yum.repos.d`

1. Find the location of the `.repo` file. This will vary depending on the repository you are adding. In this example, the `.repo` file is at `https://www.example.com/repository.repo`.
2. Add the repository with the `yum-config-manager` command.

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://
www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo | 4.0 kB 00:00
repo saved to /etc/yum.repos.d/repository.repo
```

To enable a yum repository in `/etc/yum.repos.d`

- Use the `yum-config-manager` command with the `--enable repository` flag. The following command enables the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project. By default, this repository is present in `/etc/yum.repos.d` on Amazon Linux instances, but it is not enabled.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Note

For information on enabling the EPEL repository on other distributions, such as Red Hat and CentOS, see the EPEL documentation at <https://fedoraproject.org/wiki/EPEL>.

Finding Software Packages

You can use the `yum search` command to search the descriptions of packages that are available in your configured repositories. This is especially helpful if you don't know the exact name of the package you want to install. Simply append the keyword search to the command; for multiple word searches, wrap the search query with quotation marks.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

Multiple word search queries in quotation marks only return results that match the exact query. If you don't see the expected package, simplify your search to one keyword and then scan the results. You can also try keyword synonyms to broaden your search.

```
[ec2-user ~]$ sudo yum search "find"
Loaded plugins: priorities, security, update-motd, upgrade-helper
===== N/S Matched: find
=====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative
interface
                           : to File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png,
jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png,
```

```
      : jpg)  
mlocate.x86_64 : An utility for finding files by name
```

The yum package manager also combines several packages into groups that you can install with one command to perform a particular task, such as installing a web server or build tools for software compilation. To list the groups that are already installed on your system and the available groups that you can install, use the **yum grouplist** command.

```
[ec2-user ~]$ sudo yum grouplist  
Loaded plugins: priorities, security, update-motd, upgrade-helper  
Setting up Group Process  
Installed Groups:  
  Development Libraries  
  Development tools  
  Editors  
  Legacy UNIX compatibility  
  Mail Server  
  MySQL Database  
  Network Servers  
  Networking Tools  
  PHP Support  
  Perl Support  
  System Tools  
  Web Server  
Available Groups:  
  Console internet tools  
  DNS Name Server  
  FTP Server  
  Java Development  
  MySQL Database client  
  NFS file server  
  Performance Tools  
  PostgreSQL Database client (version 8)  
  PostgreSQL Database server (version 8)  
  Scientific support  
  TeX support  
  Technical Writing  
  Web Servlet Engine  
Done
```

You can see the different packages in a group by using the **yum groupinfo** "*Group Name*" command, replacing *Group Name* with the name of the group to get information about. This command lists all of the mandatory, default, and optional packages that can be installed with that group.

If you cannot find the software you need in the default `amzn-main` and `amzn-updates` repositories, you can add more repositories, such as the Extra Packages for Enterprise Linux (EPEL) repository. For more information, see [Adding Repositories](#) (p. 316).

Installing Software Packages

The yum package manager is a great tool for installing software, because it can search all of your enabled repositories for different software packages and also handle any dependencies in the software installation process.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

To install a package from a repository, use the **yum install** *package* command, replacing *package* with the name of the software to install. For example, to install the **links** text-based web browser, enter the following command.

```
[ec2-user ~]$ sudo yum install links
```

To install a group of packages, use the **yum groupinstall** *Group Name* command, replacing *Group Name* with the name of the group you would like to install. For example, to install the "Performance Tools" group, enter the following command.

```
[ec2-user@ip-10-161-113-54 ~]$ sudo yum groupinstall "Performance Tools"
```

By default, yum will only install the mandatory and default packages in the group listing. If you would like to install the optional packages in the group also, you can set the `group_package_types` configuration parameter in the command when you execute it that adds the optional packages.

```
[ec2-user ~]$ sudo yum --  
setopt=group_package_types=mandatory,default,optional groupinstall  
"Performance Tools"
```

You can also use **yum install** to install RPM package files that you have downloaded from the Internet. To do this, simply append the path name of an RPM file to the installation command instead of a repository package name.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

Preparing to Compile Software

There is a wealth of open-source software available on the Internet that has not been pre-compiled and made available for download from a package repository. You may eventually discover a software package that you need to compile yourself, from its source code. For your system to be able to compile software, you need to install several development tools, such as **make**, **gcc**, and **autoconf**.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

Because software compilation is not a task that every Amazon EC2 instance requires, these tools are not installed by default, but they are available in a package group called "Development Tools" that is easily added to an instance with the **yum groupinstall** command.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

Software source code packages are often available for download (from web sites such as <https://github.com/> and <http://sourceforge.net/>) as a compressed archive file, called a tarball. These tarballs will usually have the `.tar.gz` file extension. You can decompress these archives with the **tar** command.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

After you have decompressed and unarchived the source code package, you should look for a `README` or `INSTALL` file in the source code directory that can provide you with further instructions for compiling and installing the source code.

To retrieve source code for Amazon Linux packages

Amazon Web Services provides the source code for maintained packages. You can download the source code for any installed packages with the `get_reference_source` command.

- Run the `get_reference_source -p package` command to download the source code for *package*. For example, to download the source code for the `htop` package, enter the following command.

```
[ec2-user ~]$ get_reference_source -p htop

Requested package: htop
Found package from local RPM database: htop-1.0.1-2.3.amzn1.x86_64
Corresponding source RPM to found package : htop-1.0.1-2.3.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to: /usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm
```

The command output lists the location of the source RPM, in this case `/usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm`.

Managing User Accounts on Your Linux Instance

Each Linux instance type launches with a default Linux system user account. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is either `root` or `ec2-user`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

Note

Linux system users should not be confused with AWS Identity and Access Management (IAM) users. For more information, see [IAM Users and Groups](#) in the *IAM User Guide*.

Using the default user account is adequate for many applications, but you may choose to add user accounts so that individuals can have their own files and workspaces. Creating user accounts for new users is much more secure than granting multiple (possibly inexperienced) users access to the `ec2-user` account, since that account can cause a lot of damage to a system when used improperly.

To add a new user to the system

Effectively adding users to a Linux instance involves two basic operations: adding the user to the system, and providing that user with a way to log in remotely.

1. To add a new user to the system, use the `adduser` command followed by any relevant options and the name of the user you wish to create.

Important

If you are adding a user to an Ubuntu system, you should add the `--disabled-password` option to avoid adding a password to the account.

```
[ec2-user ~]$ sudo adduser newuser
```

This command adds the `newuser` account to the system (with an entry in the `/etc/passwd` file), creates a `newuser` group, and creates a home directory for the account in `/home/newuser`.

2. To provide remote access to this account, you must create a `.ssh` directory in the `newuser` home directory and create a file within it named `"authorized_keys"` that contains a public key.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Managing Users**

- a. Switch to the new account so that newly created files have the proper ownership.

```
[ec2-user ~]$ sudo su - newuser
[newuser ~]$
```

Note that the prompt now says `newuser` instead of `ec2-user`; you have switched the shell session to the new account.

- b. Create a `.ssh` directory for the `authorized_keys` file.

```
[newuser ~]$ mkdir .ssh
```

- c. Change the file permissions of the `.ssh` directory to `700` (this means only the file owner can read, write, or open the directory).

Important

This step is very important; without these exact file permissions, you will not be able to log into this account using SSH.

```
[newuser ~]$ chmod 700 .ssh
```

- d. Create a file named "authorized_keys" in the `.ssh` directory.

```
[newuser ~]$ touch .ssh/authorized_keys
```

- e. Change the file permissions of the `authorized_keys` file to `600` (this means only the file owner can read or write to the file).

Important

This step is very important; without these exact file permissions, you will not be able to log into this account using SSH.

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

- f. Edit the `authorized_keys` file with your favorite text editor and paste the public key for your key pair into the file, for example:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXr
lsLnBItnctckiJ7FbtxJMXLvwwJryDUilBMTjYtwB+QhYXUMozce5Pjz5/
i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco
+CY/5WrUBkrHmFJr6HcXkvJdWPKYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Note

For more information about creating a key pair, see [Creating Your Key Pair Using Amazon EC2 \(p. 482\)](#). For more information about retrieving a public key from an existing key pair, see [Retrieving the Public Key for Your Key Pair on Linux \(p. 484\)](#).

You should now be able to log into the `newuser` account on your instance via SSH using the private key that matches the public key from [Step 2.f \(p. 321\)](#).

To remove a user from the system

If a user account is no longer needed, you can remove that account so that it may no longer be used.

- To delete a user account, the user's home directory, and the user's mail spool, execute the **userdel -r** command followed by the user name you wish to delete.

```
[ec2-user ~]$ sudo userdel -r olduser
```

Note

To keep the user's home directory and mail spool, omit the `-r` option.

Processor State Control for Your EC2 Instance

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off). P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

The following instance types provide the ability for an operating system to control processor C-states and P-states:

- `c4.8xlarge`
- `d2.8xlarge`
- `m4.10xlarge`
- `x1.32xlarge`

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload. The default C-state and P-state settings provide maximum performance, which is optimal for most workloads. However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

The following sections describe the different processor state configurations and how to monitor the effects of your configuration. These procedures were written for, and apply to Amazon Linux; however, they may also work for other Linux distributions with a Linux kernel version of 3.9 or newer. For more information about other Linux distributions and processor state control, see your system-specific documentation.

Note

The examples on this page use the **turbostat** utility (which is available on Amazon Linux by default) to display processor frequency and C-state information, and the **stress** command (which can be installed by running **sudo yum install -y stress**) to simulate a workload.

Contents

- [Highest Performance with Maximum Turbo Boost Frequency \(p. 323\)](#)
- [High Performance and Low Latency by Limiting Deeper C-states \(p. 324\)](#)
- [Baseline Performance with the Lowest Variability \(p. 325\)](#)

Highest Performance with Maximum Turbo Boost Frequency

This is the default processor state control configuration for the Amazon Linux AMI, and it is recommended for most workloads. This configuration provides the highest performance with lower variability. Allowing inactive cores to enter deeper sleep states provides the thermal headroom required for single or dual core processes to reach their maximum Turbo Boost potential.

The following example shows a `c4.8xlarge` instance with two cores actively performing work reaching their maximum processor Turbo Boost frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU   %c0  GHz  TSC SMI   %c1   %c3   %c6   %c7   %pc2   %pc3
 %pc6   %pc7  Pkg_W RAM_W PKG_% RAM_%
 5.54 3.44 2.90  0  9.18  0.00 85.28  0.00  0.00  0.00
0.00  0.00 94.04 32.70 54.18  0.00
0  0  0  0.12 3.26 2.90  0  3.61  0.00 96.27  0.00  0.00  0.00
0.00  0.00 48.12 18.88 26.02  0.00
0  0 18  0.12 3.26 2.90  0  3.61
0  1  1  0.12 3.26 2.90  0  4.11  0.00 95.77  0.00
0  1 19  0.13 3.27 2.90  0  4.11
0  2  2  0.13 3.28 2.90  0  4.45  0.00 95.42  0.00
0  2 20  0.11 3.27 2.90  0  4.47
0  3  3  0.05 3.42 2.90  0 99.91  0.00  0.05  0.00
0  3 21  97.84 3.45 2.90  0  2.11
...
  1  1 10  0.06 3.33 2.90  0 99.88  0.01  0.06  0.00
1  1 28  97.61 3.44 2.90  0  2.32
...
10.002556 sec
```

In this example, vCPUs 21 and 28 are running at their maximum Turbo Boost frequency because the other cores have entered the C6 sleep state to save power and provide both power and thermal headroom for the working cores. vCPUs 3 and 10 (each sharing a processor core with vCPUs 21 and 28) are in the C1 state, waiting for instruction.

In the following example, all 18 cores are actively performing work, so there is no headroom for maximum Turbo Boost, but they are all running at the "all core Turbo Boost" speed of 3.2 GHz.

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU   %c0  GHz  TSC SMI   %c1   %c3   %c6   %c7   %pc2   %pc3
 %pc6   %pc7  Pkg_W RAM_W PKG_% RAM_%
99.27 3.20 2.90  0  0.26  0.00  0.47  0.00  0.00  0.00
0.00  0.00 228.59 31.33 199.26  0.00
0  0  0 99.08 3.20 2.90  0  0.27  0.01  0.64  0.00  0.00  0.00
0.00  0.00 114.69 18.55 99.32  0.00
0  0 18 98.74 3.20 2.90  0  0.62
0  1  1 99.14 3.20 2.90  0  0.09  0.00  0.76  0.00
0  1 19 98.75 3.20 2.90  0  0.49
0  2  2 99.07 3.20 2.90  0  0.10  0.02  0.81  0.00
0  2 20 98.73 3.20 2.90  0  0.44
0  3  3 99.02 3.20 2.90  0  0.24  0.00  0.74  0.00
0  3 21 99.13 3.20 2.90  0  0.13
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Processor State Control**

```
0 4 4 99.26 3.20 2.90 0 0.09 0.00 0.65 0.00
0 4 22 98.68 3.20 2.90 0 0.67
0 5 5 99.19 3.20 2.90 0 0.08 0.00 0.73 0.00
0 5 23 98.58 3.20 2.90 0 0.69
0 6 6 99.01 3.20 2.90 0 0.11 0.00 0.89 0.00
0 6 24 98.72 3.20 2.90 0 0.39
...
```

High Performance and Low Latency by Limiting Deeper C-states

C-states control the sleep levels that a core may enter when it is inactive. You may want to control C-states to tune your system for latency versus performance. Putting cores to sleep takes time, and although a sleeping core allows more headroom for another core to boost to a higher frequency, it takes time for that sleeping core to wake back up and perform work. For example, if a core that is assigned to handle network packet interrupts is asleep, there may be a delay in servicing that interrupt. You can configure the system to not use deeper C-states, which reduces the processor reaction latency, but that in turn also reduces the headroom available to other cores for Turbo Boost.

A common scenario for disabling deeper sleep states is a Redis database application, which stores the database in system memory for the fastest possible query response time.

To limit deeper sleep states on Amazon Linux

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the `kernel` line of the first entry and add the `intel_idle.max_cstate=1` option to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. Save the file and exit your editor.
4. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

The following example shows a `c4.8xlarge` instance with two cores actively performing work at the "all core Turbo Boost" core frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5322] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3
%pc6 %pc7 Pkg_W RAM_W PKG_% RAM_%
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Processor State Control

```
          5.56 3.20 2.90  0 94.44  0.00  0.00  0.00  0.00  0.00
0.00  0.00 131.90 31.11 199.47 0.00
0  0  0  0.03 2.08 2.90  0 99.97  0.00  0.00  0.00  0.00  0.00
0.00  0.00 67.23 17.11 99.76 0.00
0  0 18  0.01 1.93 2.90  0 99.99
0  1  1  0.02 1.96 2.90  0 99.98  0.00  0.00  0.00
0  1 19 99.70 3.20 2.90  0  0.30
...
1  1 10  0.02 1.97 2.90  0 99.98  0.00  0.00  0.00
1  1 28 99.67 3.20 2.90  0  0.33
1  2 11  0.04 2.63 2.90  0 99.96  0.00  0.00  0.00
1  2 29  0.02 2.11 2.90  0 99.98
...
```

In this example, the cores for vCPUs 19 and 28 are running at 3.2 GHz, and the other cores are in the C1 C-state, awaiting instruction. Although the working cores are not reaching their maximum Turbo Boost frequency, the inactive cores will be much faster to respond to new requests than they would be in the deeper C6 C-state.

Baseline Performance with the Lowest Variability

You can reduce the variability of processor frequency with P-states. P-states control the desired performance (in CPU frequency) from a core. Most workloads perform better in P0, which requests Turbo Boost. But you may want to tune your system for consistent performance rather than bursty performance that can happen when Turbo Boost frequencies are enabled.

Intel Advanced Vector Extensions (AVX or AVX2) workloads can perform well at lower frequencies, and AVX instructions can use more power. Running the processor at a lower frequency, by disabling Turbo Boost, can reduce the amount of power used and keep the speed more consistent. For more information about optimizing your instance configuration and workload for AVX, see <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf>.

This section describes how to limit deeper sleep states and disable Turbo Boost (by requesting the P1 P-state) to provide low-latency and the lowest processor speed variability for these types of workloads.

To limit deeper sleep states and disable Turbo Boost on Amazon Linux

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the kernel line of the first entry and add the `intel_idle.max_cstate=1` option to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
  intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. Save the file and exit your editor.
4. Reboot your instance to enable the new kernel option.

Amazon Elastic Compute Cloud User Guide for Linux Instances Setting the Time

```
[ec2-user ~]$ sudo reboot
```

5. When you need the low processor speed variability that the P1 P-state provides, execute the following command to disable Turbo Boost.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/  
no_turbo"
```

6. When your workload is finished, you can re-enable Turbo Boost with the following command.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/  
no_turbo"
```

The following example shows a `c4.8xlarge` instance with two vCPUs actively performing work at the baseline core frequency, with no Turbo Boost.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10  
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd  
stress: info: [5389] successful run completed in 10s  
pk cor CPU   %c0  GHz  TSC SMI   %c1   %c3   %c6   %c7   %pc2   %pc3  
%pc6   %pc7  Pkg_W RAM_W PKG_% RAM_%  
      5.59 2.90 2.90   0 94.41  0.00  0.00  0.00  0.00  0.00  
0.00  0.00 128.48 33.54 200.00  0.00  
0  0  0  0.04 2.90 2.90   0 99.96  0.00  0.00  0.00  0.00  0.00  
0.00  0.00  65.33 19.02 100.00  0.00  
0  0 18  0.04 2.90 2.90   0 99.96  
0  1  1  0.05 2.90 2.90   0 99.95  0.00  0.00  0.00  
0  1 19  0.04 2.90 2.90   0 99.96  
0  2  2  0.04 2.90 2.90   0 99.96  0.00  0.00  0.00  
0  2 20  0.04 2.90 2.90   0 99.96  
0  3  3  0.05 2.90 2.90   0 99.95  0.00  0.00  0.00  
0  3 21 99.95 2.90 2.90   0  0.05  
...  
1  1 28 99.92 2.90 2.90   0  0.08  
1  2 11  0.06 2.90 2.90   0 99.94  0.00  0.00  0.00  
1  2 29  0.05 2.90 2.90   0 99.95
```

The cores for vCPUs 21 and 28 are actively performing work at the baseline processor speed of 2.9 GHz, and all inactive cores are also running at the baseline speed in the C1 C-state, ready to accept instructions.

Setting the Time for Your Linux Instance

A consistent and accurate time reference is crucial for many server tasks and processes. Most system logs include a time stamp that you can use to determine when problems occur and in what order the events take place. If you use the AWS CLI or an AWS SDK to make requests from your instance, these tools sign requests on your behalf. If your instance's date and time are not set correctly, the date in the signature may not match the date of the request, and AWS rejects the request. Network Time Protocol (NTP) is configured by default on Amazon Linux instances, and the system time is synchronized with a load-balanced pool of public servers on the Internet and set to the UTC time zone. For more information about NTP, go to <http://www.ntp.org/>.

Tasks

- [Changing the Time Zone \(p. 327\)](#)

- [Configuring Network Time Protocol \(NTP\) \(p. 328\)](#)

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

Changing the Time Zone

Amazon Linux instances are set to the UTC (Coordinated Universal Time) time zone by default, but you may wish to change the time on an instance to the local time or to another time zone in your network.

To change the time zone on an instance

1. Identify the time zone to use on the instance. The `/usr/share/zoneinfo` directory contains a hierarchy of time zone data files. Browse the directory structure at that location to find a file for your time zone.

```
[ec2-user ~]$ ls /usr/share/zoneinfo
Africa      Chile      GB         Indian     Mideast    posixrules US
America     CST6CDT   GB-Eire    Iran       MST        PRC        UTC
Antarctica  Cuba      GMT        iso3166.tab MST7MDT    PST8PDT    WET
Arctic      EET       GMT0       Israel     Navajo     right      W-SU
...
```

Some of the entries at this location are directories (such as `America`), and these directories contain time zone files for specific cities. Find your city (or a city in your time zone) to use for the instance. In this example, you can use the time zone file for Los Angeles, `/usr/share/zoneinfo/America/Los_Angeles`.

2. Update the `/etc/sysconfig/clock` file with the new time zone.
 - a. Open the `/etc/sysconfig/clock` file with your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with your editor command because `/etc/sysconfig/clock` is owned by `root`.
 - b. Locate the `ZONE` entry, and change it to the time zone file (omitting the `/usr/share/zoneinfo` section of the path). For example, to change to the Los Angeles time zone, change the `ZONE` entry to the following.

```
ZONE="America/Los_Angeles"
```

Note

Do not change the `UTC=true` entry to another value. This entry is for the hardware clock, and does not need to be adjusted when you're setting a different time zone on your instance.

- c. Save the file and exit the text editor.
3. Create a symbolic link between `/etc/localtime` and your time zone file so that the instance finds the time zone file when it references local time information.

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. Reboot the system to pick up the new time zone information in all services and applications.

```
[ec2-user ~]$ sudo reboot
```

Configuring Network Time Protocol (NTP)

Network Time Protocol (NTP) is configured by default on Amazon Linux instances; however, an instance needs access to the Internet for the standard NTP configuration to work. In addition, your instance's security group rules must allow outbound UDP traffic on port 123 (NTP), and your network ACL rules must allow both inbound and outbound UDP traffic on port 123. The procedures in this section show how to verify that the default NTP configuration is working correctly. If your instance does not have access to the Internet, you need to configure NTP to query a different server in your private network to keep accurate time.

To verify that NTP is working properly

1. Use the `ntpstat` command to view the status of the NTP service on the instance.

```
[ec2-user ~]$ ntpstat
```

If your output resembles the output below, then NTP is working properly on the instance.

```
synchronised to NTP server (12.34.56.78) at stratum 3
time correct to within 399 ms
polling server every 64 s
```

If your output states, "unsynchronised", wait a minute and try again. The first synchronization may take a minute to complete.

If your output states, "Unable to talk to NTP daemon. Is it running?", you probably need to start the NTP service and enable it to automatically start at boot time.

2. (Optional) You can use the `ntpq -p` command to see a list of peers known to the NTP server and a summary of their state.

```
[ec2-user ~]$ ntpq -p
      remote               refid      st t when poll reach  delay  offset
 jitter
=====
+littleman.deekay 204.9.54.119      2 u  15 128 377  88.649  5.946
 6.876
-bittorrent.tomh 91.189.94.4       3 u 133 128 377 182.673  8.001
 1.278
*ntp3.junkemailf 216.218.254.202  2 u  68 128 377  29.377  4.726
11.887
+tesla.selinc.co 149.20.64.28     2 u  31 128 377  28.586 -1.215
 1.435
```

If the output of this command shows no activity, check whether your security groups, network ACLs, or firewalls block access to the NTP port.

To start and enable NTP

1. Start the NTP service with the following command.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd: [ OK ]
```

2. Enable NTP to start at boot time with the `chkconfig` command.

Amazon Elastic Compute Cloud User Guide for Linux Instances Setting the Time

```
[ec2-user ~]$ sudo chkconfig ntpd on
```

3. Verify that NTP is enabled with the following command.

```
[ec2-user ~]$ sudo chkconfig --list ntpd
ntpd          0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Here **ntpd** is on in runlevels 2, 3, 4, and 5, which is correct.

To change NTP servers

You may decide not to use the standard NTP servers or you may need to use your own NTP server within your private network for instances that do not have Internet access.

1. Open the `/etc/ntp.conf` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/ntp.conf` is owned by `root`.
2. Find the `server` section, which defines the servers to poll for NTP configuration.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.amazon.pool.ntp.org iburst
server 1.amazon.pool.ntp.org iburst
server 2.amazon.pool.ntp.org iburst
server 3.amazon.pool.ntp.org iburst
```

Note

The `n.amazon.pool.ntp.org` DNS records are intended to load balance NTP traffic from AWS. However, these are public NTP servers in the `pool.ntp.org` project, and they are not owned or managed by AWS. There is no guarantee that they are geographically located near your instances, or even within the AWS network. For more information, see <http://www.pool.ntp.org/en/>.

3. Comment out the servers you don't want to use by adding a `#` character to the beginning of those server definitions.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.amazon.pool.ntp.org iburst
#server 1.amazon.pool.ntp.org iburst
#server 2.amazon.pool.ntp.org iburst
#server 3.amazon.pool.ntp.org iburst
```

4. Add an entry for each server to poll for time synchronization. You can use a DNS name for this entry or a dotted quad IP address (such as `10.0.0.254`).

```
server my-ntp-server.my-domain.com iburst
```

5. Restart the NTP service to pick up the new servers.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd: [ OK ]
```

6. Verify that your new settings work and that NTP is functioning.


```
[ec2-user ~]$ ntpstat  
synchronised to NTP server (64.246.132.14) at stratum 2  
time correct to within 99 ms
```

Changing the Hostname of Your Linux Instance

When you launch an instance, it is assigned a hostname that is a form of the private, internal IP address. A typical Amazon EC2 private DNS name looks something like this: `ip-12-34-56-78.us-west-2.compute.internal`, where the name consists of the internal domain, the service (in this case, `compute`), the region, and a form of the private IP address. Part of this hostname is displayed at the shell prompt when you log into your instance (for example, `ip-12-34-56-78`). Each time you stop and restart your Amazon EC2 instance (unless you are using an Elastic IP address), the public IP address changes, and so does your public DNS name, system hostname, and shell prompt. Instances launched into EC2-Classic also receive a new private IP address, private DNS hostname, and system hostname when they're stopped and restarted; instances launched into a VPC don't.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

Changing the System Hostname

If you have a public DNS name registered for the IP address of your instance (such as `webserver.mydomain.com`), you can set the system hostname so your instance identifies itself as a part of that domain. This also changes the shell prompt so that it displays the first portion of this name instead of the hostname supplied by AWS (for example, `ip-12-34-56-78`). If you do not have a public DNS name registered, you can still change the hostname, but the process is a little different.

To change the system hostname to a public DNS name

Follow this procedure if you already have a public DNS name registered.

1. On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
HOSTNAME=webserver.mydomain.com
```

2. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

3. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the `hostname` command should show the fully qualified domain name.

```
[ec2-user@webserver ~]$ hostname  
webserver.mydomain.com
```

To change the system hostname without a public DNS name

1. Open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the desired system hostname (such as `webserver`).

```
HOSTNAME=webserver.localdomain
```

2. Open the `/etc/hosts` file in your favorite text editor and change the entry beginning with `127.0.0.1` to match the example below, substituting your own hostname.

```
127.0.0.1 webserver.localdomain webserver localhost localhost.localdomain
```

3. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

4. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the `hostname` command should show the fully qualified domain name.

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

Changing the Shell Prompt Without Affecting the Hostname

If you do not want to modify the hostname for your instance, but you would like to have a more useful system name (such as `webserver`) displayed than the private name supplied by AWS (for example, `ip-12-34-56-78`), you can edit the shell prompt configuration files to display your system nickname instead of the hostname.

To change the shell prompt to a host nickname

1. Create a file in `/etc/profile.d` that sets the environment variable called `NICKNAME` to the value you want in the shell prompt. For example, to set the system nickname to `webserver`, execute the following command.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/  
profile.d/prompt.sh'
```

2. Open the `/etc/bashrc` file in your favorite text editor (such as `vim` or `nano`). You need to use `sudo` with the editor command because `/etc/bashrc` is owned by `root`.
3. Edit the file and change the shell prompt variable (`PS1`) to display your nickname instead of the hostname. Find the following line that sets the shell prompt in `/etc/bashrc` (several surrounding lines are shown below for context; look for the line that starts with `["$PS1"`):

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\s-\v\\$ " ] && PS1="\u@h \W\\$ "  
# You might want to have e.g. tty in prompt (e.g. more virtual machines)  
# and console windows
```

And change the `\h` (the symbol for `hostname`) in that line to the value of the `NICKNAME` variable.

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\s-\v\\$ " ] && PS1="\u@$NICKNAME \W\\$ "
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Setting Up Dynamic DNS

```
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

- (Optional) To set the title on shell windows to the new nickname, complete the following steps.

- Create a file called `/etc/sysconfig/bash-prompt-xterm`.

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- Make the file executable with the following command.

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- Open the `/etc/sysconfig/bash-prompt-xterm` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/sysconfig/bash-prompt-xterm` is owned by `root`.
- Add the following line to the file.

```
echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
```

- Log out and then log back in to pick up the new nickname value.

Changing the Hostname on Other Linux Distributions

The above procedures are intended for use with Amazon Linux only. For more information about other Linux distributions, see their specific documentation and the following articles:

- [How do I assign a static hostname to a private Amazon EC2 instance running RHEL 7 or Centos 7?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running SuSe Linux?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running Ubuntu Linux?](#)

Setting Up Dynamic DNS on Your Linux Instance

When you launch an EC2 instance, it is assigned a public IP address and a public DNS (Domain Name System) name that you can use to reach it from the Internet. Because there are so many hosts in the Amazon Web Services domain, these public names must be quite long for each name to remain unique. A typical Amazon EC2 public DNS name looks something like this: `ec2-12-34-56-78.us-west-2.compute.amazonaws.com`, where the name consists of the Amazon Web Services domain, the service (in this case, `compute`), the region, and a form of the public IP address.

Dynamic DNS services provide custom DNS host names within their domain area that can be easy to remember and that can also be more relevant to your host's use case; some of these services are also free of charge. You can use a dynamic DNS provider with Amazon EC2 and configure the instance to update the IP address associated with a public DNS name each time the instance starts. There are many different providers to choose from, and the specific details of choosing a provider and registering a name with them are outside the scope of this guide.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

To use dynamic DNS with Amazon EC2

- Sign up with a dynamic DNS service provider and register a public DNS name with their service. This procedure uses the free service from noip.com/free as an example.

Amazon Elastic Compute Cloud User Guide for Linux Instances Setting Up Dynamic DNS

2. Configure the dynamic DNS update client. After you have a dynamic DNS service provider and a public DNS name registered with their service, point the DNS name to the IP address for your instance. Many providers (including noip.com) allow you to do this manually from your account page on their website, but many also support software update clients. If an update client is running on your EC2 instance, your dynamic DNS record is updated each time the IP address changes, as after a shutdown and restart. In this example, you install the `noip2` client, which works with the service provided by noip.com.

- a. Enable the Extra Packages for Enterprise Linux (EPEL) repository to gain access to the `noip2` client.

Note

Amazon Linux instances have the GPG keys and repository information for the EPEL repository installed by default; however, Red Hat and CentOS instances must first install the `epel-release` package before you can enable the EPEL repository. For more information and to download the latest version of this package, see <https://fedoraproject.org/wiki/EPEL>.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- b. Install the `noip` package.

```
[ec2-user ~]$ sudo yum install -y noip
```

- c. Create the `noip2` configuration file. Enter the login and password information when prompted and answer the subsequent questions to configure the client.

```
[ec2-user ~]$ sudo noip2 -C
```

3. Enable the `noip` service with the `chkconfig` command.

```
[ec2-user ~]$ sudo chkconfig noip on
```

You can verify that the service is enabled with the `chkconfig --list` command.

```
[ec2-user ~]$ chkconfig --list noip
noip          0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Here, **noip** is on in runlevels 2, 3, 4, and 5 (which is correct). Now the update client starts at every boot and updates the public DNS record to point to the IP address of the instance.

4. Start the `noip` service.

```
[ec2-user ~]$ sudo service noip start
Starting noip2: [ OK ]
```

This command starts the client, which reads the configuration file (`/etc/no-ip2.conf`) that you created earlier and updates the IP address for the public DNS name that you chose.

5. Verify that the update client has set the correct IP address for your dynamic DNS name. Allow a few minutes for the DNS records to update, and then try to connect to your instance using SSH with the public DNS name that you configured in this procedure.

Running Commands on Your Linux Instance at Launch

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and `cloud-init` directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances via the command line tools), or as base64-encoded text (for API calls).

If you are interested in more complex automation scenarios, consider using AWS CloudFormation and AWS OpsWorks. For more information, see the [AWS CloudFormation User Guide](#) and the [AWS OpsWorks User Guide](#).

For information about running commands on your Windows instance at launch, see [Executing User Data](#) and [Managing Windows Instance Configuration](#) in the *Amazon EC2 User Guide for Windows Instances*.

In the following examples, the commands from the [Installing a LAMP Web Server tutorial \(p. 33\)](#) are converted to a shell script and a set of `cloud-init` directives that executes when the instance launches. In each example, the following tasks are executed by the user data:

- The distribution software packages are updated.
- The necessary web server, `php`, and `mysql` packages are installed.
- The `httpd` service is started and turned on via `chkconfig`.
- The `www` group is added, and the `ec2-user` is added to that group.
- The appropriate ownership and file permissions are set for the web directory and the files contained within it.
- A simple web page is created to test the web server and `php` engine.

Contents

- [Prerequisites \(p. 334\)](#)
- [User Data and Shell Scripts \(p. 334\)](#)
- [User Data and cloud-init Directives \(p. 336\)](#)
- [API and CLI Overview \(p. 337\)](#)

Prerequisites

The following examples assume that your instance has a public DNS name that is reachable from the Internet. For more information, see [Step 1: Launch an Instance \(p. 27\)](#). You must also configure your security group to allow `SSH` (port 22), `HTTP` (port 80), and `HTTPS` (port 443) connections. For more information about these prerequisites, see [Setting Up with Amazon EC2 \(p. 19\)](#).

Also, these instructions are intended for use with Amazon Linux, and the commands and directives may not work for other Linux distributions. For more information about other distributions, such as their support for `cloud-init`, see their specific documentation.

User Data and Shell Scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch, and the `cloud-init` output log file (`/var/log/cloud-init-output.log`) captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended.

Important

User data scripts and `cloud-init` directives only run during the first boot cycle when an instance is launched.

User data shell scripts must start with the `#!` characters and the path to the interpreter you want to read the script (commonly `/bin/bash`). For a great introduction on shell scripting, see [the BASH Programming HOW-TO](#) at the Linux Documentation Project (tldp.org).

Scripts entered as user data are executed as the `root` user, so do not use the `sudo` command in the script. Remember that any files you create will be owned by `root`; if you need non-`root` users to have file access, you should modify the permissions accordingly in the script. Also, because the script is not run interactively, you cannot include commands that require user feedback (such as `yum update` without the `-y` flag).

Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

To pass a shell script to an instance with user data

1. Follow the procedure for launching an instance at [Launching Your Instance from an AMI \(p. 279\)](#), but when you get to [Step 6 \(p. 281\)](#), paste the user data script text into the **User data** field and then complete the launch procedure. For the example below, the script creates and configures our web server.

```
#!/bin/bash
yum update -y
yum install -y httpd24 php56 mysql55-server php56-mysqlnd
service httpd start
chkconfig httpd on
groupadd www
usermod -a -G www ec2-user
chown -R root:www /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} +
find /var/www -type f -exec chmod 0664 {} +
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

2. Allow enough time for the instance to launch and execute the commands in your script, and then check to see that your script has completed the tasks that you intended. For our example, in a web browser, enter the URL of the PHP test file the script created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page.

Tip

If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding Rules to a Security Group \(p. 495\)](#).

3. (Optional) If your script did not accomplish the tasks you were expecting it to, or if you just want to verify that your script completed without errors, examine the `cloud-init` output log file at `/var/log/cloud-init-output.log` and look for error messages in the output.

For additional debugging information, you can create a Mime multipart archive that includes a `cloud-init` data section with the following directive:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends command output from your script to `/var/log/cloud-init-output.log`. For more information on `cloud-init` data formats and creating Mime multi part archive, see [cloud-init Formats](#).

User Data and cloud-init Directives

The `cloud-init` package configures specific aspects of a new Amazon Linux instance when it is launched; most notably, it configures the `.ssh/authorized_keys` file for the `ec2-user` so you can log in with your own private key.

The `cloud-init` user directives can be passed to an instance at launch the same way that a script is passed, although the syntax is different. For more information about `cloud-init`, go to <http://cloudinit.readthedocs.org/en/latest/index.html>.

Important

User data scripts and `cloud-init` directives only run during the first boot cycle when an instance is launched.

The Amazon Linux version of `cloud-init` does not support all of the directives that are available in the base package, and some of the directives have been renamed (such as `repo_update` instead of `apt-upgrade`).

Adding these tasks at boot time adds to the amount of time it takes to boot an instance. You should allow a few minutes of extra time for the tasks to complete before you test that your user data directives have completed.

To pass cloud-init directives to an instance with user data

1. Follow the procedure for launching an instance at [Launching Your Instance from an AMI \(p. 279\)](#), but when you get to [Step 6 \(p. 281\)](#), paste your `cloud-init` directive text into the **User data** field and then complete the launch procedure. For the example below, the directives create and configure a web server.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd24
- php56
- mysql55
- server
- php56-mysqld

runcmd:
- service httpd start
- chkconfig httpd on
- groupadd www
- [ sh, -c, "usermod -a -G www ec2-user" ]
- [ sh, -c, "chown -R root:www /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, + ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, + ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. Allow enough time for the instance to launch and execute the directives in your user data, and then check to see that your directives have completed the tasks you intended. For our example, in a web browser, enter the URL of the PHP test file the directives created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page.

Tip

If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding Rules to a Security Group \(p. 495\)](#).

3. (Optional) If your directives did not accomplish the tasks you were expecting them to, or if you just want to verify that your directives completed without errors, examine the `cloud-init` output log file at `/var/log/cloud-init-output.log` and look for error messages in the output. For additional debugging information, you can add the following line to your directives:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends `runcmd` output to `/var/log/cloud-init-output.log`.

API and CLI Overview

You can pass user data to your instance during launch using one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- AWS CLI: Use the `--user-data` parameter with the `run-instances` command.
- AWS Tools for Windows PowerShell: Use the `-UserData` parameter with the `New-EC2Instance` command.
- Amazon EC2 Query API: Use the `UserData` parameter with the `RunInstances` command.

Instance Metadata and User Data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see [Instance Metadata Categories \(p. 344\)](#).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see [Dynamic Data Categories \(p. 348\)](#).

You can also access the *user data* that you supplied when launching your instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data

(such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

Contents

- [Retrieving Instance Metadata \(p. 338\)](#)
- [Configuring Instances with User Data \(p. 340\)](#)
- [Retrieving User Data \(p. 341\)](#)
- [Retrieving Dynamic Data \(p. 341\)](#)
- [Example: AMI Launch Index Value \(p. 342\)](#)
- [Instance Metadata Categories \(p. 344\)](#)
- [Instance Identity Documents \(p. 349\)](#)

Retrieving Instance Metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can use a tool such as cURL, or if your instance supports it, the GET command; for example:

```
$ curl http://169.254.169.254/latest/meta-data/
```

```
$ GET http://169.254.169.254/latest/meta-data/
```

You can also download the Instance Metadata Query tool, which allows you to query the instance metadata without having to type out the full URI or category names:

<http://aws.amazon.com/code/1825>

All metadata is returned as text (content type text/plain). A request for a specific metadata resource returns the appropriate value, or a 404 - Not Found HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a 404 - Not Found HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

Examples of Retrieving Instance Metadata

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

```
$ curl http://169.254.169.254/  
1.0  
2007-01-19  
2007-03-01  
2007-08-29  
2007-10-10
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

```
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
latest
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see [Instance Metadata Categories](#) (p. 344).

```
$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

These examples get the value of some of the metadata items from the preceding example.

```
$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-12345678
```

```
$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-fea54097
```

```
$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

This example shows the formats in which public key 0 is available.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/  
openssh-key
```

This example gets public key 0 (in the OpenSSH key format).

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key  
ssh-rsa MIICiTCcAFICCCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVMxCzAJBgNVBAGTAldBMRAdG9YDQYJQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRItEAYDVQQDEw1UZXR0Q21sYWVxHmZAd  
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAdG9YD  
VQYJQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z  
b2x1MRItEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb251QGFT  
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE  
Ibb3OhjZncvcQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJiI0zbnYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvJx79LjSTb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

This example shows the information available for a specific network interface (indicated by the MAC address) on a NAT instance in the EC2-Classic platform.

```
$ curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/02:29:96:8f:6a:2d/  
device-number  
local-hostname  
local-ipv4s  
mac  
owner-id  
public-hostname  
public-ipv4s
```

This example gets the subnet ID for an instance launched into a VPC.

```
$ curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

Configuring Instances with User Data

When you specify user data, note the following:

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded before being submitted to the API. The EC2 command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encoding, see <http://tools.ietf.org/html/rfc4648>.
- User data is executed only at launch. If you stop an instance, modify the user data, and start the instance, the new user data is not executed automatically.

To specify user data when you launch an instance

You can specify user data when you launch an instance. For more information, see [Launching an Instance \(p. 279\)](#), [cloud-init \(p. 145\)](#), and [Running Commands on Your Linux Instance at Launch \(p. 334\)](#).

Modify User Data for a Running Instance

You can modify user data for an instance that previously had user data assigned and is currently running. The new user data is available on your instance after you reboot it.

To modify the user data for an Amazon EBS-backed instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **View/Change User Data**. Note that you can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then choose **Save**.

Retrieving User Data

To retrieve user data, use the following URI:

```
http://169.254.169.254/latest/user-data
```

Requests for user data returns the data as it is (content type application/x-octetstream).

This shows an example of returning comma-separated user data.

```
$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

This shows an example of returning line-separated user data.

```
$ curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

[instance-1]
reboot-on-error: yes
```

Retrieving Dynamic Data

To retrieve dynamic data from within a running instance, use the following URI:

```
http://169.254.169.254/latest/dynamic/
```

This example shows how to retrieve the high-level instance identity categories:

```
$ curl http://169.254.169.254/latest/dynamic/instance-identity/  
pkcs7  
signature  
document
```

For more information about dynamic data and examples of how to retrieve it, see [Instance Identity Documents](#) (p. 349).

Example: AMI Launch Index Value

This example demonstrates how you can use both user data and instance metadata to configure your instances.

Alice wants to launch four instances of her favorite database AMI, with the first acting as master and the remaining three acting as replicas. When she launches them, she wants to add user data about the replication strategy for each replicant. She is aware that this data will be available to all four instances, so she needs to structure the user data in a way that allows each instance to recognize which parts are applicable to it. She can do this using the `ami-launch-index` instance metadata value, which will be unique for each instance.

Here is the user data that Alice has constructed:

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

The `replicate-every=1min` data defines the first replicant's configuration, `replicate-every=5min` defines the second replicant's configuration, and so on. Alice decides to provide this data as an ASCII string with a pipe symbol (`|`) delimiting the data for the separate instances.

Alice launches four instances using the `run-instances` command, specifying the user data:

```
aws ec2 run-instances --image-id ami-12345678 --count 4 --instance-type  
t2.micro --user-data "replicate-every=1min | replicate-every=5min |  
replicate-every=10min"
```

After they're launched, all instances have a copy of the user data and the common metadata shown here:

- AMI id: `ami-12345678`
- Reservation ID: `r-1234567890abcabc0`
- Public keys: `none`
- Security group name: `default`
- Instance type: `t2.micro`

However, each instance has certain unique metadata.

Instance 1

Metadata	Value
instance-id	i-1234567890abcdef0

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

Metadata	Value
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Instance 2

Metadata	Value
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

Instance 3

Metadata	Value
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

Instance 4

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice can use the `ami-launch-index` value to determine which portion of the user data is applicable to a particular instance.

1. She connects to one of the instances, and retrieves the `ami-launch-index` for that instance to ensure it is one of the replicants:

```
$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. She saves the `ami-launch-index` as a variable:

```
$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-launch-index`
```

3. She saves the user data as a variable:

```
$ user_data=`curl http://169.254.169.254/latest/user-data/`
```

4. Finally, Alice uses the `cut` command to extract the portion of the user data that is applicable to that instance:

```
$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

Instance Metadata Categories

The following table lists the categories of instance metadata.

Data	Description	Version Introduced
<code>ami-id</code>	The AMI ID used to launch the instance.	1.0
<code>ami-launch-index</code>	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	1.0
<code>ami-manifest-path</code>	The path to the AMI manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is unknown.	1.0
<code>ancestor-ami-ids</code>	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an <code>ancestor-amis</code> key.	2007-10-10
<code>block-device-mapping/ami</code>	The virtual device that contains the root/boot file system.	2007-12-15
<code>block-device-mapping/ebs</code> <i>N</i>	The virtual devices associated with Amazon EBS volumes, if any are present. Amazon EBS volumes	2007-12-15

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

Data	Description	Version Introduced
	are only available in metadata if they were present at launch time or when the instance was last started. The <i>N</i> indicates the index of the Amazon EBS volume (such as <code>ebs1</code> or <code>ebs2</code>).	
<code>block-device-mapping/ephemeral</code> <i>N</i>	The virtual devices associated with ephemeral devices, if any are present. The <i>N</i> indicates the index of the ephemeral volume.	2007-12-15
<code>block-device-mapping/root</code>	The virtual devices or partitions associated with the root devices, or partitions on the virtual device, where the root (<code>/</code> or <code>C:</code>) file system is associated with the given instance.	2007-12-15
<code>block-device-mapping/swap</code>	The virtual devices associated with <code>swap</code> . Not always present.	2007-12-15
<code>hostname</code>	The private hostname of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	1.0
<code>iam/info</code>	If there is an IAM role associated with the instance at launch, contains information about the last time the instance profile was updated, including the instance's <code>LastUpdatedDate</code> , <code>InstanceProfileArn</code> , and <code>InstanceProfileId</code> . Otherwise, not present.	2012-01-12
<code>iam/security-credentials/role-name</code>	If there is an IAM role associated with the instance at launch, <i>role-name</i> is the name of the role, and <i>role-name</i> contains the temporary security credentials associated with the role (for more information, see Retrieving Security Credentials from Instance Metadata (p. 537)). Otherwise, not present.	2012-01-12
<code>instance-action</code>	Notifies the instance that it should reboot in preparation for bundling. Valid values: <code>none</code> <code>shutdown</code> <code>bundle-pending</code> .	2008-09-01
<code>instance-id</code>	The ID of this instance.	1.0
<code>instance-type</code>	The type of instance. For more information, see Instance Types (p. 156).	2007-08-29

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

Data	Description	Version Introduced
<code>kernel-id</code>	The ID of the kernel launched with this instance, if applicable.	2008-02-01
<code>local-hostname</code>	The private DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	2007-01-19
<code>local-ipv4</code>	The private IP address of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	1.0
<code>mac</code>	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	2011-01-01
<code>network/interfaces/macs/ mac/device-number</code>	The unique device number associated with that interface. The device number corresponds to the device name; for example, a <code>device-number</code> of 2 is for the <code>eth2</code> device. This category corresponds to the <code>DeviceIndex</code> and <code>device-index</code> fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
<code>network/interfaces/macs/ mac/ipv4-associations/ public-ip</code>	The private IPv4 addresses that are associated with each <code>public-ip</code> address and assigned to that interface.	2011-01-01
<code>network/interfaces/macs/ mac/local-hostname</code>	The interface's local hostname.	2011-01-01
<code>network/interfaces/macs/ mac/local-ipv4s</code>	The private IP addresses associated with the interface.	2011-01-01
<code>network/interfaces/macs/ mac/mac</code>	The instance's MAC address.	2011-01-01
<code>network/interfaces/macs/ mac/owner-id</code>	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

Data	Description	Version Introduced
<code>network/interfaces/macs/mac/public-hostname</code>	The interface's public DNS. If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/public-ipv4s</code>	The Elastic IP addresses associated with the interface. There may be multiple IP addresses on an instance.	2011-01-01
<code>network/interfaces/macs/mac/security-groups</code>	Security groups to which the network interface belongs. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/security-group-ids</code>	IDs of the security groups to which the network interface belongs. Returned only for instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see Security Groups for Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/subnet-id</code>	The ID of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/subnet-ipv4-cidr-block</code>	The CIDR block of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/vpc-id</code>	The ID of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/vpc-ipv4-cidr-block</code>	The CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>placement/availability-zone</code>	The Availability Zone in which the instance launched.	2008-02-01
<code>product-codes</code>	Product codes associated with the instance, if any.	2007-03-01
<code>public-hostname</code>	The instance's public DNS. If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2007-01-19
<code>public-ipv4</code>	The public IP address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Metadata and User Data**

Data	Description	Version Introduced
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	1.0
ramdisk-id	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
reservation-id	The ID of the reservation.	1.0
security-groups	The names of the security groups applied to the instance. After launch, you can only change the security groups of instances running in a VPC. Such changes are reflected here and in <code>network/interfaces/macs/<i>mac</i>/security-groups</code> .	1.0
services/domain	The domain for AWS resources for the region; for example, <code>amazonaws.com</code> for <code>us-east-1</code> .	2014-02-25
services/partition	The partition that the resource is in. For standard AWS regions, the partition is <code>aws</code> . If you have resources in other partitions, the partition is <code>aws-<i>partitionname</i></code> . For example, the partition for resources in the China (Beijing) region is <code>aws-cn</code> .	2015-10-20
spot/termination-time	The approximate time, in UTC, that the operating system for your Spot instance will receive the shutdown signal. This item is present and contains a time value (for example, <code>2015-01-05T18:02:00Z</code>) only if the Spot instance has been marked for termination by Amazon EC2. The <code>termination-time</code> item is not set to a time if you terminated the Spot instance yourself.	2014-11-05

Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: <code>enabled</code> <code>disabled</code>	2009-04-04
instance-identity/document	JSON containing instance attributes, such as <code>instance-id</code> , private IP address, etc. See Instance Identity Documents (p. 349) .	2009-04-04

Data	Description	Version introduced
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance Identity Documents (p. 349).	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity. See Instance Identity Documents (p. 349).	2009-04-04

Instance Identity Documents

An instance identity document is a JSON file that describes an instance. The instance identity document is accompanied by a signature and a PKCS7 signature which can be used to verify the accuracy, origin, and authenticity of the information provided in the document. For example, you may have downloaded free software with paid updates.

The instance identity document is generated when the instance is launched, and exposed to the instance through [instance metadata](#) (p. 337). It validates the attributes of the instances, such as the subscribed software, instance size, instance type, operating system, and AMI.

Important

Due to the dynamic nature of instance identity documents and signatures, we recommend retrieving the instance identity document and signature regularly.

Obtaining the Instance Identity Document and Signatures

To retrieve the instance identity document, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/document
```

```
{
  "devpayProductCodes" : null,
  "availabilityZone" : "us-east-1d",
  "privateIp" : "10.158.112.84",
  "version" : "2010-08-31",
  "region" : "us-east-1",
  "instanceId" : "i-1234567890abcdef0",
  "billingProducts" : null,
  "instanceType" : "t1.micro",
  "accountId" : "123456789012",
  "pendingTime" : "2015-11-19T16:32:11Z",
  "imageId" : "ami-5fb8c835",
  "kernelId" : "aki-919dcaf8",
  "ramdiskId" : null,
  "architecture" : "x86_64"
}
```

To retrieve the instance identity signature, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/signature
```

```
dExamplesjNQhhJan7pORLpLSr7lJEF4V2DhKGLyoYVBoUYrY9njyBCmhEayaGrhtS/AWY+LPx
lVSQURF5n0gwPNCuO6ICT0fNrm5IH7w9ydyexamplejJw8XvWPxbuRkcN0TAA1p4RtCAqm4ms
```

```
x2oALjWSCBExample=
```

To retrieve the PKCS7 signature, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
```

```
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMx CzAJBgNVBAGTAldBMRAdG9YDVQqHEwdTZWF0dGx1MQ8wDQYDVQqKEwZBbWF6
b24x FDASBgNVBASTC0lBTSBDb25zb2xlMRlWEAYDVQqDEw1UZXRNOQ2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGfTYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDIOMjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMx CzAJBgNVBAGTAldBMRAdG9YD
VQqHEwdTZWF0dGx1MQ8wDQYDVQqKEwZBbWF6b24x FDASBgNVBASTC0lBTSBDb25z
b2xlMRlWEAYDVQqDEw1UZXRNOQ2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGfT
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGvIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gPE
Ibb3OhjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYuntneD9+h8Mg9q6q+auNKyExzyLwaxLAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJiLJ00zbnNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE
```

Example: Verifying the PKCS7 Signature

You can use the PKCS7 signature to verify your instance by validating it against the AWS public certificate for the region.

The AWS public certificate for the US East (N. Virginia), US West (Oregon), US West (N. California), EU (Ireland), EU (Frankfurt), Asia Pacific (Tokyo), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Mumbai), and South America (São Paulo) regions is as follows:

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAg0CCQCWukjZ5V4aZzAJBgqhkiG9w0BAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQqIEExBXYXNoaW5ndG9uIFN0YXRlMRAdG9YDVQqHEwdTZWF0dGx1MSAwHgYD
VQqKExdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQzAeFw0xMjA0MDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQqIEExBXYXNoaW5ndG9u
IFN0YXRlMRAdG9YDVQqHEwdTZWF0dGx1MSAwHgYDVQqKExdBbWF6b24gV2ViIFNl
cnZpY2VzIEExMQzCCAbcwggESBgqhkiG9w0BAQDMFwIIBHwKBGQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7Eg1K9ITHJLP+X8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAII1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0CaO/buyCU1CiYQk40KNHCChfNiZbd1x1E9rpUp7bnF
lRa2vIntMX3carVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKml1qx41lHW
MXrs3IqIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGBYqGSM44BAMDlwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

The AWS public certificate for the AWS GovCloud (US) region is as follows:

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAg0CCQCWukjZ5V4aZzAJBgqhkiG9w0BAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQqIEExBXYXNoaW5ndG9uIFN0YXRlMRAdG9YDVQqHEwdTZWF0dGx1MSAwHgYD
VQqKExdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQzAeFw0xMjA0MDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQqIEExBXYXNoaW5ndG9u
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Instance Metadata and User Data

```
IFN0YXRlMRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFNl
cnZpY2VzIEExMQZCCAbcwggEsBgcqhkJ0OQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAIIj
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCG19fueQ2s6IL0CaO/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQOGx5ho8WqD+aTeb+k2tn92BBPqeZqpWRa5P/+jrdKml1qx411HW
MXrs3Igb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGBYqGSM44BAMDLwAwLAIUWXBlk40xTwsW
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

For more information about AWS GovCloud (US), see the [AWS GovCloud \(US\) User Guide](#).

For other regions, contact [AWS Support](#) to get the AWS public certificate.

To verify the PKCS7 signature

1. From your Amazon Linux instance, create a temporary file for the PKCS7 signature:

```
PKCS7=$(mktemp)
```

2. Populate the file with the -----BEGIN PKCS7----- header, then append the contents of the PKCS7 signature from the instance metadata, a new line, and the -----END PKCS7----- footer:

```
echo "-----BEGIN PKCS7-----" > $PKCS7
```

```
curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >>
$PKCS7
```

```
echo "" >> $PKCS7
```

```
echo "-----END PKCS7-----" >> $PKCS7
```

3. Create a temporary file for the instance identity document, and populate it with the contents of the document from your instance metadata:

```
DOCUMENT=$(mktemp)
```

```
curl -s http://169.254.169.254/latest/dynamic/instance-identity/document >
$DOCUMENT
```

4. Open a text editor and create a file called `AWSpubkey`. Copy and paste the contents of the AWS public certificate above to the file and save it.
5. Use the OpenSSL tools to verify the signature as follows:

```
openssl smime -verify -in $PKCS7 -inform PEM -content $DOCUMENT -certfile
AWSpubkey -noverify > /dev/null
Verification successful
```

Remotely Manage Your Instances

You can use Amazon Elastic Compute Cloud (Amazon EC2) Run Command to remotely manage the configuration of your Amazon EC2 instances, virtual machines (VMs), or servers in your on-premises environment or in an environment provided by other cloud providers. Using scripts, commands, or the Amazon EC2 console, Run Command enables you to perform common administrative tasks and manage configuration changes across a fleet without having to log on locally to each instance, server, or VM.

Note

For information about Run Command for Windows, see [Amazon EC2 Run Command](#) in the *Amazon EC2 User Guide for Windows Instances*.

How To Get Started

The following table includes information to help you get started with Run Command.

Topic	Details
Tutorial: Execute a Command Using Run Command (p. 64)	The tutorial shows you how to quickly send a command using Run Command with AWS Command Line Interface (AWS CLI).
Run Command Components and Concepts (p. 352)	Learn about Run Command features and concepts.
Run Command Prerequisites (p. 355)	Verify that your instances meet the minimum requirements for Run Command.
Setting Up Managed Instances (p. 371)	Register on-premises servers, VMs, or servers hosted by other cloud providers with AWS. After a server or VM has been registered, it is called a <i>managed instance</i> .
Run Command Walkthroughs (p. 387)	Learn from these detailed walkthroughs how to execute commands using Run Command from either the Amazon EC2 console or the AWS CLI.

Related Content

- [Amazon EC2 Simple Systems Manager API Reference](#)
- [SSM AWS Tools for Windows PowerShell Reference](#)
- [SSM AWS CLI Reference](#)
- [AWS SDKs](#)

Amazon EC2 Run Command Components and Concepts

As you get started with Amazon EC2 Run Command, you'll benefit from understanding the components and concepts of this feature.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Components and Concepts**

Component/Concept	Details
Amazon EC2 Simple Systems Manager (SSM)	Run Command is a component of SSM. Run Command uses the SSM API. For more information, see Amazon EC2 Simple Systems Manager API Reference .
Managed Instances	A managed instance is an on-premises server or virtual machine (VM) hosted in your data center, or a VM hosted by other cloud providers, that is configured for Run Command. For more information about managed instances, see Setting Up Run Command On Managed Instances (p. 371) .
Commands	You can configure Amazon EC2 instances and managed instances by sending commands from your local machine. You don't need to log on locally to configure your machines. You can send commands using one of the following: the Command History page of the Amazon EC2 console , AWS Tools for Windows PowerShell, the AWS Command Line Interface (AWS CLI), the SSM API, or Amazon SDKs. For more information, see SSM AWS Tools for Windows PowerShell Reference , SSM AWS CLI Reference , and the AWS SDKs .
SSM Documents	An SSM document defines the plugins to run and the parameters to use when a command executes on a machine. When you execute a command, you specify the SSM document that Run Command uses. Run Command includes pre-defined documents that enable you to quickly perform common tasks on a machine. You can also create your own SSM documents. The first time you execute a command from a new SSM document, the system stores the document with your AWS account. For more information, see Creating SSM Documents (p. 378) .
SSM Agent	The SSM agent is AWS software that you install on your EC2 instances and managed instances. The agent processes Run Command requests and configures your machine as specified in the request. For more information, see Installing the SSM Agent (p. 357) .
IAM Roles and Polices	AWS user accounts and instances must be configured with AWS Identity and Access Management (IAM) roles and trust policies that enable them to communicate with the SSM API. For more information, see Delegating Access to Amazon EC2 Run Command (p. 365) .

How It Works

After you verify prerequisites for your EC2 instances and managed instances, you send a command from your local machine. The SSM service verifies the integrity of the command and any parameters and then forwards the request to the Amazon EC2 messaging service. The SSM agent running each instance (or EC2Config service on EC2 Windows instances) communicates with the EC2 messaging service to retrieve commands. The agent processes the command, configures the instance as specified, and logs the output and results.

Note

The agent attempts to execute each command once. You can send multiple commands at the same time.

The system manages the queuing, execution, cancellation, and reporting of each command. However, the order of command execution is not guaranteed. By default, Run Command uses throttle limits to ensure that no more than 60 commands are issued per minute per instance. If an instance is not running or is unresponsive when you execute a command, the system queues the command and attempts to run it when the instance is responsive. By default, the system will queue a command and attempt to run it for up to 31 days after request. For more information about command status, see [Monitoring Commands \(p. 394\)](#).

Run Command reports the status and results of each command for each instance, server, or VM. Run Command stores the command history for 30 days. The information is also stored in AWS CloudTrail and remains available until you delete the data. For more information, see [Auditing API Calls in the Amazon EC2 Simple Systems Manager API Reference](#).

More about SSM Documents

After you configure Run Command prerequisites, you determine what type of configuration change you want to make on your instance and which SSM document will enable you to make that change. Run Command includes pre-defined SSM documents that enable you to quickly execute commands on instances. The commands available to you depend on the permissions your administrator specified for you. Any command that begins with AWS-* uses a pre-defined SSM document provided by AWS. A developer or administrator can create additional documents and provision these for you based on your permissions. For more information, see [Creating SSM Documents \(p. 378\)](#).

Important

Only trusted administrators should be allowed to use AWS pre-configured documents. The commands or scripts specified in SSM documents run with administrative privilege on your instances because the Amazon SSM agent runs as root. If a user has permission to execute any of the pre-defined SSM documents (any document that begins with AWS-*), then that user also has administrator access to the instance. For all other users, you should create restrictive documents and share them with specific users. For more information about restricting access to Run Command, see [Delegating Access to Amazon EC2 Run Command \(p. 365\)](#).

Run Command includes the following pre-configured SSM documents.

Amazon Pre-configured SSM documents for Linux

Name	Description
AWS-RunShellScript	Run shell scripts
AWS-UpdateSSMAgent	Update the Amazon SSM agent

You can select a document from a list in the [Amazon EC2 console](#) or use a `list documents` command to view a list a commands available to you in either the AWS CLI or AWS Tools for Windows PowerShell.

Amazon EC2 Run Command Prerequisites

Amazon EC2 Run Command has the following limitations and prerequisites. For information about Windows prerequisites, see [Amazon EC2 Run Command Prerequisites](#) in the *Amazon EC2 User Guide for Windows Instances*.

Limitations

Run Command is [only available in these regions](#).

Note

For managed instances, we recommend that you choose the region closest to your data center or computing environment.

Prerequisites

Requirement	Details	For More Information
Supported Operating System	<p>Instances must be running a supported version of Linux.</p> <p>64-Bit and 32-Bit Systems</p> <ul style="list-style-type: none">• Amazon Linux 2014.09, 2014.03 or later• Ubuntu Server 14.04 LTS or 12.04 LTS• Red Hat Enterprise Linux (RHEL) 6.5 or later• CentOS 6.3 or later <p>64-Bit Systems Only</p> <ul style="list-style-type: none">• Amazon Linux 2015.09, 2015.03 or later• Red Hat Enterprise Linux (RHEL) 7.x or later• CentOS 7.1 or later	Finding a Linux AMI (p. 75)
SSM agent	<p>The SSM agent processes Run Command requests and configures your machine as specified in the request. You must download and install the SSM agent to your EC2 instance, on-premises servers or VMs, or VMs hosted by other cloud providers.</p> <p>Important Because of a known issue, users in the following regions <i>must download</i> the latest version of the SSM agent: Asia Pacific (Seoul), ap-northeast-2 and China</p>	Installing the SSM Agent (p. 357)

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Prerequisites**

Requirement	Details	For More Information
	<p>(Beijing), cn-north-1. This is a one time installation. After installing the latest version, you can update the agent using Run Command.</p> <p>The source code for the SSM agent is available on GitHub so that you can adapt the agent to meet your needs. We encourage you to submit pull requests for changes that you would like to have included. However, Amazon Web Services does not currently provide support for running modified copies of this software.</p>	
Access to Run Command	<p>Before you can execute commands using Run Command, you must configure an AWS Identity and Access Management (IAM) EC2 instance role for instances that will process commands. You must also configure a separate user role for users executing commands. Both roles require permission policies that enable them to communicate with the SSM API.</p> <p>Note For managed instances, you must also create an IAM service role that enables your on-premises server or VM to communicate with the SSM service. For more information, see Create a Managed Instance Service Role (p. 371).</p>	Delegating Access to Amazon EC2 Run Command (p. 365)
Internet Access	Verify that your EC2 instances have outbound Internet access. Inbound Internet access is not required.	Internet Gateways

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Installing the SSM Agent**

Requirement	Details	For More Information
Amazon S3 Bucket (Optional)	You can store command output in an Amazon Simple Storage Service (S3) bucket. Command output in the Amazon EC2 console is truncated after 2500 characters. Additionally, you might want to create an Amazon S3 key prefix (a subfolder) to help you organize Run Command output.	Create a Bucket

Note

SSM communicates with the SSM Agent on your instance by using the EC2 Messaging service. If you monitor traffic, you will see your instances communicating with `ec2messages.*` endpoints.

After you configure prerequisites, see [Executing a Command Using Amazon EC2 Run Command \(p. 374\)](#) for step-by-step procedures using the Amazon EC2 console. To view AWS Command Line Interface (AWS CLI) and AWS Tools for Windows PowerShell examples, see [Amazon EC2 Run Command Walkthroughs \(p. 387\)](#).

Related Topics

- [Amazon EC2 Simple Systems Manager API Reference](#)
- [Amazon EC2 Simple Systems Manager](#) section of the *AWS Command Line Interface Reference*

Installing the SSM Agent

The Amazon EC2 Simple Systems Manager (SSM) agent processes Run Command requests and configures your machine as specified in the request. Use the following procedures to install, configure, or uninstall the SSM agent.

Note

The source code for the SSM agent is available on [GitHub](#) so that you can adapt the agent to meet your needs. We encourage you to submit [pull requests](#) for changes that you would like to have included. However, Amazon Web Services does not currently provide support for running modified copies of this software.

The commands in this section require you to replace `region` with a [region where SSM is available](#). To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance. For managed instances, there is no cross-region data transfer cost, but we recommend that you specify the region closest to your data center or computing environment.

Contents

- [Install the SSM Agent on EC2 Instances at Start-Up \(p. 358\)](#)
- [Manually Install the SSM Agent on EC2 Instances \(p. 359\)](#)
- [Configure the SSM Agent to Use a Proxy \(p. 363\)](#)
- [Uninstall the SSM Agent \(p. 364\)](#)

Install the SSM Agent on EC2 Instances at Start-Up

You can install the SSM agent when you launch an instance for the first time by using EC2 User Data. On the **Configure Instance Details** page of the launch wizard, expand **Advanced Details** and then copy and paste one of the following scripts into the **User data** field. For example:

▼ **Advanced Details**

User data ⓘ As text As file Input is already base64 encoded

```
#!/bin/bash
yum -y update
curl https://amazon-ssm-us-east-1.s3.amazonaws.com/
yum install -y amazon-ssm-agent.rpm
```

Amazon Linux, RHEL, and CentOS 64-bit

```
#!/bin/bash
cd /tmp
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/amazon-ssm-agent.rpm -o amazon-ssm-agent.rpm
yum install -y amazon-ssm-agent.rpm
```

Amazon Linux, RHEL, and CentOS 32-bit

```
#!/bin/bash
cd /tmp
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_386/amazon-ssm-agent.rpm -o amazon-ssm-agent.rpm
yum install -y amazon-ssm-agent.rpm
```

Ubuntu Server 64-bit

```
#!/bin/bash
cd /tmp
curl https://amazon-ssm-region.s3.amazonaws.com/latest/debian_amd64/amazon-ssm-agent.deb -o amazon-ssm-agent.deb
dpkg -i amazon-ssm-agent.deb
```

Ubuntu Server 32-bit

```
#!/bin/bash
cd /tmp
curl https://amazon-ssm-region.s3.amazonaws.com/latest/debian_386/amazon-ssm-agent.deb -o amazon-ssm-agent.deb
dpkg -i amazon-ssm-agent.deb
```

Save your changes, and complete the wizard. When the instance launches, the system copies the SSM agent to the instance and starts it. When the instance is online, you can configure it using Run Command. For more information, see [Executing a Command Using Amazon EC2 Run Command](#) (p. 374).

Manually Install the SSM Agent on EC2 Instances

Use one of the following scripts to install the SSM agent on Amazon Linux, Ubuntu, CentOS, or Red Hat Enterprise Linux. In these scripts, replace *region* with a [region where SSM is available](#).

- [Install the SSM Agent on Amazon Linux \(p. 359\)](#)
- [Install the SSM Agent on Ubuntu \(p. 360\)](#)
- [Install the SSM Agent on Red Hat Enterprise Linux \(p. 360\)](#)
- [Install the SSM Agent on CentOS \(p. 362\)](#)

Install the SSM Agent on Amazon Linux

Connect to your Amazon Linux instance and perform the following steps to install the SSM agent. Perform these steps on each instance that will execute commands using Run Command.

To install the SSM agent on Amazon Linux

1. Create a temporary directory on the instance.

```
mkdir /tmp/ssm
```

2. Use one of the following commands to download the SSM installer to the temporary directory. Replace *region* with one of the AWS Regions where SSM is available. To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance.

64-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

32-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_386/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

3. Run the SSM installer.

```
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
```

4. Run the following command to determine if the SSM agent is running. The command should return "amazon-ssm-agent is running."

```
sudo status amazon-ssm-agent
```

5. Execute the following commands if the previous command returns, "amazon-ssm-agent is stopped."

- a. Start the service.

```
sudo start amazon-ssm-agent
```

- b. Check the status of the agent.

```
sudo status amazon-ssm-agent
```

Install the SSM Agent on Ubuntu

Connect to your Ubuntu instance and perform the following steps to install the SSM agent. Perform these steps on each instance that will execute commands using Run Command.

To install the SSM agent on Ubuntu

1. Create a temporary directory on the instance.

```
mkdir /tmp/ssm
```

2. Use one of the following commands to download the SSM installer to the temporary directory. Replace *region* with one of the AWS Regions where SSM is available. To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance.

64-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/debian_amd64/  
amazon-ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb
```

32-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/debian_386/amazon-  
ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb
```

3. Run the SSM installer.

```
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb
```

4. Run the following command to determine if the SSM agent is running. The command should return "amazon-ssm-agent is running."

```
sudo status amazon-ssm-agent
```

5. Execute the following commands if the previous command returned, "amazon-ssm-agent is stopped."

- a. Start the service.

```
sudo start amazon-ssm-agent
```

- b. Check the status of the agent.

```
sudo status amazon-ssm-agent
```

Install the SSM Agent on Red Hat Enterprise Linux

Connect to your Red Hat Enterprise Linux (RHEL) instance and perform the following steps to install the SSM agent. Perform these steps on each instance that will execute commands using Run Command.

To install the SSM agent on Red Hat Enterprise Linux

1. Create a temporary directory on the instance.

Amazon Elastic Compute Cloud User Guide for Linux Instances Installing the SSM Agent

```
mkdir /tmp/ssm
```

2. Use one of the following commands to download the SSM installer to the temporary directory. Replace *region* with one of the AWS Regions where SSM is available. To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance.

64-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

32-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_386/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

3. Run the SSM installer.

```
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
```

4. Run one of the following commands to determine if the SSM agent is running. The command should return "amazon-ssm-agent is running."

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x

```
sudo status amazon-ssm-agent
```

5. Execute the following commands if the previous command returned "amazon-ssm-agent is stopped."
 - a. Start the service.

RHEL 7.x

```
sudo systemctl start amazon-ssm-agent
```

RHEL 6.x

```
sudo start amazon-ssm-agent
```

- b. Check the status of the agent.

RHEL 7.x

```
sudo systemctl status amazon-ssm-agent
```

RHEL 6.x


```
sudo status amazon-ssm-agent
```

Install the SSM Agent on CentOS

Connect to your CentOS instance and perform the following steps to install the SSM agent. Perform these steps on each instance that will execute commands using Run Command.

To install the SSM agent on CentOS

1. Create a temporary directory on the instance.

```
mkdir /tmp/ssm
```

2. Use one of the following commands to download the SSM installer to the temporary directory. Replace *region* with one of the AWS Regions where SSM is available. To avoid cross-region data transfer costs for the download, specify the region of your EC2 instance.

64-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

32-Bit

```
curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_386/amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm
```

3. Run the SSM installer.

```
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm
```

4. Run one of the following commands to determine if the SSM agent is running. The command should return "amazon-ssm-agent is running."

CentOS 7.x

```
sudo systemctl status amazon-ssm-agent
```

CentOS 6.x

```
status amazon-ssm-agent
```

5. Execute the following commands if the previous command returned "amazon-ssm-agent is stopped."

- a. Start the service.

CentOS 7.x

```
sudo systemctl start amazon-ssm-agent
```

CentOS 6.x

```
sudo start amazon-ssm-agent
```

- b. Check the status of the agent.

CentOS 7.x

```
sudo systemctl status amazon-ssm-agent
```

CentOS 6.x

```
sudo status amazon-ssm-agent
```

Configure the SSM Agent to Use a Proxy

You can configure the SSM agent to communicate through an HTTP proxy by adding the `http_proxy` and `no_proxy` settings to the SSM agent configuration file. This section includes procedures for *upstart* and *systemd* environments.

To configure the SSM agent to use a proxy (Upstart)

1. Connect to the instance where you installed the SSM agent.
2. Open the `amazon-ssm-agent.conf` file in an editor such as VIM. By default, the file is located here:

```
/etc/init/amazon-ssm-agent.conf
```

3. Add the `http_proxy` setting to the file in the following format:

```
env http_proxy=http://hostname:port
```

4. Add the `no_proxy` setting to the file in the following format. You must specify the IP address listed here. It is the instance metadata endpoint for SSM and without this IP address calls to SSM fail:

```
env no_proxy=169.254.169.254
```

5. Save your changes and close the editor.
6. Restart the SSM agent using the following command:

```
/sbin/restart amazon-ssm-agent
```

The following Upstart example includes the `http_proxy` and `no_proxy` settings in the `amazon-ssm-agent.conf` file:

```
description "Amazon SSM Agent"  
author "Amazon.com"  
  
start on (runlevel [345] and started network)  
stop on (runlevel [!345] or stopping network)  
  
respawn  
  
env http_proxy=http://i-1234567890abcdef0:443  
env no_proxy=169.254.169.254
```

```
chdir /usr/bin/  
exec ./amazon-ssm-agent
```

To configure the SSM agent to use a proxy (systemd)

1. Connect to the instance where you installed the SSM agent.
2. Open the `amazon-ssm-agent.service` file in an editor such as VIM. By default, the file is located here:

```
/etc/systemd/system/amazon-ssm-agent.service
```

3. Add the `http_proxy` setting to the file in the following format:

```
Environment="HTTP_PROXY=http://hostname:port"
```

4. Add the `no_proxy` setting to the file in the following format. You must specify the IP address listed here. It is the instance metadata endpoint for SSM and without this IP address calls to SSM fail:

```
Environment="no_proxy=169.254.169.254"
```

5. Save your changes and close the editor.
6. Restart the SSM agent using the following commands:

```
systemctl stop amazon-ssm-agent  
systemctl daemon-reload
```

The following systemd example includes the `http_proxy` and `no_proxy` settings in the `amazon-ssm-agent.service` file:

```
Type=simple  
Environment="HTTP_PROXY=http://i-1234567890abcdef0:443"  
Environment="no_proxy=169.254.169.254"  
WorkingDirectory=/opt/amazon/ssm/  
ExecStart=/usr/bin/amazon-ssm-agent  
KillMode=process  
Restart=on-failure  
RestartSec=15min
```

Uninstall the SSM Agent

Use the following commands to uninstall the SSM agent.

To uninstall the SSM agent on Amazon Linux, RHEL, or Cent OS

```
sudo yum erase amazon-ssm-agent -y
```

To uninstall the SSM agent on Ubuntu

```
sudo dpkg -r amazon-ssm-agent
```

Delegating Access to Amazon EC2 Run Command

Amazon EC2 Run Command requires an IAM role for EC2 instances that will process commands and a separate role for users executing commands. Both roles require permission policies that enable them to communicate with the [SSM API](#). You can choose to use SSM managed policies or you can create your own roles and specify permissions as described in this section.

If you are configuring managed instances (on-premises servers or VMs that you want to configure using Run Command) you must also configure an IAM service role. For more information, see [Create a Managed Instance Service Role](#) (p. 371).

This section includes the following information.

- [Use SSM Managed Policies](#) (p. 365)
- [Configure Your Own Roles and Polices](#) (p. 365)
- [Create EC2 Instances that Use the EC2 Instance Role](#) (p. 370)

After you configure roles and policies, we recommend that you further delegate access by creating commands that perform specific operations and assign access to specific users. For low-level actions that anyone in your group can perform, you can create low-level commands and make them public. For more information, see [Creating SSM Documents](#) (p. 378).

Use SSM Managed Policies

IAM managed policies for SSM can help you quickly configure access and permissions for Run Command users and instances. You can find these policies in the **Policies** page of the IAM console by searching for SSM, as shown in the following screen shot.



The managed policies perform the following functions:

- **AmazonEC2RoleforSSM (instance trust policy):** Enables an instance to communicate with the Run Command API.
- **AmazonSSMFullAccess (user trust policy):** Grants the user access to the Run Command API and SSM JSON documents. Assign this policy to administrators and trusted power users.
- **AmazonSSMReadOnlyAccess (user trust policy):** Grants the user access to read-only API actions, such as Get and List.

For information about how to configure these policies, see [Managed Policies and Inline Policies](#).

Configure Your Own Roles and Polices

If you choose not to use SSM managed policies, then use the following procedures to create and configure an SSM EC2 instance role and an SSM user account.

Important

If you want to use an existing EC2 instance role and user account, you must attach the policies shown in this section to the role and the user account. You must also verify that `ec2.amazonaws.com` is listed in the trust policy for the EC2 instance role. For more information, see [Verify the Trust Policy](#) (p. 369).

Topics

- [Create the IAM Policy for EC2 Instances \(p. 366\)](#)
- [Create the IAM User Policy \(p. 367\)](#)
- [Create a Restrictive IAM User Policy \(p. 368\)](#)
- [Create the EC2 Instance Role \(p. 369\)](#)
- [Verify the Trust Policy \(p. 369\)](#)
- [Create the User Account \(p. 370\)](#)

Create the IAM Policy for EC2 Instances

The following IAM policy enables EC2 instances to communicate with the Run Command API. You will create the role and attach this policy to that role later in this topic.

To create an IAM policy for EC2 instances

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. Beside **Create Your Own Policy**, choose **Select**.
4. Type a policy name (for example, *RunCommandInstance*) and description, and then copy and paste the following policy into the **Policy Document** field:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:GetDocument",
        "ssm:ListAssociations",
        "ssm:UpdateAssociationStatus",
        "ssm:UpdateInstanceInformation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2messages:AcknowledgeMessage",
        "ec2messages>DeleteMessage",
        "ec2messages:FailMessage",
        "ec2messages:GetEndpoint",
        "ec2messages:GetMessages",
        "ec2messages:SendReply"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeInstanceStatus"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
        "ds:CreateComputer",
        "ds:DescribeDirectories"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads"
    ],
    "Resource": "*"
  }
]
```

Important

In the last section of this IAM policy, you can restrict access to the Amazon S3 bucket by specifying an Amazon Resource Name (ARN). For example, you can change the last `"Resource": "*"` item to `"Resource": "arn:aws:s3:::AnS3Bucket/*"`

5. Choose **Validate Policy**. Verify that the policy is valid. If you receive an error, verify that you included the opening and closing brackets `{ }`. After the policy is validated, choose **Create Policy**.

Create the IAM User Policy

The IAM user policy determines which SSM documents a user can see in the **Command document** list. Users can see this list in either the Amazon EC2 console or by calling `ListDocuments` using the AWS CLI. The policy also limits the actions the user can perform with an SSM JSON document.

Note

You will create a user account and attach this policy to that account later on.

The IAM policy in the following procedure enables the user to perform any SSM action on the instance. Assign this policy only to trusted administrators. For all other users, create a restrictive IAM policy, as described in this section.

To create the IAM user policy

1. Repeat the previous procedure to create a policy for a user.
2. Copy and paste the following policy into the **Policy Document** field and create the policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:*",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*"
    }
  ]
}
```

Create a Restrictive IAM User Policy

Create restrictive IAM user policies to further delegate access to Run Command. The following example IAM policy allows a user to list SSM JSON documents and view details about those documents, send a command using the document, and cancel or view details about the command after it has been sent. The user has permission to execute the document on three instances, as determined by the "arn:aws:ec2:us-east-1:*:instance/i-xxxxxxxxxxxxxxxx" items in the second Resource section. If you want to give the user access to run the command on any instance for which the user currently has access (as determined by the AWS user account), you could specify "arn:aws:ec2:us-east-1:*:instance/*" in the Resource section and remove the other instance resources.

Note that the Resource section includes an S3 ARN entry:

```
arn:aws:s3:::bucket_name
```

You can also format this entry as follows:

```
arn:aws:s3:::bucket_name/*
arn:aws:s3:::bucket_name/key_prefix_name
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ssm:ListDocuments",
        "ssm:DescribeDocument",
        "ssm:GetDocument",
        "ssm:DescribeInstanceInformation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ssm:SendCommand",
      "Effect": "Allow",

```

```
    "Resource": [
      "arn:aws:ec2:us-east-1:*:instance/i-1234567890abcdef0",
      "arn:aws:ec2:us-east-1:*:instance/i-0598c7d356eba48d7",
      "arn:aws:ec2:us-east-1:*:instance/i-345678abcdef12345",
      "arn:aws:s3:::bucket_name",
      "arn:aws:ssm:us-east-1:*:document/RestartLinuxService"
    ],
  },
  {
    "Action": [
      "ssm:CancelCommand",
      "ssm:ListCommands",
      "ssm:ListCommandInvocations"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": "ec2:DescribeInstanceStatus",
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

For more information about creating IAM user policies, see [Managed Policies and Inline Policies](#).

Create the EC2 Instance Role

The EC2 Instance role enables the instance to communicate with the Run Command API. The role uses the EC2 instance policy you created earlier.

To create the EC2 instance role

1. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create New Role**.
2. On the **Set Role Name** page, enter a name for the role that designates it as the instance role, for example, *RunCommandInstance*. Choose **Next Step**.
3. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.
4. On the **Attach Policy** page, select the *RunCommandInstance* policy you created earlier. Choose **Next Step**.
5. Review the role information and then choose **Create Role**.

Verify the Trust Policy

If you want to use an existing EC2 instance role, you must verify that `ec2.amazonaws.com` is listed in the trust policy for the role. If you created a new EC2 instance role, you must add `ec2.amazonaws.com` as a trusted entity.

To verify the trust policy

1. In the navigation pane of the IAM console, choose **Roles**, and then choose the server role you just created.
2. Choose **Trust Relationships**.
3. Under **Trusted Entities** verify that `ec2.amazonaws.com` is listed. If it's not listed, choose **Edit Trust Relationship**.
4. Copy and paste the following policy into the **Policy Document** field and create the policy:


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Create the User Account

The user account enables a user to call the Run Command API on an instance. This account uses the IAM user policy you created earlier.

To create the user account

1. From the **Users** page on the [IAM console](#), choose **Create New Users**.
2. Specify a user name (for example, *RunCommandUser*) and verify that the **Generate an access key for each user** option is selected.
3. Choose **Create**.
4. Choose **Download Credentials**. By default, the system prompts you to save the credentials as a .csv file.

Important

Make a note of the *RunCommandUser* access key and secret key from the .csv file you downloaded.

5. Choose **Close**.
6. In the **IAM Dashboard**, choose **Users**, and then locate the user you just created.
7. Choose the user name (do *not* select the option beside the name), and then choose **Attach Policy**.
8. Choose the user policy you created earlier, and then choose **Attach Policy**.

Create EC2 Instances that Use the EC2 Instance Role

This procedure describes how to create an EC2 instance that uses the role you created. You must assign a role to an EC2 instance when you launch it. You can't assign a role to an instance that is already running. Instead, you would create an image of the instance, and then launch an instance from that image, with the role assigned.

To create an instance that use the EC2 instance role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Linux instance.
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In the **IAM role** drop-down list choose the EC2 instance role you created earlier.

6. Complete the wizard.

If you create other instances that you want to configure using Run Command, you must specify the EC2 instance role for each instance.

Setting Up Run Command On Managed Instances

You can remotely manage on-premises servers and virtual machines (VMs), and VMs from other cloud providers, using Amazon EC2 Run Command. An on-premises server or VM that is configured for Run Command is called a *managed instance*. By setting up managed instances, you create a consistent and secure way to remotely manage your on-premises and cloud workloads using the same tools or scripts.

Note

For information about setting up managed instances for Windows, see [Setting Up Run Command On Managed Instances \(On-Premises Servers and VMs\)](#) in the *Amazon EC2 User Guide for Windows Instances*.

Managed instances support the following features and services.

- Send commands using the [AWS CLI](#), [AWS Tools for Windows PowerShell](#), the **Commands** page in the [Amazon EC2 console](#), or AWS SDKs.
- Send commands using the public SSM documents or documents that you create. Public SSM documents currently include AWS-RunShellScript and AWS-UpdateSSMAgent.
- Control the Run Command actions that users can perform using AWS Identity and Access Management (IAM) roles and policies.
- Record and audit the Run Command actions that users perform using AWS CloudTrail.
- Create base images as templates for launching new servers using Sysprep.
- Organize managed instances using standard Amazon Elastic Compute Cloud (EC2) tags.

To get started using Run Command on managed instances

1. **Create IAM service and user roles:** The IAM service role enables your on-premises servers and VMs to communicate with the SSM service. The IAM user role enables users to communicate with the SSM API to execute commands from either the Amazon EC2 console or by directly calling the API. This topic describes how to create the service role and provides a link to a new topic with information about how to create a user role.
2. **Create a managed-instance activation:** A managed-instance activation registers your servers, VMs, or VMs hosted by other cloud providers with Amazon EC2 Simple Systems Manager. This topic describes how to create a managed-instance activation.
3. **Verify prerequisites:** Verify that your on-premises servers and VMs meet the minimum requirements for Run Command. For more information, see [Amazon EC2 Run Command Prerequisites \(p. 355\)](#).
4. **Deploy the SSM agent:** The SSM agent processes Run Command requests and configures your machine as specified in the request. You must download and install the SSM agent on your on-premises servers or VMs, or VMs hosted by other cloud providers. This topic describes how to install the SSM agent on your managed instances.

Create a Managed Instance Service Role

Managed instances require an IAM role to communicate with the SSM service. The role grants AssumeRole trust to the SSM service.

The following example shows how to create the IAM service role for SSM using AWS Tools for Windows PowerShell. For examples of how to create a service role using the AWS Management Console or the AWS CLI, see [Creating a Role to Delegate Permissions to an AWS Service](#).

Note

You only need to create the service role once for each AWS account.

To create an IAM service role for managed instances using AWS Tools for Windows PowerShell

1. Create a text file (in this example it is named `SSMService-Trust.json`) with the following trust policy. Save the file with the `.json` file extension.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ssm.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

2. Use [New-IAMRole](#) as follows to create a service role. This example creates a role named `SSMServiceRole`.

```
New-IAMRole -RoleName SSMServiceRole -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
```

3. Use [Register-IAMRolePolicy](#) as follows to enable the `SSMServiceRole` to create a session token. The session token gives your managed instance permission to execute commands using Run Command.

```
Register-IAMRolePolicy -RoleName SSMServiceRole -PolicyArn arn:aws:iam::ssm:policy/service-role/AmazonEC2RoleforSSM
```

You must now create IAM roles that enable users to communicate with the SSM API. For more information, see [Delegating Access to Amazon EC2 Run Command \(p. 365\)](#).

Create a Managed Instance Activation

To set up managed instances, you need to create a managed-instance activation. After you complete the activation, you receive an activation code and ID. This code/ID combination functions like an Amazon EC2 access ID and secret key to provide secure access to the Run Command service from your managed instances.

Important

Store the managed-instance activation code and ID in a safe place. You specify this code and ID when you install the SSM agent on an on-premises server or VM. If you lose them, you must create a new activation.

The procedures in this section require that you specify a [region where SSM is available](#). We recommend that you specify the region closest to your data center or computing environment.

Note

When you create a managed-instance activation, you specify a date when the activation expires. If you want to register additional managed instances after the expiry date, you must create a new activation. The expiry date has no impact on registered and running instances.

To create a managed-instance activation using the console

1. Open the [Amazon EC2 console](#), expand **Run Command** in the navigation pane, and choose **Activations**.
2. Choose **Create an Activation**.
3. Fill out the form and choose **Create Activation**.

To create a managed-instance activation using the AWS Tools for Windows PowerShell

1. Execute the following command.

```
New-SSMActivation -DefaultInstanceName name -IamRole IAM service role -  
RegistrationLimit number of managed instances -Region region
```

For example:

```
New-SSMActivation -DefaultInstanceName MyWebServers -IamRole  
RunCommandServiceRole -RegistrationLimit 10 -Region us-east-1
```

2. Press Enter. If the activation is successful, the system returns an activation code and an ID. Store the activation code and ID in a safe place.

To create a managed-instance activation using the AWS CLI

1. Execute the following command in the CLI.

```
aws ssm create-activation --default-instance-name name --iam-role IAM  
service role --registration-limit number of managed instances --  
region region
```

For example:

```
aws ssm create-activation --default-instance-name MyWebServers --iam-role  
RunCommandServiceRole --registration-limit 10 --region us-east-1
```

2. Press Enter. If the activation is successful, the system returns an activation code and an ID. Store the activation code and ID in a safe place.

Install the SSM Agent on Managed Instances

Before you begin, locate the activation code and ID that was sent to you after you completed the managed-instance activation. You will specify the code and ID in the following procedure.

To install the SSM agent on a managed instance

1. Log on to the on-premises server or VM.
2. Copy and paste the following command block into the AWS CLI. Specify your activation code, activation ID, and the region where you want to download the SSM agent from. Note that `sudo` is not necessary if you are a root user.

On Amazon Linux, RHEL 6.x, and CentOS 6.x

```
mkdir /tmp/ssm
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Executing Commands

```
sudo curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/  
amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm  
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm  
sudo stop amazon-ssm-agent  
sudo amazon-ssm-agent -register -code "code" -id "id" -region "region"  
sudo start amazon-ssm-agent
```

On RHEL 7.x and CentOS 7.x

```
mkdir /tmp/ssm  
sudo curl https://amazon-ssm-region.s3.amazonaws.com/latest/linux_amd64/  
amazon-ssm-agent.rpm -o /tmp/ssm/amazon-ssm-agent.rpm  
sudo yum install -y /tmp/ssm/amazon-ssm-agent.rpm  
sudo systemctl stop amazon-ssm-agent  
sudo amazon-ssm-agent -register -code "code" -id "id" -region "region"  
sudo systemctl start amazon-ssm-agent
```

On Ubuntu

```
mkdir /tmp/ssm  
sudo curl https://amazon-ssm-region.s3.amazonaws.com/latest/debian_amd64/  
amazon-ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb  
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb  
sudo stop amazon-ssm-agent  
sudo amazon-ssm-agent -register -code "code" -id "id" -region "region"  
sudo start amazon-ssm-agent
```

3. Press Enter.

The command downloads and installs the SSM agent onto the local server or VM. The command also registers the server or VM with the SSM service. The server or VM is now a managed instance.

For information about how to execute commands on managed instances using Run Command, see [Executing a Command Using Amazon EC2 Run Command \(p. 374\)](#).

Executing a Command Using Amazon EC2 Run Command

You can run commands on EC2 instances using the **Command History** page in the [Amazon EC2 console](#). You can also execute commands using the [CLI](#), the [Amazon EC2 Simple Systems Manager API Reference](#), or the [AWS SDKs](#). For more information, see the .

Topics

- [Running Shell Scripts Using Amazon EC2 Run Command \(p. 374\)](#)
- [Updating the Amazon SSM Agent Using Amazon EC2 Run Command \(p. 375\)](#)

Running Shell Scripts Using Amazon EC2 Run Command

You can use the [AWS-RunShellScript](#) document to run shell scripts on your instances.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the

Command History page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status](#) (p. 396).

To run shell scripts using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-RunShellScript**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Amazon EC2 Run Command Prerequisites](#) (p. 355).
5. Enter a script or the path to a script in the **Commands** field.
6. (Optional) In the **Working Directory** field, enter the path to the folder on your EC2 instances where you want to run the command.
7. (Optional) In the **Execution Timeout** field, enter the number of seconds the Amazon SSM agent will attempt to run the command before it times out and fails.
8. (Optional) In the **Comment** field, enter information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

9. In the **Timeout (seconds)** field, enter the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
10. In the **Amazon S3 bucket** field enter the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

11. In the **S3 key prefix**, field enter the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

For information about how to view the results of your command, see [Viewing Command Output in the Amazon EC2 Console](#) (p. 377).

For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Updating the Amazon SSM Agent Using Amazon EC2 Run Command

You can use the AWS-UpdateSSMAgent document to update the Amazon SSM agent running on your instances. You can update to either the latest version or downgrade to an older version. When you execute the command, the system downloads the version from AWS, installs it, and then uninstalls the

version that existed before the command was run. If an error occurs during this process, the system rolls back to the version on the server before the command was run and the command status shows that the command failed.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 396\)](#).

To update the Amazon SSM Agent using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-UpdateSSMAgent**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Amazon EC2 Run Command Prerequisites \(p. 355\)](#).
5. (Optional) In the **Version** field, type a specific version of the Amazon SSM agent to install. You can install older versions of the agent. If you do not specify a version, the service will be updated to the latest version.
6. (Optional) In the **Allow downgrade** list, choose **True** if you want to install an earlier version of the Amazon SSM agent. If you choose this option, you must specify the earlier version number. Choose **False** if you want the system to install only the newest version of the service.
7. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

8. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
9. In the **S3 bucket** field type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

10. In the **S3 key prefix**, field enter the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

For information about how to view the results of your command, see [Viewing Command Output in the Amazon EC2 Console \(p. 377\)](#).

For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Viewing Command Output in the Amazon EC2 Console

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

<input type="checkbox"/>	Command ID	Instance ID	Document name	Status	Requested date
<input type="checkbox"/>	edf7712c-1d2b-...	i-e6c6f145	AWS-RunShellScript	Success	November 29, 2015
<input checked="" type="checkbox"/>	edf7712c-1d2b-...	i-a65e3219	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	edf7712c-1d2b-...	i-016edad5	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	755fa4c0-7a67-...	i-016edad5	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	755fa4c0-7a67-...	i-e6c6f145	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	755fa4c0-7a67-...	i-a65e3219	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	70aae367-f5ce-...	i-e6c6f145	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	70aae367-f5ce-...	i-016edad5	AWS-RunShellScript	Success	November 29, 2015
<input type="checkbox"/>	70aae367-f5ce-...	i-a65e3219	AWS-RunShellScript	Success	November 29, 2015

Command ID: edf7712c-1d2b-4af9-a703-964008a233a0 Instance ID: i-a65e3219

Description **Output**

Output

Plugin name	Status	Response code	Start Time	Finish Time
aws:runShell...	Success	0	November 29, 2015 at 8:55:41 ...	November 29, 2015

4. The command output page shows the results of your command execution.

[Commands](#) > Output

Output for aws:runShellScript

```
eth0    Link encap:Ethernet  HWaddr 12:97:34:47:9b:55
        inet addr:172.31.44.89  Bcast:172.31.47.255  Mask:255.255.240.0
        inet6 addr: fe80::1097:34ff:fe47:9b55/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
        RX packets:2019936 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2110772 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:223188147 (223.1 MB)  TX bytes:523059869 (523.0 MB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Creating SSM Documents

When you execute a command using Amazon EC2 Run Command, the system reads the actions to be performed from a document that defines the plugins to run and parameters to use. This document is called an SSM document. The first time you execute a command from a new SSM document, the system stores the document with your AWS account.

Limitations

As you begin working with SSM documents, be aware of the following limitations.

- You can create a maximum of 200 SSM documents per AWS account.
- SSM documents that you create are only available in the region where you created them. To add a document in another region, copy the content and recreate it in the new region.

Note

If you need to create more than the maximum number of SSM documents, contact AWS Support.

When giving a user access to Run Command the best practice is to start with a policy of least privilege. Create different SSM documents that allow the user to do a minimum number of tasks. For example, you could specify the name of a specific service on an instance so that a Run Command user could only restart that service.

Create an SSM Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents** and then choose **Create Document**.
3. Enter a descriptive name for the document and then specify plugins and parameters in the **Content** field in JSON format. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
4. Choose **Create Document** to save it with your AWS user account.

Create an SSM Document Using Windows PowerShell

1. Specify plugins and parameters in a file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
2. Create the document and save it with your AWS user account using AWS Tools for Windows PowerShell.

```
$json = Get-Content C:\your file | Out-String  
New-SSMDocument -Name document name -Content $json
```

Create an SSM Document Using the AWS CLI

1. Specify plugins and parameters in a file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
2. Create the document and save it with your AWS user account using the AWS CLI.

```
aws ssm create-document --content file://c:\temp\your file --name  
"document name"
```

Sample SSM Documents

SSM documents are currently supported in JavaScript Object Notation (JSON) and use the following:

- `schemaVersion 1.2`
- A `runtimeConfig` that uses one or more plugins to execute tasks. Plugins are platform specific, meaning they run on either a supported version of Windows or Linux. For more information about plugins, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

Use the following examples as a foundation to create your own documents.

Restrictive SSM Document for Linux

The following example shows a highly-restrictive SSM document that uses the `AWS-RunShellScript` document on Linux. The user can only run the `ifconfig` command to check the IP configuration of the instance:

```
{
  "schemaVersion": "1.2",
  "description": "Check ip configuration of a Linux instance.",
  "parameters": {
  },
  "runtimeConfig": {
    "aws:runShellScript": {
      "properties": [
        {
          "id": "0.aws:runShellScript",
          "runCommand": ["ifconfig"]
        }
      ]
    }
  }
}
```

This section includes several JSON templates that you can use to create your own SSM documents. These templates are based on the AWS public SSM documents. <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/>;

AWS-RunShellScript

```
{
  "schemaVersion": "1.2",
  "description": "Run a shell script or specify the commands to run.",
  "parameters": {
    "commands": {
      "type": "StringList",
      "description": "(Required) Specify a shell script or a command to run.",
      "minItems": 1,
      "displayType": "textarea"
    },
    "workingDirectory": {
      "type": "String",
      "default": "",
      "description": "(Optional) The path to the working directory on your instance.",
      "maxChars": 4096
    },
    "executionTimeout": {
      "type": "String",
      "default": "3600",
      "description": "(Optional) The time in seconds for a command to complete before it is considered to have failed. Default is 3600 (1 hour). Maximum is 28800 (8 hours).",
      "allowedPattern": "([1-9][0-9]{0,3})|(1[0-9]{1,4})|(2[0-7][0-9]{1,3})|(28[0-7][0-9]{1,2})|(28800)"
    }
  },
  "runtimeConfig": {
    "aws:runShellScript": {
      "properties": [
        {
          "id": "0.aws:runShellScript",
```

```
        "runCommand": "{{ commands }}",
        "workingDirectory": "{{ workingDirectory }}",
        "timeoutSeconds": "{{ executionTimeout }}"
    }
}
}
```

AWS-UpdateSSMAgent

```
{
  "schemaVersion": "1.2",
  "description": "Update the Amazon SSM Agent to the latest version or
specified version.",
  "parameters": {
    "version": {
      "default": "",
      "description": "(Optional) A specific version of the Amazon SSM
Agent to install. If not specified, the agent will be updated to the latest
version.",
      "type": "String"
    },
    "allowDowngrade": {
      "default": "false",
      "description": "(Optional) Allow the Amazon SSM Agent service
to be downgraded to an earlier version. If set to false, the service can
be upgraded to newer versions only (default). If set to true, specify the
earlier version.",
      "type": "String",
      "allowedValues": [
        "true",
        "false"
      ]
    }
  },
  "runtimeConfig": {
    "aws:updateSsmAgent": {
      "properties": [
        {
          "agentName": "amazon-ssm-agent",
          "allowDowngrade": "{{ allowDowngrade }}",
          "targetVersion": "{{ version }}"
        }
      ]
    }
  }
}
```

Sharing SSM Documents

You can share Amazon EC2 Simple Systems Manager (SSM) documents privately or publicly. To privately share an SSM document, you modify the document permissions and allow specific individuals to access it according to their Amazon Web Services (AWS) ID. To publicly share a SSM document, you modify the document permissions and specify All.

Warning

Use shared SSM documents only from trusted sources. When using any shared document, carefully review the contents of the document before using it so that you understand how it will change the configuration of your instance. For more information about shared document best practices, see [Guidelines for Sharing and Using Shared SSM Documents \(p. 382\)](#).

Limitations

As you begin working with SSM documents, be aware of the following limitations.

- Only the owner can share a document.
- You must stop sharing a document before you can delete it. For more information, see [How to Modify Permissions for a Shared Document \(p. 385\)](#).
- You can share a document with a maximum of 20 AWS accounts.
- You can publicly share a maximum of five SSM documents.

Note

If you need to share more than the maximum number of AWS accounts or SSM documents, contact AWS Support.

This topic includes the following sections.

- [Guidelines for Sharing and Using Shared SSM Documents \(p. 382\)](#)
- [How to Share an SSM Document \(p. 383\)](#)
- [How to Modify Permissions for a Shared Document \(p. 385\)](#)
- [How to Use a Shared SSM Document \(p. 385\)](#)

Guidelines for Sharing and Using Shared SSM Documents

Review the following guidelines before you share or use a shared document.

Remove Sensitive Information

Review your SSM document carefully and remove any sensitive information. For example, verify that the document does not include your AWS credentials. If you share a document with specific individuals, those users can view the information in the document. If you share a document publicly, anyone can view the information in the document.

Limit Run Command Actions Using an IAM User Trust Policy

Create a restrictive AWS Identity and Access Management (IAM) user policy for users who will have access to the document. The IAM policy determines which SSM documents a user can see in either the Amazon EC2 console or by calling `ListDocuments` using the AWS CLI or AWS Tools for Windows PowerShell. The policy also limits the actions the user can perform with an SSM document. You can create a restrictive policy so that a user can only use specific documents. For more information, see [Delegating Access to Amazon EC2 Run Command \(p. 365\)](#).

Review the Contents of a Shared Document Before Using It

Review the contents of every document that is shared with you, especially public documents, to understand the commands that will be executed on your instances. A document could intentionally or unintentionally have negative repercussions after it is run. If the document references an external network, review the external source before you use the document.

Send Commands Using the Document Hash

When you share a document, the system creates a Sha-256 hash and assigns it to the document. The system also saves a snapshot of the document content. When you send a command using a shared document, you can specify the hash in your command to ensure that the following conditions are true:

- You are executing a command from the correct SSM document
- The content of the document has not changed since it was shared with you.

If the hash does not match the specified document or if the content of the shared document has changed, the command returns an `InvalidDocument` exception. Note: The hash cannot verify document content from external locations.

How to Share an SSM Document

You can share an SSM document by using the Amazon EC2 console or by programmatically calling the `ModifyDocumentPermission` API operation using the AWS CLI, AWS Tools for Windows PowerShell, or the AWS SDK. Before you share a document, get the AWS account IDs of the people with whom you want to share. You will specify these account IDs when you share the document.

Share a Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents**.
3. In the documents list, choose the document you want to share. Choose the **Permissions** tab and verify that you are the document owner. Only a document owner can share a document.
4. Choose **Edit**.
5. To share the command publicly, choose **Public** and then choose **Save**. To share the command privately, choose **Private**, enter the AWS account ID, choose **Add Permission**, and then choose **Save**.

Share a Document Using the AWS CLI

The following procedure requires that you specify a region for your CLI session. Run Command is currently available in the following SSM [regions](#).

1. Open the AWS CLI on your local computer and execute the following command to specify your credentials.

```
aws config

AWS Access Key ID: [your key]
AWS Secret Access Key: [your key]
Default region name:[us-east-1]
Default output format [None]:
```

2. Use the following command to list all of the SSM documents that are available for you. The list includes documents that you created and documents that were shared with you.

```
aws ssm list-documents --document-filter-list key=Owner,value=all
```

3. Use the following command to get a specific document.

```
aws ssm get-document --name document name
```

4. Use the following command to get a description of the document.

```
aws ssm describe-document --name document name
```

5. Use the following command to view the permissions for the document.

Amazon Elastic Compute Cloud User Guide for Linux Instances Sharing SSM Documents

```
aws ssm describe-document-permission --name document name --permission-type Share
```

6. Use the following command to modify the permissions for the document and share it. You must be the owner of the document to edit the permissions. This command privately shares the document with a specific individual, based on that person's AWS account ID.

```
aws ssm modify-document-permission --name document name --permission-type Share --account-ids-to-add AWS account ID
```

Use the following command to share a document publicly.

```
aws ssm modify-document-permission --name document name --permission-type Share --account-ids-to-add 'all'
```

Share a Document Using AWS Tools for Windows PowerShell

The following procedure requires that you specify a region for your PowerShell session. Run Command is currently available in the following SSM [regions](#).

1. Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to specify your credentials.

```
Set-AWSCredentials -AccessKey your key -SecretKey your key
```

2. Use the following command to set the region for your PowerShell session. The example uses the us-west-2 region.

```
Set-DefaultAWSRegion -Region us-west-2
```

3. Use the following command to list all of the SSM documents available for you. The list includes documents that you created and documents that were shared with you.

```
Get-SSMDocumentList -DocumentFilterList (@{"key"="Owner";"value"="All"})
```

4. Use the following command to get a specific document.

```
Get-SSMDocument -Name document name
```

5. Use the following command to get a description of the document.

```
Get-SSMDocumentDescription -Name document name
```

6. Use the following command to view the permissions of the document.

```
Get-SSMDocumentPermission -Name document name -PermissionType Share
```

7. Use the following command to modify the permissions for the document and share it. You must be the owner of the document to edit the permissions. This command privately shares the document with a specific individual, based on that person's AWS account ID.

```
Edit-SSMDocumentPermission -Name document name -PermissionType Share -  
AccountIdsToAdd AWS account ID
```

Use the following command to share a document publicly.

```
Edit-SSMDocumentPermission -Name document name -AccountIdsToAdd ('all') -  
PermissionType Share
```

How to Modify Permissions for a Shared Document

If you share a command, users can view and use that command until you either remove access to the SSM document or delete the SSM document. However, you cannot delete a document as long as it is shared. You must stop sharing it first and then delete it.

Stop Sharing a Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents**.
3. In the documents list, choose the document you want to stop sharing. Choose the **Permissions** tab and verify that you are the document owner. Only a document owner can stop sharing a document.
4. Choose **Edit**.
5. Delete the AWS account ID that should no longer have access to the command, and then choose **Save**.

Stop Sharing a Document Using the AWS CLI

Open the AWS CLI on your local computer and execute the following command to stop sharing a command.

```
aws ssm modify-document-permission --name document name --permission-type  
Share --account-ids-to-remove 'AWS account ID'
```

Stop Sharing a Document Using AWS Tools for Windows PowerShell

Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to stop sharing a command.

```
Edit-SSMDocumentPermission -Name document name -AccountIdsToRemove AWS  
account ID -PermissionType Share
```

How to Use a Shared SSM Document

When you share an SSM document, the system generates an Amazon Resource Name (ARN) and assigns it to the command. If you select and execute a shared document from the Amazon EC2 console, you do not see the ARN. However, if you want to execute a shared SSM document from a command line application, you must specify a full ARN. You are shown the full ARN for an SSM document when you execute the command to list documents.

Note

You are not required to specify ARNs for AWS public documents (documents that begin with AWS-*) or commands that you own.

This section includes examples of how to view and execute shared SSM documents from the AWS CLI and AWS Tools for Windows PowerShell.

Using a Shared SSM Document from the AWS CLI

To list all public SSM documents

```
aws ssm list-documents --document-filter-list key=Owner,value=Public
```

To list private SSM documents that have been shared with you

```
aws ssm list-documents --document-filter-list key=Owner,value=Private
```

To list all SSM documents available to you

```
aws ssm list-documents --document-filter-list key=Owner,value=All
```

Execute a command from a shared SSM document using a full ARN

```
aws ssm send-command --document-name FullARN/name
```

For example:

```
aws ssm send-command --document-name arn:aws:ssm:us-east-1:12345678912:document/highAvailabilityServerSetup --instance-ids i-12121212
```

Using a Shared SSM Document from the AWS Tools for Windows PowerShell

To list all public SSM documents

```
Get-SSMDocumentList -DocumentFilterList @(New-Object Amazon.SimpleSystemsManagement.Model.DocumentFilter("Owner", "Public"))
```

To list private SSM documents that have been shared with you

```
Get-SSMDocumentList -DocumentFilterList @(New-Object Amazon.SimpleSystemsManagement.Model.DocumentFilter("Owner", "Shared"))
```

To get information about an SSM document that has been shared with you

```
Get-SSMDocument -Name FullARN/name
```

For example:

```
Get-SSMDocument -Name arn:aws:ssm:us-east-1:12345678912:document/highAvailabilityServerSetup
```

To get a description of an SSM document that has been shared with you

```
Get-SSMDocumentDescription -Name FullARN/name
```

For example:

```
Get-SSMDocumentDescription -Name arn:aws:ssm:us-east-1:12345678912:document/  
highAvailabilityServerSetup
```

To execute a command from a shared SSM document using a full ARN

```
Send-SSMCommand -DocumentName FullARN/name -InstanceId IDs
```

For example:

```
Send-SSMCommand -DocumentName arn:aws:ssm:us-east-1:555450671542:document/  
highAvailabilityServerSetup -InstanceId @"i-273d4e9e"
```

Amazon EC2 Run Command Walkthroughs

The following examples or walkthroughs to help you understand how to execute commands using Run Command from either the Amazon EC2 console or the AWS CLI.

Caution

If this is your first time using Run Command, we recommend executing commands against a test instance or an instance that is not being used in a production environment.

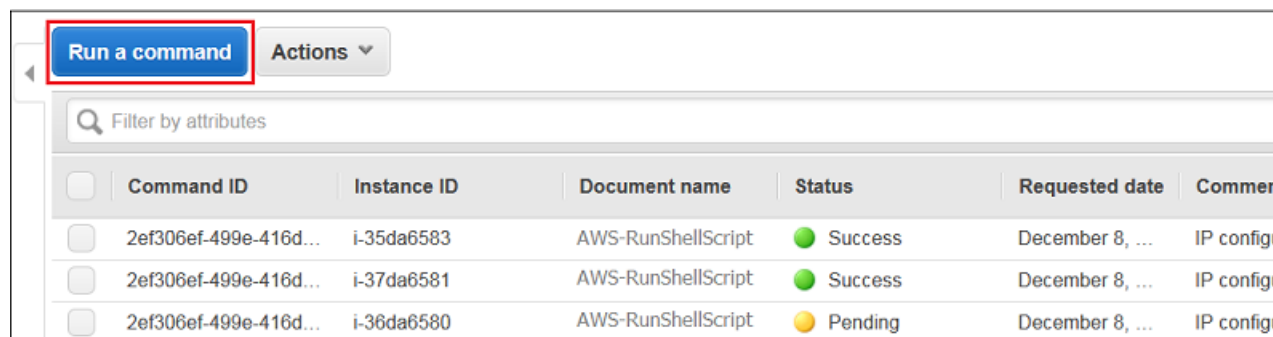
Contents

- [Amazon EC2 Run Command Walkthrough Using the Console \(p. 387\)](#)
- [Amazon EC2 Run Command Walkthrough Using the AWS CLI \(p. 391\)](#)

Amazon EC2 Run Command Walkthrough Using the Console

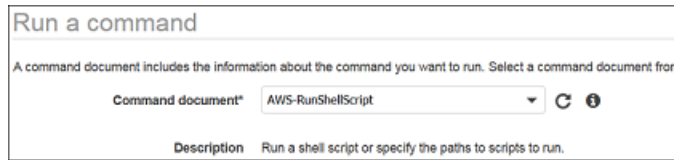
The following sample walkthrough shows you how to execute commands using Run Command from the **Command History** page in the Amazon EC2 console. You will learn how to execute a command with the AWS-RunShellScript SSM document. To view CLI examples, see [Amazon EC2 Run Command Walkthrough Using the AWS CLI \(p. 391\)](#).

1. In the [Amazon EC2 console](#) choose **Command History** in the navigation pane, and then choose **Run a command**.



Amazon Elastic Compute Cloud User Guide for Linux Instances Walkthroughs

2. In the **Command document** list, choose **AWS-RunShellScript**.



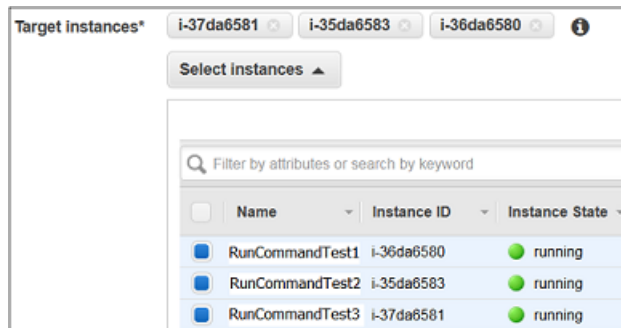
Run a command

A command document includes the information about the command you want to run. Select a command document from

Command document* AWS-RunShellScript

Description Run a shell script or specify the paths to scripts to run.

3. Choose **Select instances** and then choose the instances where you want to execute the command. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Amazon EC2 Run Command Prerequisites](#) (p. 355).



Target instances* i-37da6581 i-35da6583 i-36da6580

Select instances

Filter by attributes or search by keyword

<input type="checkbox"/>	Name	Instance ID	Instance State
<input checked="" type="checkbox"/>	RunCommandTest1	i-36da6580	running
<input checked="" type="checkbox"/>	RunCommandTest2	i-35da6583	running
<input checked="" type="checkbox"/>	RunCommandTest3	i-37da6581	running

4. Type a valid shell script or command in the **Commands** field. You can specify a **Working Directory** and **Execution Timeout**, if you want. The **Execution Timeout** is the number of seconds the amazon-ssm-agent will attempt to run the command before it is considered to have failed. We recommend entering a comment in the **Comments** field. A comment will help you identify the command in the list of pending commands and make it easier to view the output.



Commands* ps aux | less

Working Directory

Execution Timeout 3600

Comment List all running processes

5. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails.
6. In the **S3 bucket** field type the name of an Amazon S3 bucket where you want to store command output. Enter an Amazon S3 subfolder in the **S3 key prefix** field. A subfolder can help you organize output if you are executing multiple commands against multiple instances.

Amazon Elastic Compute Cloud User Guide for Linux Instances Walkthroughs

Timeout (seconds) ⓘ

S3 bucket ⓘ

S3 key prefix ⓘ

7. Choose **Run** to execute the command simultaneously on the selected instances. Run Command displays a status screen.
8. Choose **View result**.

Run a command

✓ **Success**
We are running your command against the instances listed below.

Instance IDs i-36da6580, i-37da6581, i-35da6583

Command ID a0f0eb16-5073-482b-b875-0c8ae9038cbf

9. The command list shows three invocations for the command because it was sent to three instances. Each invocation has its own **Command ID** and status. To view status, choose an invocation, choose the **Output** tab for the invocation, and then choose **View Output**.

Run a command Actions ▾

Filter by attributes

<input type="checkbox"/>	Command ID	Instance ID	Document name	Status	Requested date	Comment
<input checked="" type="checkbox"/>	a0f0eb16-5073-482...	i-35da6583	AWS-RunShellScript	● Success	December 8, ...	Checking ser
<input type="checkbox"/>	a0f0eb16-5073-482...	i-36da6580	AWS-RunShellScript	● Success	December 8, ...	Checking ser
<input type="checkbox"/>	a0f0eb16-5073-482...	i-37da6581	AWS-RunShellScript	● Success	December 8, ...	Checking ser

Command ID: a0f0eb16-5073-482b-b875-0c8ae9038cbf **Instance ID:** i-35da6583

Description **Output**

Output

Plugin name	Status	Response code	Start Time	Finish Time
aws:runShellS...	● Success	0	December 8, 2015 at 5:26:40 P...	December 8, 2015 at 5:26:40 P...

10. The system displays the output in your browser. If the output is longer than 2500 characters, only the first 2500 characters are shown and the rest is truncated.

Amazon Elastic Compute Cloud User Guide for Linux Instances Walkthroughs

```

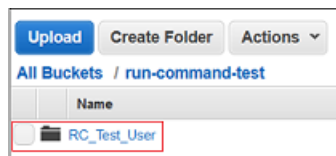
Commands > Output

Output for aws:runShellScript

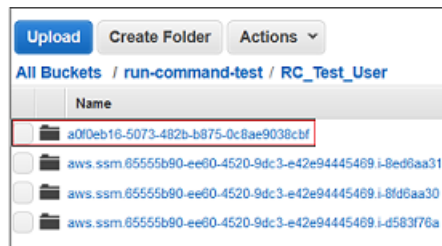
root      2  0.0  0.0    0    0 ?      S   Dec08  0:00 [kthreadd]
root      3  0.0  0.0    0    0 ?      S   Dec08  0:00 [ksoftirqd/0]
root      5  0.0  0.0    0    0 ?      S<  Dec08  0:00 [kworker/0:0H]
root      6  0.0  0.0    0    0 ?      S   Dec08  0:00 [kworker/u30:0]
root      7  0.0  0.0    0    0 ?      S   Dec08  0:00 [rcu_sched]
root      8  0.0  0.0    0    0 ?      S   Dec08  0:00 [rcu_bh]
root      9  0.0  0.0    0    0 ?      S   Dec08  0:00 [migration/0]
root     10  0.0  0.0    0    0 ?      S   Dec08  0:00 [migration/1]
root     11  0.0  0.0    0    0 ?      S   Dec08  0:00 [ksoftirqd/1]
root     12  0.0  0.0    0    0 ?      S   Dec08  0:00 [kworker/1:0]
root     13  0.0  0.0    0    0 ?      S<  Dec08  0:00 [kworker/1:0H]
root     14  0.0  0.0    0    0 ?      S<  Dec08  0:00 [khelper]
root     15  0.0  0.0    0    0 ?      S   Dec08  0:00 [kdevtmpfs]
root     16  0.0  0.0    0    0 ?      S<  Dec08  0:00 [netns]
root     17  0.0  0.0    0    0 ?      S   Dec08  0:00 [kworker/u30:1]
root     19  0.0  0.0    0    0 ?      S<  Dec08  0:00 [perf]
root     24  0.0  0.0    0    0 ?      S   Dec08  0:00 [xenwatch]
root     25  0.0  0.0    0    0 ?      S   Dec08  0:00 [xenbus]
root     26  0.0  0.0    0    0 ?      S   Dec08  0:00 [kworker/0:1]
root    129  0.0  0.0    0    0 ?      S   Dec08  0:00 [khungtaskd]
root    130  0.0  0.0    0    0 ?      S<  Dec08  0:00 [writeback]
root    132  0.0  0.0    0    0 ?      SN  Dec08  0:00 [ksmd]
root    133  0.0  0.0    0    0 ?      SN  Dec08  0:00 [khugepaged]
root    134  0.0  0.0    0    0 ?      S<  Dec08  0:00 [crypto]
root    135  0.0  0.0    0    0 ?      S<  Dec08  0:00 [kintegrityd]
root    136  0.0  0.0    0    0 ?      S<  Dec08  0:00 [bioset]
root    138  0.0  0.0    0    0 ?      S<  Dec08  0:00 [kblockd]
root    488  0.0  0.0    0    0 ?      S<  Dec08  0:00 [md]
root    494  0.0  0.0    0    0 ?      S   Dec08  0:00 [kworker/1:1]
ro
---Output truncated---

```

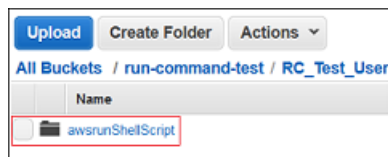
11. You can view the full command output in Amazon S3. Open the [S3 console](#) and choose your Amazon S3 bucket.



12. Choose the *Command-ID.Instance-ID* for which you want to view command output.



13. Choose the **awsrunShellScript** sub-folder.



14. Choose the **stdout.text** file. Amazon S3 displays the full command output.

Amazon Elastic Compute Cloud User Guide for Linux Instances Walkthroughs

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	19616	2596	?	Ss	Dec08	0:00	/sbin/init
root	2	0.0	0.0	0	0	?	S	Dec08	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	Dec08	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	Dec08	0:00	[kworker/0:0H]
root	6	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/u30:0]
root	7	0.0	0.0	0	0	?	S	Dec08	0:00	[rcu_sched]
root	8	0.0	0.0	0	0	?	S	Dec08	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	Dec08	0:00	[migration/0]
root	10	0.0	0.0	0	0	?	S	Dec08	0:00	[migration/1]
root	11	0.0	0.0	0	0	?	S	Dec08	0:00	[ksoftirqd/1]
root	12	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/1:0]
root	13	0.0	0.0	0	0	?	S<	Dec08	0:00	[kworker/1:0H]
root	14	0.0	0.0	0	0	?	S<	Dec08	0:00	[khelper]
root	15	0.0	0.0	0	0	?	S	Dec08	0:00	[kdevtmpfs]
root	16	0.0	0.0	0	0	?	S<	Dec08	0:00	[netns]
root	17	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/u30:1]
root	19	0.0	0.0	0	0	?	S<	Dec08	0:00	[perf]
root	24	0.0	0.0	0	0	?	S	Dec08	0:00	[xenwatch]
root	25	0.0	0.0	0	0	?	S	Dec08	0:00	[xenbus]
root	26	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/0:1]
root	129	0.0	0.0	0	0	?	S	Dec08	0:00	[khungtaskd]
root	130	0.0	0.0	0	0	?	S<	Dec08	0:00	[writeback]
root	132	0.0	0.0	0	0	?	SN	Dec08	0:00	[ksmd]
root	133	0.0	0.0	0	0	?	SN	Dec08	0:00	[khugepaged]
root	134	0.0	0.0	0	0	?	S<	Dec08	0:00	[crypto]
root	135	0.0	0.0	0	0	?	S<	Dec08	0:00	[kintegrityd]
root	136	0.0	0.0	0	0	?	S<	Dec08	0:00	[bioset]
root	138	0.0	0.0	0	0	?	S<	Dec08	0:00	[kblockd]
root	488	0.0	0.0	0	0	?	S<	Dec08	0:00	[md]
root	494	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/1:1]
root	619	0.0	0.0	0	0	?	S	Dec08	0:00	[kswapd0]
root	690	0.0	0.0	0	0	?	S	Dec08	0:00	[fsnotify_mark]
root	705	0.0	0.0	0	0	?	S<	Dec08	0:00	[kthrotld]
root	754	0.0	0.0	0	0	?	S<	Dec08	0:00	[deferwq]
root	1424	0.0	0.0	0	0	?	S<	Dec08	0:00	[ata_sff]
root	1432	0.0	0.0	0	0	?	S	Dec08	0:00	[scsi_ah_0]
root	1438	0.0	0.0	0	0	?	S<	Dec08	0:00	[scsi_tmf_0]
root	1441	0.0	0.0	0	0	?	S	Dec08	0:00	[scsi_ah_1]
root	1442	0.0	0.0	0	0	?	S<	Dec08	0:00	[scsi_tmf_1]
root	1505	0.0	0.0	0	0	?	S	Dec08	0:00	[jbd2/xvda1-8]
root	1506	0.0	0.0	0	0	?	S<	Dec08	0:00	[ext4-rsv-conver]
root	1548	0.0	0.0	11424	2736	?	Ss	Dec08	0:00	/sbin/udevd -d
root	1670	0.0	0.0	0	0	?	S<	Dec08	0:00	[kpsmouse]
root	1673	0.0	0.0	0	0	?	S	Dec08	0:00	[kworker/0:2]
root	1686	0.0	0.0	11316	2236	?	S	Dec08	0:00	/sbin/udevd -d
root	1687	0.0	0.0	11240	1956	?	S	Dec08	0:00	/sbin/udevd -d

Amazon EC2 Run Command Walkthrough Using the AWS CLI

The following sample walkthrough shows you how to use the AWS CLI to view information about commands and command parameters, how to execute commands, and how to view the status of those commands.

Tip

The **Command History** page in the console includes a section called **AWS Command Line Interface command**. This section displays a usable CLI script that's generated based on the parameters you entered.

Step 1: Getting Started

You must either have administrator privileges on the instances you want to configure or you must have been granted the appropriate permission in IAM. Also note, this example uses the us-east-1 region. Run Command is currently available in the following SSM [regions](#). For more information, see [Amazon EC2 Run Command Prerequisites](#) (p. 355).

To execute commands using the AWS CLI

1. Open the AWS CLI on your local computer and execute the following command to specify your credentials and the region.

```
aws configure
```

2. The system prompts you to specify the following.

```
AWS Access Key ID [None]: key_name  
AWS Secret Access Key [None]: key_name  
Default region name [None]: us-east-1  
Default output format [None]: ENTER
```

3. List all available documents

This command lists all of the documents available for your account based on IAM permissions. The command returns a list of Linux and Windows documents.

```
aws ssm list-documents
```

4. Verify that an instance is ready to receive commands

The output of the following command shows if instances are online.

```
aws ssm describe-instance-information --output text --query  
"InstanceInformationList[*]"
```

5. Use the following command to view details about a particular instance.

Note

To execute the commands in this walkthrough, you must replace the instance and command IDs. The command ID is returned as a response of the **send-command**. The instance ID is available from the Amazon EC2 console.

```
aws ssm describe-instance-information --instance-information-filter-list  
key=InstanceIds,valueSet=instance ID
```

Step 2: Running Shell Scripts

Using Run Command and the AWS-RunShellScript document, you can execute any command or script on an EC2 instance as if you were logged on locally.

To view the description and available parameters

- Use the following command to view a description of the SSM JSON document.

```
aws ssm describe-document --name "AWS-RunShellScript" --query  
"[Document.Name,Document.Description]"
```

- Use the following command to view the available parameters and details about those parameters.

```
aws ssm describe-document --name "AWS-RunShellScript" --query  
"Document.Parameters[*]"
```

Step 3: Send a Command Using the AWS-RunShellScript document - Example 1

Use the following command to get IP information for an instance.

```
aws ssm send-command --instance-ids "instance ID" --document-name "AWS-RunShellScript" --comment "IP config" --parameters commands=ifconfig --output text
```

Get command information with response data

The following command uses the Command ID that was returned from the previous command to get the details and response data of the command execution. The system returns the response data if the command completed. If the command execution shows "Pending" you will need to execute this command again to see the response data.

```
aws ssm list-command-invocations --command-id "command ID" --details
```

Step 4: Send a Command Using the AWS-RunShellScript document - Example 2

The following command displays the default user account running the commands.

```
sh_command_id=$(aws ssm send-command --instance-ids "instance ID" --document-name "AWS-RunShellScript" --comment "Demo run shell script on Linux Instance" --parameters commands=whoami --output text --query "Command.CommandId")
```

Get command status

The following command uses the Command ID to get the status of the command execution on the instance. This example uses the Command ID that was returned in the previous command.

```
aws ssm list-commands --command-id $sh_command_id
```

Get command details

The following command uses the Command ID from the previous command to get the status of the command execution on a per instance basis.

```
aws ssm list-command-invocations --command-id $sh_command_id --details
```

Get command information with response data for a specific instance

The following command returns the output of the original aws ssm send-command for a specific instance.

```
aws ssm list-command-invocations --instance-id instance ID --command-id $sh_command_id --details
```

Step 5: Additional Examples

The following command returns the version of Python running on an instance.

```
sh_command_id=$(aws ssm send-command --instance-ids "instance ID" --document-name "AWS-RunShellScript" --comment "Demo run shell script on Linux
```


Amazon Elastic Compute Cloud User Guide for Linux Instances Cancelling a Command

```
Instances" --parameters commands='python' --version --output text --query  
"Command.CommandId" )
```

The following command executes a Python script using Run Command.

```
py_command_id=$(aws ssm send-command --instance-ids "instance ID" --  
document-name "AWS-RunShellScript" --comment "Demo run shell script on Linux  
Instances" --parameters commands='#!/usr/bin/python', 'print \"Hello world  
from python\"' --output text --query "Command.CommandId" )
```

Cancelling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance  
ID"
```

For information about the status of a cancelled command, see [Monitoring Commands \(p. 394\)](#).

Monitoring Commands

There are several ways to monitor the status of the commands you send with Run Command:

- Configure CloudWatch Events to log status changes.
- Configure Amazon SNS to send notifications for all status changes or specific statuses like Failed or TimedOut.
- Click the Refresh icon on the **Command History** page in the Amazon EC2 console.
- Call the `ListCommand` or `ListCommandInvocation` API using a command line tool.

This section includes information about how to automate monitoring by using CloudWatch Events and SNS notifications.

When you send a command to multiple instances at the same time, each copy of the command targeting each instance is a *command invocation*. For example, if you use the AWS-RunShellScript JSON document and send an **ifconfig** command to 20 instances, that command has 20 invocations. Run Command monitoring options enable you to examine the details of each invocation, which can help you troubleshoot problems if a command fails.

Command Status

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Monitoring Commands**

Each command invocation individually reports status. Furthermore, each SSM document consists of one or more code-execution blocks called *plugins*. The plugins for a given command invocation individually report status as well. The following table describes Run Command execution statuses.

For more information about plugins, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

Status	Details
Pending	The command was not yet received by the SSM agent. If the command is not received by the agent before the value specified by the Timeout (seconds) parameter is reached, then the status changes to Timed Out .
In Progress	The command was received by the SSM agent, or the command started executing on the instance. Depending on the result of all command plugins, the status will change to Success , Failed , or Timed Out . If the SSM agent is not available on the instance, the command status will show In Progress until the agent is available again. The status will then change to a terminal state.
Success	The command, including all plugins, finished executing on the instance.
Failed	One or more plugins failed to execute. To help you identify and troubleshoot problems with a command, if a plugin fails, the entire invocation status is changed to Failed .
Timed Out	The command did not reach the SSM agent on the instance before the value specified by the Timeout (seconds) parameter was reached. Or, the command execution ran longer than the allowable execution time. The amount of time a command execution can run varies by plugin. Default values range from two hours to eight hours.
Cancelling	You sent the command to cancel a command invocation. The command has not reached a terminal state (Success , Failed , or Timed Out). If the SSM agent responds within the time-frame specified by the Timeout (seconds) parameter, then the status will change to reflect a terminal state. Otherwise the status will change to Timed Out .
Cancelled	The command was successfully cancelled.

Note

In the Amazon EC2 console, click the Refresh symbol in the upper-right corner to monitor command status as command execution progresses.

Contents

- [Getting Amazon SNS Notifications When a Command Changes Status \(p. 396\)](#)
- [Log Command Execution Status Changes for Run Command \(p. 401\)](#)

Getting Amazon SNS Notifications When a Command Changes Status

You can configure Amazon Simple Notification Service (Amazon SNS) to send notifications about the status of commands you send using Amazon EC2 Run Command. Amazon SNS coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can receive a notification whenever a command changes to a new state or changes to a specific state, such as failed or timed out. In cases where you send a command to multiple instances, you can receive a notification for each copy of the command sent to a specific instance. Each copy is called an *invocation*.

Amazon SNS can deliver notifications as HTTP or HTTPS POST, email (SMTP, either plain-text or in JSON format), or as a message posted to an Amazon Simple Queue Service (Amazon SQS) queue. For more information, see [What Is Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

For example, if you configure Amazon SNS to send a notification when a command status changes to failed, SNS sends an email notification with the details of the command execution.

Note

If you prefer, you can use Amazon CloudWatch Events to configure a target to invoke an AWS Lambda function when a command changes status. For more information, see [Log Command Execution Status Changes for Run Command \(p. 401\)](#).

To set up Amazon SNS notifications when a command changes status, you must complete the following tasks.

1. [Configure Account Permissions \(p. 398\)](#)
2. [Create an IAM Role for Notifications \(p. 398\)](#)
3. [Configure Amazon SNS \(p. 399\)](#)
4. [Send a Command that Returns Status Notifications \(p. 400\)](#)

Configure Amazon SNS Notifications for SSM

Run Command supports sending Amazon SNS notifications for commands that enter the following statuses. For information about the conditions that cause a command to enter one of these statuses, see [Monitoring Commands \(p. 394\)](#).

- In Progress
- Success
- Failed
- Timed Out
- Canceled

Note

Commands sent using Run Command also report Cancelling and Pending status. These statuses are not captured by SNS notifications.

If you configure Run Command for SNS notifications, SNS sends summary messages that include the following information:

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Monitoring Commands**

Field	Type	Description
EventTime	String	The time the event was triggered. The time stamp is important because SNS does not guarantee message delivery order. Example: 2016-04-26T13:15:30Z
DocumentName	String	The name of the SSM document used to execute this command.
CommandId	String	The ID generated by Run Command after the command was sent.
ExpiresAfter	Date	If this time is reached and the command has not already started executing, it will not execute.
OutputS3BucketName	String	The Amazon Simple Storage Service (Amazon S3) bucket where the responses to the command execution should be stored.
OutputS3KeyPrefix	String	The Amazon S3 directory path inside the bucket where the responses to the command execution should be stored.
RequestedDateTime	String	The time and date the request was sent to this specific instance.
InstanceId	String	The instance targeted by the command.
Status	String	Command status for the command.

If you send a command to multiple instances, Amazon SNS can send messages about each copy or invocation of the command that include the following information:

Field	Type	Description
EventTime	String	The time the event was triggered. The time stamp is important because SNS does not guarantee message delivery order. Example: 2016-04-26T13:15:30Z
DocumentName	String	The name of the SSM document used to execute this command.

Amazon Elastic Compute Cloud User Guide for Linux Instances Monitoring Commands

Field	Type	Description
RequestedDateTime	String	The time and date the request was sent to this specific instance.
CommandId	String	The ID generated by Run Command after the command was sent.
InstanceId	String	The instance targeted by the command.
Status	String	Command status for this invocation.

Configure Account Permissions

When you send a command that is configured for notifications, you specify a service role Amazon Resource Name (ARN). For example: `--service-role-arn=arn:aws:iam::123456789012:myrole`. This service role is used by SSM to trigger SNS notifications.

To receive notifications from the Amazon SNS service, you must either attach the `iam:PassRole` policy to your existing AWS Identity and Access Management (IAM) user account, or create a new IAM account and attach this policy to it. If you create a new account, you must also attach the `AmazonSSMFullAccess` policy so the account can communicate with the SSM API.

Use the following procedure to attach an IAM policy to your user account. If you need to create a new user account, see [Creating an IAM User in Your AWS Account](#) in the *IAM User Guide*.

To attach the `iam:PassRole` policy to your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane at the left, choose **Users** and then double-click your user account.
3. At the top of the page, copy your **User ARN** to the clipboard.
4. In the **Managed Policies** section, verify that either the `AmazonSSMFullAccess` policy is listed or there is a comparable policy that gives you permission to the SSM API.
5. Choose the down arrow beside **Inline Policies**, and then choose **click here**.
6. On the **Set Permissions** page, choose **Policy Generator**, and then choose **Select**.
7. Verify that **Effect** is set to **Allow**.
8. From **AWS Services** choose **AWS Identity and Access Management**.
9. From **Actions** choose **PassRole**.
10. In the **Amazon Resource Name (ARN)** field, paste your ARN.
11. Choose **Add Statement**, and then choose **Next**.
12. On the **Review Policy** page, choose **Apply Policy**.

Create an IAM Role for Notifications

In the previous procedure, you added an IAM policy to your user account so that you could send commands that return notifications. In the following procedure, you will create a role so that the SSM service can act on your behalf when sending notifications.

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. In **Step 1: Set Role Name** enter a name that identifies this role as a Run Command role for notifications.
4. In **Step 2: Select Role Type** choose **Amazon EC2**. The system skips **Step 3: Establish Trust** because this is a managed policy.
5. In **Step 4: Attach Policy** choose **AmazonSNSFullAccess**.
6. Choose **Next Step** and then choose **Create Role**. The system returns you to the **Roles** page.
7. Locate the role you just created and double-click it.
8. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**.
9. Add "ssm.amazonaws.com" to the existing policy as the following code snippet illustrates:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com",
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Note

You must add a comma after the existing entry. "Service": "sns.amazonaws.com", or the JSON will not validate.

10. Choose **Update Trust Policy**.
11. Copy or make a note of the **Role ARN**. You will specify this ARN when you send a command that is configured to return notifications.

Configure Amazon SNS

To use Amazon SNS to send email notifications, you must first create a *topic* and then subscribe your email addresses to the topic.

Create an Amazon SNS Topic

An Amazon SNS topic is a logical access point, a communication channel that Run Command uses to send the notifications. You create a topic by specifying a name for your topic.

For more information, see [Create a Topic](#) in the *Amazon Simple Notification Service Developer Guide*.

Note

After you create the topic, copy or make a note of the **Topic ARN**. You will specify this ARN when you send a command that is configured to return status notifications.

Subscribe to the Amazon SNS Topic

To receive the notifications that Run Command sends to the topic, you must subscribe an endpoint to the topic. In this procedure, for **Endpoint**, specify the email address where you want to receive the notifications from Run Command.

For more information, see [Subscribe to a Topic](#) in the *Amazon Simple Notification Service Developer Guide*.

Confirm Your Amazon SNS Subscription

Amazon SNS sends a confirmation email to the email address that you specified in the previous step.

Make sure you open the email from AWS Notifications and choose the link to confirm the subscription before you continue with the next step.

You will receive an acknowledgement message from AWS. Amazon SNS is now configured to receive notifications and send the notification as an email to the email address that you specified.

Send a Command that Returns Status Notifications

This section shows you how to send a command that is configured to return status notifications using either the Amazon EC2 console or the AWS Command Line Interface (AWS CLI).

To send a command from the Amazon EC2 console that returns notifications

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose an SSM document.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Amazon EC2 Run Command Prerequisites](#) (p. 355).
5. Enter information in the fields required by the SSM document. In the **SNS Notifications** section, choose **Enable SNS notifications**.
6. In the **Role** field, type or paste the IAM role ARN you created earlier.
7. In the **SNS Topic** field, type or paste the Amazon SNS ARN you created earlier.
8. In the **Notify me on** field, choose the events for which you want to receive notifications.
9. In the **Notify me for** field, choose to receive notifications for each copy of a command sent to multiple instances (invocations) or the command summary.
10. Choose **Run**.
11. Check your email for a message from Amazon SNS and open the email. Amazon SNS can take a few minutes to send the mail.

To send a command that is configured for notifications from the AWS CLI

1. Open the AWS CLI.
2. Specify parameters in the following command.

```
aws ssm send-command --instance-ids "ID-1, ID-2" --document-name "name"  
  --parameters commands=date --service-role ServiceRole ARN --notification-  
  config NotificationArn=SNS ARN
```

For example

```
aws ssm send-command --instance-ids "i-12345678, i-34567890" --
document-name "AWS-RunPowerShellScript" --parameters commands=date --
service-role arn:aws-cn:iam::123456789012:myrole --notification-config
NotificationArn=arn:aws-cn:sns:cn-north-1:123456789012:test
```

3. Press Enter.
4. Check your email for a message from Amazon SNS and open the email. Amazon SNS can take a few minutes to send the mail.

For more information about configuring Run Command from the command line, see [Amazon EC2 Simple Systems Manager API Reference](#) and the [SSM AWS CLI Reference](#).

Log Command Execution Status Changes for Run Command

You can use Amazon CloudWatch Events and a simple AWS Lambda function to log command execution status changes. You can create a rule that runs whenever there is a state transition, or when there is a transition to one or more states that are of interest.

Amazon EC2 Simple Systems Manager Event Types

SSM sends the following data to CloudWatch Events.

Example 1—EC2 Command Status-change Notification: This example includes information about execution status changes for a command that was sent to multiple instances.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Run Command - Command Status change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-03-14T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345670",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345679"
  ],
  "detail": {
    "command-id": "aws.ssm.12345678-1234-1234-1234-12345678",
    "requested-date-time": "2016-03-14T18:43:48Z",
    "expire-after": "2016-03-14T18:43:48Z",
    "output-s3bucket-name": "mybucket",
    "output-s3key-prefix": "test",
    "parameters": "parameter",
    "status": "Success"
  }
}
```

Example 2—EC2 Command Invocation Status-change Notification: This example includes information about a command that was sent to multiple instances, but the event shows details for only one instance, or *invocation* of that command.


```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Run Command - Command Invocation Status change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-03-14T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678"
  ],
  "detail": {
    "command-id": "aws.ssm.12345678-1234-1234-1234-12345678",
    "instance-id": "i-12345678",
    "requested-date-time": "2016-03-14T18:43:48Z",
    "status": "Success"
  }
}
```

Log SSM Command Execution Status Changes

In the following example scenario, you will create a simple AWS Lambda function, route events from SSM to it, and then test your scenario to ensure that it's set up correctly.

To log command execution status changes for Run Command, you must do the following.

1. [Step 1: Create an AWS Lambda Function \(p. 402\)](#)
2. [Step 2: Route Events to Your AWS Lambda Function \(p. 403\)](#)
3. [Step 3: Test Your Amazon CloudWatch Events Rule \(p. 403\)](#)

Step 1: Create an AWS Lambda Function

To create an AWS Lambda function

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. Choose **Create a Lambda function**, and then on the **Select blueprint** screen, choose **hello-world**.
3. On the **Configure function** screen, in the **Name** field, type a name for the event. This example uses **SomethingHappened**.
4. In the **Lambda function code** section, edit the sample code to match the following example:

```
console.log('Loading function');

exports.handler = function(event, context, callback) {
  console.log('SomethingHappened()');
  console.log('Here is the event:', JSON.stringify(event, null, 2));
  callback(null, "Ready");
};
```

5. Under **Lambda function handler and role**, in the **Role** field, if you have a **lambda_basic_execution_rule**, select it. Otherwise, create a new basic execution role.
6. Choose **Next**, and then on the **Review** screen, choose **Edit** to make any changes. If you're satisfied with the function, choose **Create function**.

Step 2: Route Events to Your AWS Lambda Function

To create a CloudWatch Events rule

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Events**.
3. Choose **Create rule**, and then under **Event selector**, choose **EC2 instance state-change notification**.
4. Choose **Specific state(s)**, and then **Running** from the list.
5. Do one of the following:
 - To make the rule respond to any of your instances in the region, choose **Any instance**.
 - To make the rule respond to a specific instance, choose **Specific instance(s)** and then in the text box, enter the instance ID.
6. Under **Targets**, choose **Add target**. In the **Select target type** list, choose **AWS Lambda function**.
7. In the **Function** list, select the function that you created in "**Step 1: Create an AWS Lambda Function**."
8. Choose **Configure input**, and then choose one of the following options:
 - **Matched event**
 - Sends all of the data fields in the event to CloudWatch Logs.
 - **Part of the matched event**
 - Sends only the specified data field of the event to CloudWatch Logs. You specify the part of the event using a string formatted `$.first_parameter.second_parameter`
 - For example, to send just the Amazon EC2 instance ID, type `$.detail.state` in the field.
 - **Constant**
 - Sends a JSON-formatted text string that you specify to CloudWatch Logs. For example, to send a text string for the event, type `{"Name":"MyInstance"}`. The constant must be valid JSON.
9. Choose **Configure details**. On the **Configure rule details** screen, in the **Name** field, type a name for the rule.
10. In the **Description** field, type a brief description for your rule, for example, **Log command execution status changes**.
11. If you're satisfied with the rule, choose **Create rule**.

Step 3: Test Your Amazon CloudWatch Events Rule

You can test your rule by executing a command with Run Command. After waiting a few minutes for the command to process, check your AWS Lambda metrics in the Amazon CloudWatch Events console to verify that your function was invoked.

To test your CloudWatch Events rule using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane choose **Command History**, and then execute a command to one or more instances. For more information about executing a command, see [Executing a Command Using Amazon EC2 Run Command \(p. 374\)](#).
3. To view your AWS Lambda metrics, open the CloudWatch console <https://console.aws.amazon.com/cloudwatch/>.
4. In the navigation pane, under **Metrics**, choose **Lambda** to view the metrics generated by your Lambda function.

5. To view the output from your function, in the navigation pane, choose **Logs**, and then in the **Log Groups** list, select the `/aws/lambda` log group that contains the data.
6. Under **Log Streams**, select a log stream to view the data about command execution status changes.

Troubleshooting Amazon EC2 Run Command

Use the following information to help troubleshoot problems with Run Command. For information about troubleshooting Run Command for Windows, see [Troubleshooting Run Command](#) in the User Guide for Windows.

Where Are My Instances?

If you do not see the expected list of instances when you choose **Select Target instances** then verify that your instance is configured with an AWS Identity and Access Management (IAM) role that enables the instance to communicate with the SSM API. Also verify that your user account has an IAM user trust policy that enables your account to communicate with the SSM API. The following procedures describe how to configure the instance role and the user trust policy.

Note

You must assign the IAM instance role when you *create* a new instance. You can't assign a role to an instance that is already running. To configure an existing instance to use an SSM-supported role, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Linux instance.
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Run Command role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
6. In the EC2 Management Console, choose the **Refresh** button beside **Create New IAM role**.
7. In the **IAM role** drop-down list, choose the role you just created.
8. Complete the wizard to create and launch the new instance.

Grant Your User Account Access to SSM

Use the following procedure to attach an the **AmazonSSMFullAccess** IAM policy to your user account. This policy grants you full access to SSM API actions.

To create an IAM policy for EC2 instances

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.

2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type AmazonSSMFullAccess and press Enter.
4. Select the checkbox next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.



5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

After you attach the policy, see if your instances are visible in the **Select Target instances** section of the EC2 console. If they are not visible, then one or more of the prerequisites have not been met. For more information, see [Amazon EC2 Run Command Prerequisites](#) (p. 355).

Check Instance Status Using the Health API

You can use the Amazon EC2 Health API to quickly determine the following information about Amazon EC2 instances:

- The version of the SSM agent
- The status of one or more instances
- The operating system
- The status of the SSM agent
- The last time the instance sent a heartbeat value

Use the following command to get status details about one or more instances:

```
aws ssm describe-instance-information --instance-information-filter-list  
key=InstanceIds,valueSet=instance-ID
```

Use the following command with no filters to see all instances registered to your account that are currently reporting an online status. Substitute the ValueSet="Online" with "ConnectionLost" or "Inactive" to view those statuses:

```
aws ssm describe-instance-information --instance-information-filter-list  
key=PingStatus,valueSet=Online
```

Use the following command to see which instances are running the latest version of the SSM agent. Substitute ValueSet="LATEST" with a specific version (for example, 1.0.145 or 1.0) to view those details:

```
aws ssm describe-instance-information --instance-information-filter-list  
key=AgentVersion,valueSet=LATEST
```

If the describe-instance-information API operation returns an AgentStatus of Online, then your instance is ready to be managed using Run Command. If the status is Inactive, the instance has one or more of the following problems.

- The SSM agent is not installed. For more information, see [Installing the SSM Agent \(p. 357\)](#).
- The instance does not have outbound internet connectivity.
- The instance was not launched with an IAM role that enables it to communicate with the SSM API, or the permissions for the IAM role are not correct for Run Command. For more information, see [Delegating Access to Amazon EC2 Run Command \(p. 365\)](#).

Troubleshooting the Amazon SSM Agent

If you experience problems executing commands using Run Command, there might be a problem with the SSM agent. Use the following information to help you troubleshoot the agent.

View Agent Logs

The SSM agent logs information in the following files using `cihub/seelog`. The information in these files can help you troubleshoot problems.

- `/var/log/amazon/ssm/amazon-ssm-agent.log`
- `/var/log/amazon/ssm/error.log`

You can enable extended logging by updating the `seelog.xml` file. By default, the configuration file is located here: `/opt/amazon/ssm/seelog.xml`.

For more information about `cihub/seelog` configuration, go to the [cihub/seelog Wiki](#). For examples of `cihub/seelog` configurations, go to [cihub/seelog examples](#).

Importing and Exporting Virtual Machines

You can use the Amazon Web Services (AWS) VM Import/Export tools to import virtual machine (VM) images from your local environment into AWS and convert them into ready-to-use Amazon EC2 Amazon machine images (AMIs) or instances. Later, you can export the VM images back to your local environment as an instance. VM Import/Export allows you to leverage your existing investments in the VMs that you have built to meet your IT security, configuration management, and compliance requirements by bringing those VMs into Amazon Elastic Compute Cloud (Amazon EC2) as ready-to-use AMIs or instances.

For more information, see [VM Import/Export](#) and the [VM Import/Export User Guide](#).

Monitoring Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources you will monitor?
- How often you will monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you've collected. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your Amazon EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
CPU utilization	CPUUtilization (p. 421)	
Memory utilization		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
Memory used		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Memory available		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Network utilization	NetworkIn (p. 421) NetworkOut (p. 421)	
Disk performance	DiskReadOps (p. 421) DiskWriteOps (p. 421)	
Disk Swap utilization (Linux instances only) Swap used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Page File utilization (Windows instances only) Page File used (Windows instances only) Page File available (Windows instances only)		Sending Performance Counters to CW; and Logs to CloudWatch Logs
Disk Reads/Writes	DiskReadBytes (p. 421) DiskWriteBytes (p. 421)	
Disk Space utilization (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space available (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

Automated and Manual Monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Topics

- [Automated Monitoring Tools](#) (p. 409)
- [Manual Monitoring Tools](#) (p. 410)

Automated Monitoring Tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System Status Checks** - monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
 - Loss of network connectivity
 - Loss of system power
 - Software issues on the physical host
 - Hardware issues on the physical host

For more information, see [Status Checks for Your Instances](#) (p. 411).

- **Instance Status Checks** - monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
 - Failed system status checks
 - Misconfigured networking or startup configuration
 - Exhausted memory
 - Corrupted file system
 - Incompatible kernel

For more information, see [Status Checks for Your Instances](#) (p. 411).

- **Amazon CloudWatch Alarms** - watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state, the state must have changed and been maintained for a specified number of periods. For more information, see [Monitoring Your Instances Using CloudWatch](#) (p. 419).
- **Amazon CloudWatch Logs** - monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see [Monitoring Log Files](#).
- **Amazon EC2 Monitoring Scripts** - Perl scripts that can monitor memory, disk, and swap file usage in your instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).
- **AWS Management Pack for Microsoft System Center Operations Manager** - links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS

Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see [AWS Management Pack for Microsoft System Center](#).

Manual Monitoring Tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
 - Service Health and Scheduled Events by region
 - Instance state
 - Status checks
 - Alarm status
 - Instance metric details (In the navigation pane click **Instances**, select an instance, and then click the **Monitoring** tab)
 - Volume metric details (In the navigation pane click **Volumes**, select a volume, and then click the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best Practices for Monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.
- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
 - What are your goals for monitoring?
 - What resources you will monitor?
 - How often you will monitor these resources?
 - What monitoring tools will you use?
 - Who will perform the monitoring tasks?
 - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances

Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

Contents

- [Status Checks for Your Instances \(p. 411\)](#)
- [Scheduled Events for Your Instances \(p. 415\)](#)

Status Checks for Your Instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. This data augments the information that Amazon EC2 already provides about the intended state of each instance (such as `pending`, `running`, `stopping`) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see [Creating and Editing Status Check Alarms \(p. 414\)](#).

Contents

- [Types of Status Checks \(p. 411\)](#)
- [Viewing Status Checks \(p. 412\)](#)
- [Reporting Instance Status \(p. 413\)](#)
- [Creating and Editing Status Check Alarms \(p. 414\)](#)

Types of Status Checks

There are two types of status checks: system status checks and instance status checks.

System Status Checks

Monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).

The following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power

- Software issues on the physical host
- Hardware issues on the physical host

Instance Status Checks

Monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

Viewing Status Checks

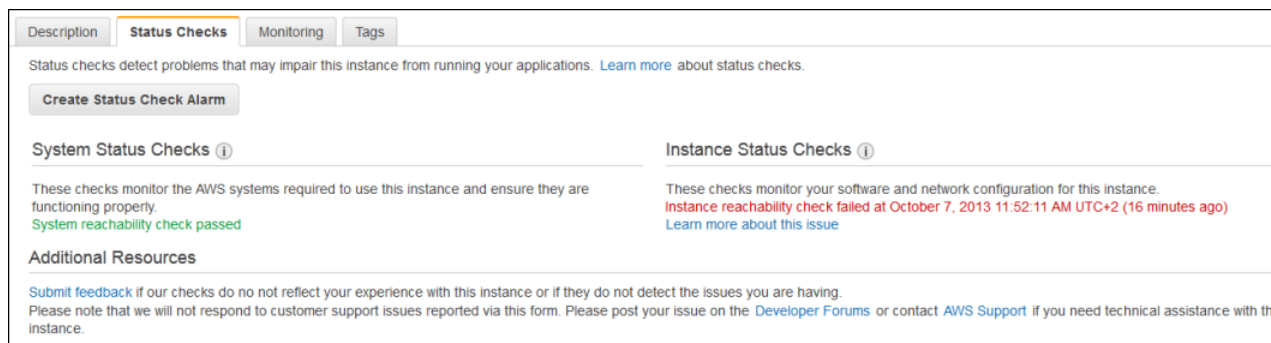
Amazon EC2 provides you with several ways to view and work with status checks.

Viewing Status Using the Console

You can view status checks using the AWS Management Console.

To view status checks using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.



5. If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, choose **AWS Support** to submit a request for assistance. To troubleshoot system or instance status check failures yourself, see [Troubleshooting Instances with Failed Status Checks](#) (p. 782).

Viewing Status Using the Command Line or API

You can view status checks for running instances using the `describe-instance-status` (AWS CLI) command.

To view the status of all instances, use the following command:

```
aws ec2 describe-instance-status
```

To get the status of all instances with a instance status of `impaired`:

```
aws ec2 describe-instance-status --filters Name=instance-  
status.status,Values=impaired
```

To get the status of a single instance, use the following command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:

- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceStatus](#) (Amazon EC2 Query API)

If you have an instance with a failed status check, see [Troubleshooting Instances with Failed Status Checks](#) (p. 782).

Reporting Instance Status

You can provide feedback if you are having problems with an instance whose status is not shown as `impaired`, or want to send AWS additional details about the problems you are experiencing with an `impaired` instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Reporting Status Feedback Using the Console

To report instance status using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Submit feedback**.
5. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting Status Feedback Using the Command Line or API

Use the following [report-instance-status](#) (AWS CLI) command to send feedback about the status of an `impaired` instance:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status  
impaired --reason-codes code
```

Alternatively, use the following commands:

- [Send-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [ReportInstanceStatus](#) (Amazon EC2 Query API)

Creating and Editing Status Check Alarms

You can create instance status and system status alarms to notify you when an instance has a failed status check.

Creating a Status Check Alarm Using the Console

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email or stop, terminate, or recover an instance when it fails an instance status check or system status check.

To create a status check alarm

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Create Status Check Alarm**.
5. Select **Send a notification to**. Choose an existing SNS topic, or click **create topic** to create a new one. If creating a new topic, in **With these recipients**, enter your email address and the addresses of any additional recipients, separated by commas.
6. (Optional) Choose **Take the action**, and then select the action that you'd like to take.
7. In **Whenever**, select the status check that you want to be notified about.

Note

If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.

8. In **For at least**, set the number of periods you want to evaluate and in **consecutive periods**, select the evaluation period duration before triggering the alarm and sending an email.
9. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.
10. Choose **Create Alarm**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by clicking the link contained in that message. Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

To edit a status check alarm

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, select **CloudWatch Monitoring**, and then choose **Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, choose the name of the alarm.
5. In the **Edit Alarm** dialog box, make the desired changes, and then choose **Save**.

Creating a Status Check Alarm Using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The metric is `StatusCheckFailed`.

To create a status check alarm using the CLI

1. Select an existing SNS topic or create a new one. For more information, see [Using the AWS CLI with Amazon SNS](#) in the *AWS Command Line Interface User Guide*.
2. Use the following `list-metrics` command to view the available Amazon CloudWatch metrics for Amazon EC2:

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following `put-metric-alarm` command to create the alarm:

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

Note

- `--period` is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes.
- `--evaluation-periods` is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2.
- `--alarm-actions` is the list of actions to perform when this alarm is triggered. Each action is specified as an Amazon Resource Name (ARN). This example configures the alarm to send an email using Amazon SNS.

Scheduled Events for Your Instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event, with details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the [Account Settings](#) page.

Contents

- [Types of Scheduled Events](#) (p. 415)
- [Viewing Scheduled Events](#) (p. 416)
- [Working with Instances Scheduled for Retirement](#) (p. 417)
- [Working with Instances Scheduled for Reboot](#) (p. 418)
- [Working with Instances Scheduled for Maintenance](#) (p. 418)

Types of Scheduled Events

Amazon EC2 supports the following types of scheduled events for your instances:

- **Instance stop:** The instance will be stopped. When you start it again, it's migrated to a new host computer. Applies only to instances backed by Amazon EBS.
- **Instance retirement:** The instance will be stopped or terminated.

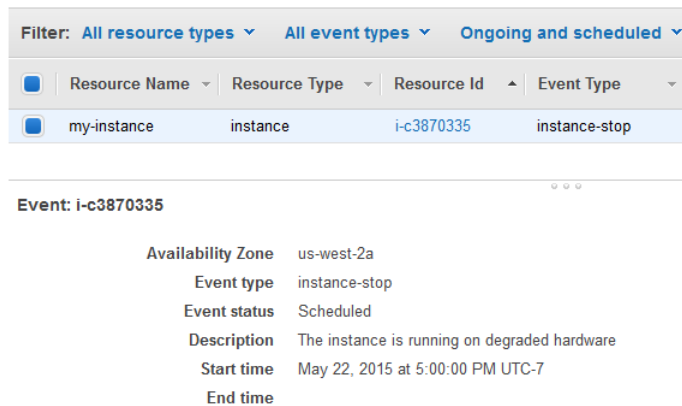
- **Reboot:** Either the instance will be rebooted (instance reboot) or the host computer for the instance will be rebooted (system reboot).
- **System maintenance:** The instance might be temporarily affected by network maintenance or power maintenance.

Viewing Scheduled Events

In addition to receiving notification of scheduled events in email, you can check for scheduled events.

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Events**. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.

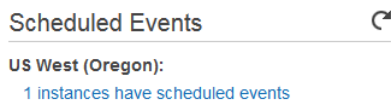



Filter:	All resource types	All event types	Ongoing and scheduled
Resource Name	Resource Type	Resource Id	Event Type
my-instance	instance	i-c3870335	instance-stop

Event: i-c3870335

Availability Zone	us-west-2a
Event type	instance-stop
Event status	Scheduled
Description	The instance is running on degraded hardware
Start time	May 22, 2015 at 5:00:00 PM UTC-7
End time	

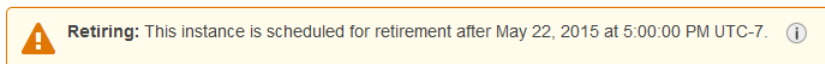
3. Alternatively, in the navigation pane, choose **EC2 Dashboard**. Any resources with an associated event are displayed under **Scheduled Events**.





Scheduled Events 

US West (Oregon):
1 instances have scheduled events

4. Note that events are also shown for affected resource. For example, in the navigation pane, choose **Instances**, and then select an instance. If the instance has an associated event, it is displayed in the lower pane.



 **Retiring:** This instance is scheduled for retirement after May 22, 2015 at 5:00:00 PM UTC-7. 

To view scheduled events for your instances using the command line or API

Use the following AWS CLI command:

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

The following is example output showing an instance retirement event:

```
{
```

```
"InstanceStatuses": [
  {
    "InstanceStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    },
    "AvailabilityZone": "us-west-2a",
    "InstanceId": "i-1234567890abcdef0",
    "InstanceState": {
      "Code": 16,
      "Name": "running"
    },
    "SystemStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    },
    "Events": [
      {
        "Code": "instance-stop",
        "Description": "The instance is running on degraded
hardware",
        "NotBefore": "2015-05-23T00:00:00.000Z"
      }
    ]
  }
]
```

Alternatively, use the following commands:

- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceStatus](#) (Amazon EC2 Query API)

Working with Instances Scheduled for Retirement

When AWS detects irreparable failure of the underlying host computer for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see [Instance Retirement](#) (p. 303).

Important

Any data stored on instance store volumes is lost when an instance is stopped or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you will need later before the instance is stopped or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance](#) (p. 299).

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Working with Instances Scheduled for Reboot

When AWS needs to perform tasks such as installing updates or maintaining the underlying host computer, it can schedule an instance or the underlying host computer for the instance for a reboot. Regardless of any existing instances that are scheduled for reboot, a new instance launch does not require a reboot, as the updates are already applied on the underlying host.

You can determine whether the reboot event is an instance reboot or a system reboot.

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Select **Instance resources** from the filter list, and then select your instance.
4. In the bottom pane, locate **Event type**. The value is either `system-reboot` or `instance-reboot`.

To view the type of scheduled reboot event using the AWS CLI

Use the following `describe-instance-status` command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Actions for Instance Reboot

You can wait for the reboot to occur within its scheduled maintenance window. Alternatively, you can reboot your instance yourself at a time that is convenient for you. For more information, see [Reboot Your Instance](#) (p. 302).

After you reboot your instance, the scheduled event for the instance reboot is canceled immediately and the event's description is updated. The pending maintenance to the underlying host computer is completed, and you can begin using your instance again after it has fully booted.

Actions for System Reboot

No action is required on your part; the system reboot occurs during its scheduled maintenance window. A system reboot typically completes in a matter of minutes. To verify that the reboot has occurred, check that there is no longer a scheduled event for the instance. We recommend that you check whether the software on your instance is operating as you expect.

Working with Instances Scheduled for Maintenance

When AWS needs to maintain the underlying host computer for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance will be restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for this instance status to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

Actions for Instances Backed by Amazon EBS

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance](#) (p. 299).

Actions for Instances Backed by Instance Store

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Monitoring Your Instances Using CloudWatch

You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see [Enable or Disable Detailed Monitoring for Your Instances](#) (p. 419).

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of the graphs in the console.

For more information about Amazon CloudWatch, see the [Amazon CloudWatch Developer Guide](#).

Contents

- [Enable or Disable Detailed Monitoring for Your Instances](#) (p. 419)
- [List the Available CloudWatch Metrics for Your Instances](#) (p. 421)
- [Get Statistics for Metrics for Your Instances](#) (p. 426)
- [Graph Metrics for Your Instances](#) (p. 441)
- [Create a CloudWatch Alarm](#) (p. 444)
- [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance](#) (p. 450)

Enable or Disable Detailed Monitoring for Your Instances

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. The following table describes basic and detailed monitoring for instances.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	Data is available in 1-minute periods for an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. For information about pricing, see the Amazon CloudWatch product page .

Enabling Detailed Monitoring

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To enable detailed monitoring for an existing instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Select the instance, click **Actions**, select **CloudWatch Monitoring**, and then click **Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, click **Yes, Enable**.
5. In the **Enable Detailed Monitoring** confirmation dialog box, click **Close**.

To enable detailed monitoring when launching an instance using the console

When launching an instance using the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page.

To enable detailed monitoring for an existing instance using the AWS CLI

Use the following `monitor-instances` command to enable detailed monitoring for the specified instances.

```
$ aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To enable detailed monitoring when launching an instance using the AWS CLI

Use the `run-instances` command with the `--monitoring` flag to enable detailed monitoring.

```
$ aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

Disabling Detailed Monitoring

You can disable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To disable detailed monitoring using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, click **Instances**.
3. Select the instance, click **Actions**, select **CloudWatch Monitoring**, and then click **Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, click **Yes, Disable**.
5. In the **Disable Detailed Monitoring** confirmation dialog box, click **Close**.

To disable detailed monitoring using the AWS CLI

Use the following `unmonitor-instances` command to disable detailed monitoring for the specified instances.

```
$ aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

List the Available CloudWatch Metrics for Your Instances

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the previous 5 minutes of activity for the instance. If you've enabled detailed monitoring, each data point covers the previous 1 minute of activity.

For information about getting the statistics for these metrics, see [Get Statistics for Metrics for Your Instances](#) (p. 426).

Instance Metrics

The following table describes the metrics for your instances. For information about the metrics provided for your EBS volumes, see [Amazon EBS Metrics](#) (p. 658). For information about the metrics provided for your Spot fleets, see [CloudWatch Metrics for Spot Fleet](#) (p. 251).

The `AWS/EC2` namespace includes the following metrics.

Metric	Description
CPUCreditUsage	<p>(Only valid for T2 instances) The number of CPU credits consumed during the specified period.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
CPUCreditBalance	<p>(Only valid for T2 instances) The number of CPU credits that an instance has accumulated.</p> <p>This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
List Available Metrics**

Metric	Description
CPUUtilization	<p>The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.</p> <p>Note Depending on the instance type, tools in your operating system may show a lower percentage than CloudWatch when the instance is not allocated a full processor core.</p> <p>Units: Percent</p>
DiskReadOps	<p>Completed read operations from all instance store volumes available to the instance in a specified period of time.</p> <p>Note To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>Units: Count</p>
DiskWriteOps	<p>Completed write operations to all instance store volumes available to the instance in a specified period of time.</p> <p>Note To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>Units: Count</p>
DiskReadBytes	<p>Bytes read from all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: Bytes</p>
DiskWriteBytes	<p>Bytes written to all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: Bytes</p>
NetworkIn	<p>The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance.</p> <p>Units: Bytes</p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
List Available Metrics**

Metric	Description
NetworkOut	<p>The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance.</p> <p>Units: Bytes</p>
NetworkPacketsIn	<p>The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>
NetworkPacketsOut	<p>The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>
StatusCheckFailed	<p>A combination of StatusCheckFailed_Instance and StatusCheckFailed_System that reports if either of the status checks has failed. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p>Note Status check metrics are available at 1 minute frequency. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state. Status check metrics become available within a few minutes of the instance being in the running state.</p> <p>Units: Count</p>
StatusCheckFailed_Instance	<p>Reports whether the instance has passed the Amazon EC2 instance status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p>Note Status check metrics are available at 1 minute frequency. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state. Status check metrics become available within a few minutes of the instance being in the running state.</p> <p>Units: Count</p>

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
List Available Metrics**

Metric	Description
StatusCheckFailed_System	<p>Reports whether the instance has passed the EC2 system status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p>Note Status check metrics are available at a 1 minute frequency. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state. Status check metrics become available within a few minutes of the instance being in the running state.</p> <p>Units: Count</p>

Amazon EC2 Dimensions

You can use the dimensions in the following table to refine the metrics returned for your instances.

Dimension	Description
AutoScalingGroupName	This dimension filters the data you request for all instances in a specified capacity group. An <i>Auto Scaling group</i> is a collection of instances you define if you're using Auto Scaling. This dimension is available only for Amazon EC2 metrics when the instances are in such an Auto Scaling group. Available for instances with Detailed or Basic Monitoring enabled.
ImageId	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
InstanceId	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
InstanceType	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

Listing Metrics Using the Console

You can view metrics by category. Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.

Amazon Elastic Compute Cloud User Guide for Linux Instances List Available Metrics

3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Per-Instance Metrics**.

CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: **1,014**

Billing Metrics : 35 Total Estimated Charge : 1 By Service : 13 By Linked Account : 3 By Linked Account and Service : 18	DynamoDB Metrics : 4 Table Metrics : 4	EBS Metrics : 80 Per-Volume Metrics : 80
EC2 Metrics : 272 Per-Instance Metrics : 181 By Auto Scaling Group : 56 By Image (AMI) Id : 14 Aggregated by Instance Type : 14 Across All Instances : 7	ELB Metrics : 95 Per-LB Metrics : 29 Per LB, per AZ Metrics : 37 By Availability Zone : 20 Across All LBs : 9	ElastiCache Metrics : 87 Cache Node Metrics : 87
OpsWorks Metrics : 45 Instance Metrics : 15 Layer Metrics : 15 Stack Metrics : 15	RDS Metrics : 104 Per-Database Metrics : 52 By Database Class : 26 By Database Engine : 13 Across All Databases : 13	Redshift Metrics : 26 Node Metrics : 13 Aggregated by Cluster : 13

5. In the upper pane, scroll down to view the full list of metrics.

EC2 > Per-Instance Metrics

<input type="checkbox"/>	InstanceId	InstanceName	Metric Name
<input type="checkbox"/>	i-0314eff14c446eb49		CPUUtilization
<input type="checkbox"/>	i-0314eff14c446eb49		DiskReadBytes
<input type="checkbox"/>	i-0314eff14c446eb49		DiskReadOps
<input type="checkbox"/>	i-0314eff14c446eb49		DiskWriteBytes

↑

Select a metric above to view graph

Click a checkbox to select a metric
Click on text to add to search

Update Graph

▼ Time Range

Relative Absolute UTC

From: hours ago

To: minutes ago

Zoom: 1h | 3h | 6h | 12h | 1d |

Listing Metrics Using the AWS CLI

Use the `list-metrics` command to list the CloudWatch metrics for your instances.

To list the available metrics across all instances

Specify the `AWS/EC2` namespace to limit the results to the metrics sent by Amazon EC2.

```
$ aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:


```
...
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-1234567890abcdef0"
      }
    ],
    "MetricName": "NetworkOut"
  },
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-1234567890abcdef0"
      }
    ],
    "MetricName": "CPUUtilization"
  },
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-1234567890abcdef0"
      }
    ],
    "MetricName": "NetworkIn"
  },
  ...
```

To list the available metrics for a specified instance

Specify the `AWS/EC2` namespace and the `InstanceId` dimension to limit the results to metrics sent by Amazon EC2 for the specified instance.

```
$ aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances

Specify the `AWS/EC2` namespace and the metric name to limit the results to the specified metric sent by Amazon EC2.

```
$ aws cloudwatch list-metrics --namespace AWS/EC2 --metric-
name CPUUtilization
```

Get Statistics for Metrics for Your Instances

The following scenarios demonstrate how to get statistics for the CloudWatch metrics for your instances.

Contents

- [Get Statistics for a Specific Instance \(p. 427\)](#)

- [Aggregate Statistics Across Instances](#) (p. 430)
- [Aggregate Statistics by Auto Scaling Group](#) (p. 434)
- [Aggregate Statistics by Image \(AMI\) ID](#) (p. 437)

Get Statistics for a Specific Instance

The following scenario walks you through how to use the AWS Management Console or the `get-metric-statistics` command to determine the maximum CPU utilization of a specific EC2 instance.

Note

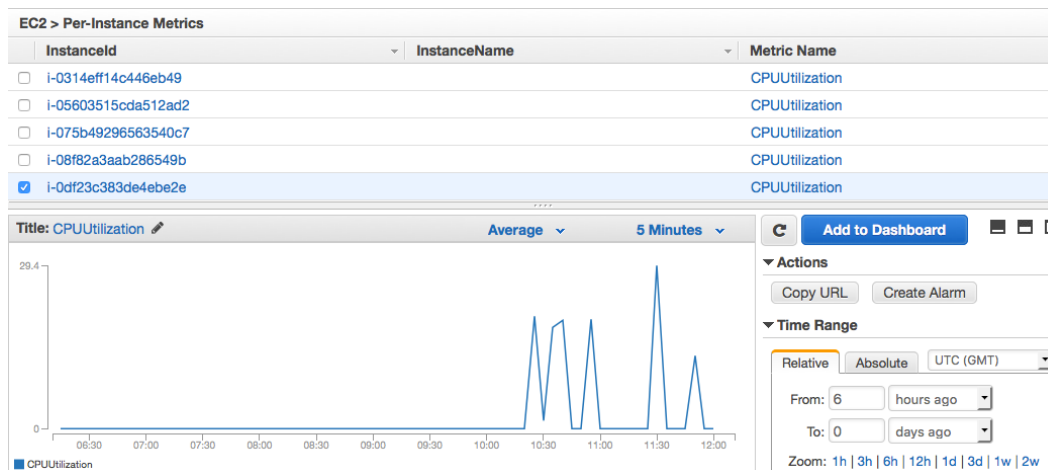
Start and end times must be within the last 14 days.

For this example, we assume that you have an EC2 instance ID. You can get an active EC2 instance ID through the AWS Management Console or with the `describe-instances` command.

AWS Management Console

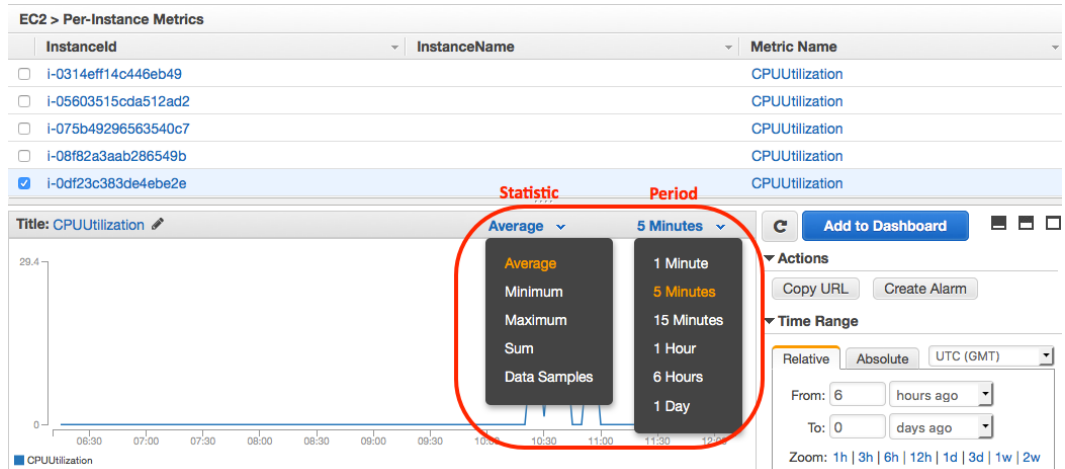
To display the CPU utilization for a specific instance using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, select **EC2: Metrics**.
5. Select a row that contains **CPUUtilization** for a specific instance, which displays a graph for this metric for the instance.



6. To change the statistic or the period for the metric, choose a different value from the list.

Amazon Elastic Compute Cloud User Guide for Linux Instances Get Statistics for Metrics



AWS Command Line Interface

To get the CPU utilization for a specific instance using the AWS CLI

Use the following `get-metric-statistics` command.

```
$ aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --period 3600 --statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --start-time 2014-02-18T23:18:00 --end-time 2014-02-19T23:18:00
```

The returned statistics are six-minute values for the requested two-day time interval. Each value represents the maximum CPU utilization percentage for a single EC2 instance. The following is example output:

```
{
  "Datapoints": [
    {
      "Timestamp": "2014-02-19T00:18:00Z",
      "Maximum": 0.33000000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-19T03:18:00Z",
      "Maximum": 99.670000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-19T07:18:00Z",
      "Maximum": 0.34000000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-19T12:18:00Z",
      "Maximum": 0.34000000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-19T02:18:00Z",

```

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Get Statistics for Metrics

```
"Maximum": 0.34000000000000002,  
"Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T01:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T17:18:00Z",  
  "Maximum": 3.3900000000000001,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T13:18:00Z",  
  "Maximum": 0.33000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-18T23:18:00Z",  
  "Maximum": 0.67000000000000004,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T06:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T11:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T10:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T19:18:00Z",  
  "Maximum": 8.0,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T15:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T14:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{  
  "Timestamp": "2014-02-19T16:18:00Z",  
  "Maximum": 0.34000000000000002,  
  "Unit": "Percent"  
},  
{
```

```
    "Timestamp": "2014-02-19T09:18:00Z",  
    "Maximum": 0.34000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T04:18:00Z",  
    "Maximum": 2.0,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T08:18:00Z",  
    "Maximum": 0.68000000000000005,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T05:18:00Z",  
    "Maximum": 0.33000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T18:18:00Z",  
    "Maximum": 6.669999999999999,  
    "Unit": "Percent"  
  }  
],  
"Label": "CPUUtilization"  
}
```

Aggregate Statistics Across Instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. This scenario shows you how to use detailed monitoring with either the AWS Management Console, the `GetMetricStatistics` API, or the `get-metric-statistics` command to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the `AWS/EC2` namespace. To get statistics for other metrics, see [Amazon CloudWatch Namespaces, Dimensions, and Metrics Reference](#).

Important

This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

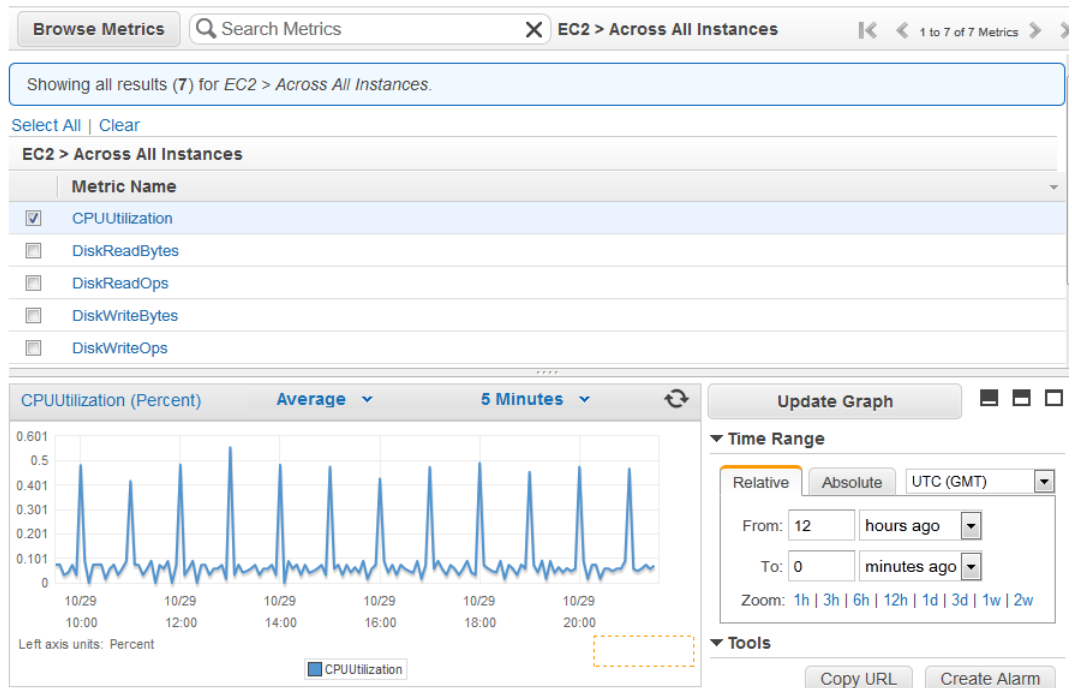
AWS Management Console

To display average CPU utilization across your instances

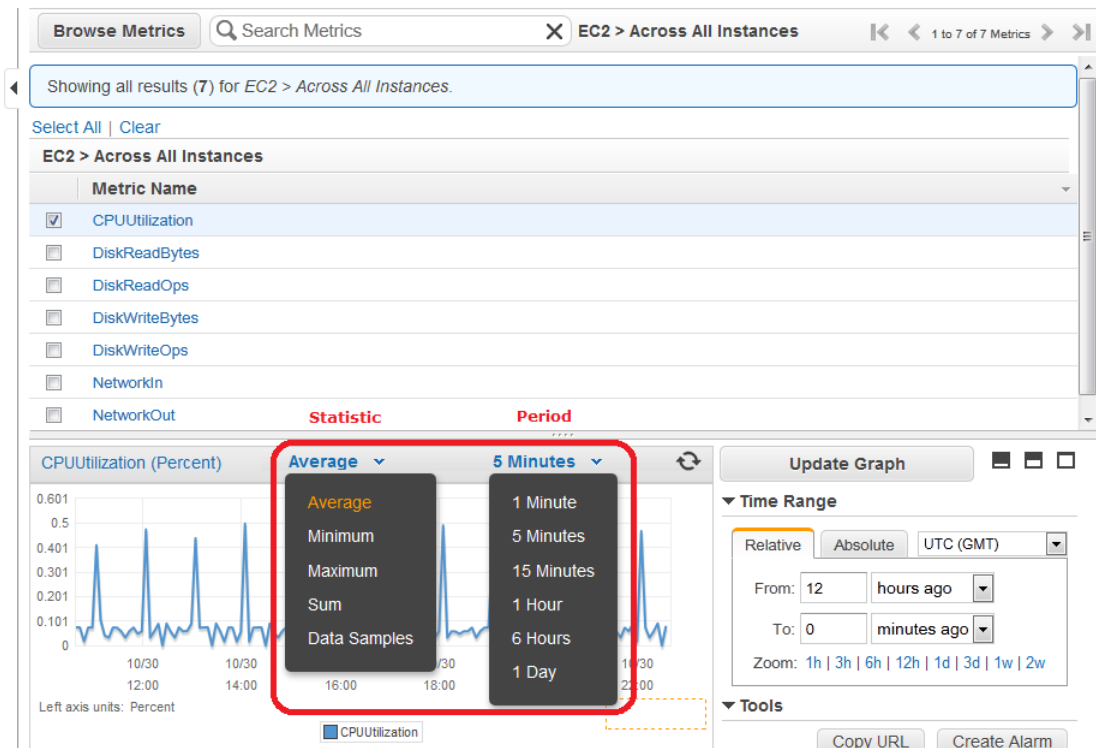
1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Across All Instances**.
5. In the upper pane, select the row that contains **CPUUtilization**.

Amazon Elastic Compute Cloud User Guide for Linux Instances Get Statistics for Metrics

A graph showing CPU utilization for your EC2 instances is displayed in the details pane.



- To change the statistic or period for the metric, choose a different value from the list.



AWS Command Line Interface

To get average CPU utilization across your instances

Use the `get-metric-statistics` command as follows.

```
$ aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
CPUUtilization --period 3600 --statistics "Average" "SampleCount" --start-  
time 2014-02-11T23:18:00 --end-time 2014-02-12T23:18:00
```

The following is example output:

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2014-02-12T07:18:00Z",  
      "Average": 0.038235294117647062,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T09:18:00Z",  
      "Average": 0.16670833333333332,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2014-02-11T23:18:00Z",  
      "Average": 0.041596638655462197,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T16:18:00Z",  
      "Average": 0.039458333333333345,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 239.0,  
      "Timestamp": "2014-02-12T21:18:00Z",  
      "Average": 0.041255230125523033,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T01:18:00Z",  
      "Average": 0.044583333333333336,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 239.0,  
      "Timestamp": "2014-02-12T18:18:00Z",  
      "Average": 0.043054393305439344,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Get Statistics for Metrics**

```
    "Timestamp": "2014-02-12T13:18:00Z",
    "Average": 0.039458333333333345,
    "Unit": "Percent"
  },
  {
    "SampleCount": 238.0,
    "Timestamp": "2014-02-12T15:18:00Z",
    "Average": 0.041260504201680689,
    "Unit": "Percent"
  },
  {
    "SampleCount": 240.0,
    "Timestamp": "2014-02-12T19:18:00Z",
    "Average": 0.037666666666666668,
    "Unit": "Percent"
  },
  {
    "SampleCount": 240.0,
    "Timestamp": "2014-02-12T06:18:00Z",
    "Average": 0.037541666666666675,
    "Unit": "Percent"
  },
  {
    "SampleCount": 240.0,
    "Timestamp": "2014-02-12T20:18:00Z",
    "Average": 0.039333333333333338,
    "Unit": "Percent"
  },
  {
    "SampleCount": 240.0,
    "Timestamp": "2014-02-12T08:18:00Z",
    "Average": 0.039250000000000014,
    "Unit": "Percent"
  },
  {
    "SampleCount": 239.0,
    "Timestamp": "2014-02-12T03:18:00Z",
    "Average": 0.037740585774058588,
    "Unit": "Percent"
  },
  {
    "SampleCount": 240.0,
    "Timestamp": "2014-02-12T11:18:00Z",
    "Average": 0.039500000000000007,
    "Unit": "Percent"
  },
  {
    "SampleCount": 238.0,
    "Timestamp": "2014-02-12T02:18:00Z",
    "Average": 0.039789915966386563,
    "Unit": "Percent"
  },
  {
    "SampleCount": 238.0,
    "Timestamp": "2014-02-12T22:18:00Z",
    "Average": 0.039705882352941181,
    "Unit": "Percent"
  },
  {

```



```
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T14:18:00Z",  
    "Average": 0.08245833333333328,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T05:18:00Z",  
    "Average": 0.04287500000000001,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T17:18:00Z",  
    "Average": 0.03945833333333345,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T10:18:00Z",  
    "Average": 0.08341666666666667,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 236.0,  
    "Timestamp": "2014-02-12T00:18:00Z",  
    "Average": 0.036567796610169498,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T12:18:00Z",  
    "Average": 0.03954166666666676,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T04:18:00Z",  
    "Average": 0.04300000000000003,  
    "Unit": "Percent"  
  }  
],  
  "Label": "CPUUtilization"  
}
```

Aggregate Statistics by Auto Scaling Group

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API with the `DiskWriteBytes` metric to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified `AutoScalingGroupName`.

Note

Start and end times must be within the last 14 days.

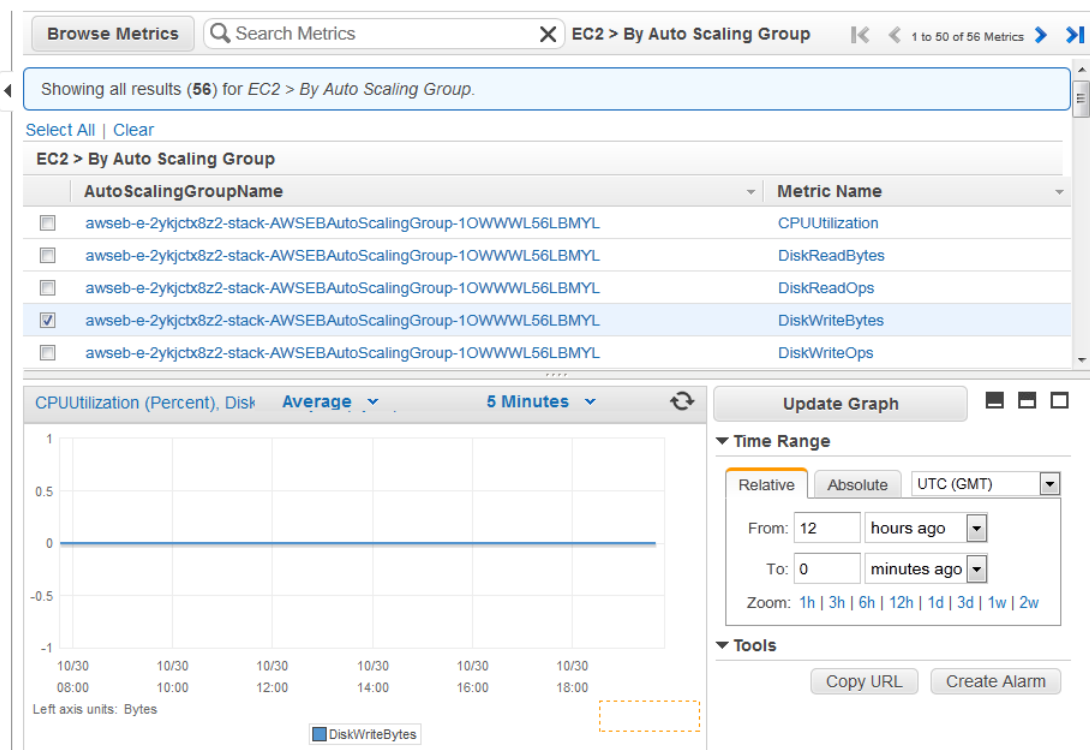
We assume for this example that an EC2 application is running and has an Auto Scaling group named `test-group-1`.

AWS Management Console

To display total `DiskWriteBytes` for an auto-scaled application

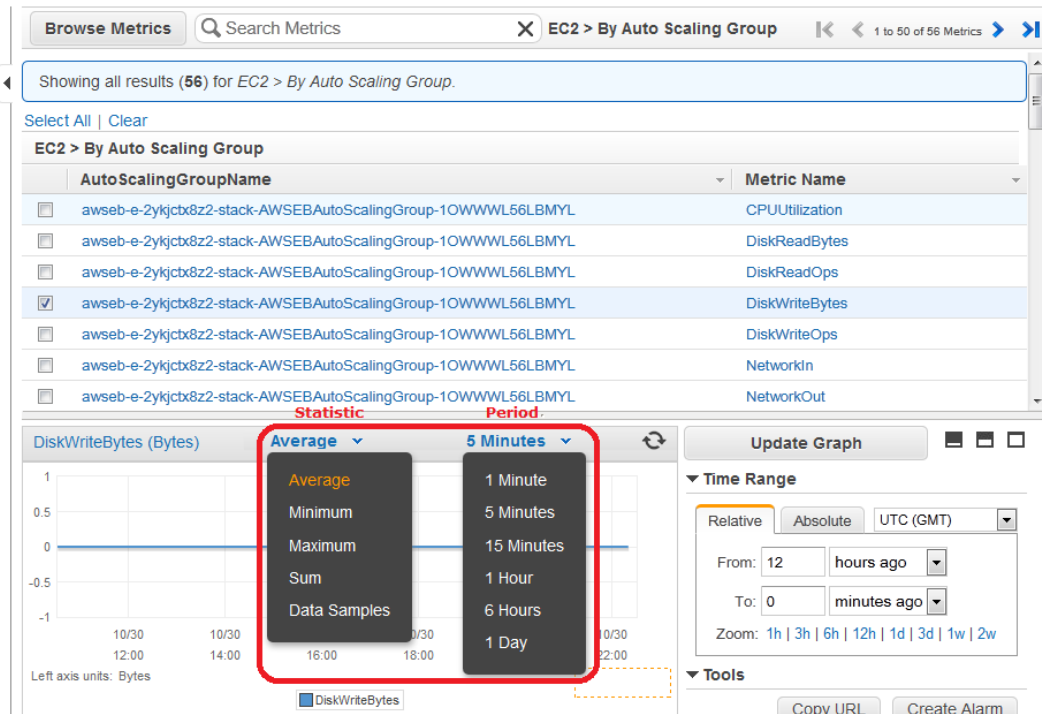
1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Auto Scaling Group**.
5. Select the row that contains **DiskWriteBytes**.

A graph showing `DiskWriteBytes` for all EC2 instances appears in the details pane.



6. To change the statistic or period for the metric, choose a different value from the list.

Amazon Elastic Compute Cloud User Guide for Linux Instances Get Statistics for Metrics



AWS Command Line Interface

To get total DiskWriteBytes for an auto-scaled application

Use the `get-metric-statistics` command as follows.

```
$ aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
DiskWriteBytes --period 360 --statistics "Sum" "SampleCount" --dimensions  
Name=AutoScalingGroupName,Value=my-group --start-time 2014-02-16T23:18:00 --  
end-time 2014-02-18T23:18:00
```

The following is example output:

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 18.0,  
      "Timestamp": "2014-02-19T21:36:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    },  
    {  
      "SampleCount": 5.0,  
      "Timestamp": "2014-02-19T21:42:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    }  
  ],  
  "Label": "DiskWriteBytes"  
}
```

Aggregate Statistics by Image (AMI) ID

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API to determine average CPU utilization for all instances that match a given image ID. The average is over 60-second time intervals for a one-day period.

Note

Start and end times must be within the last 14 days.

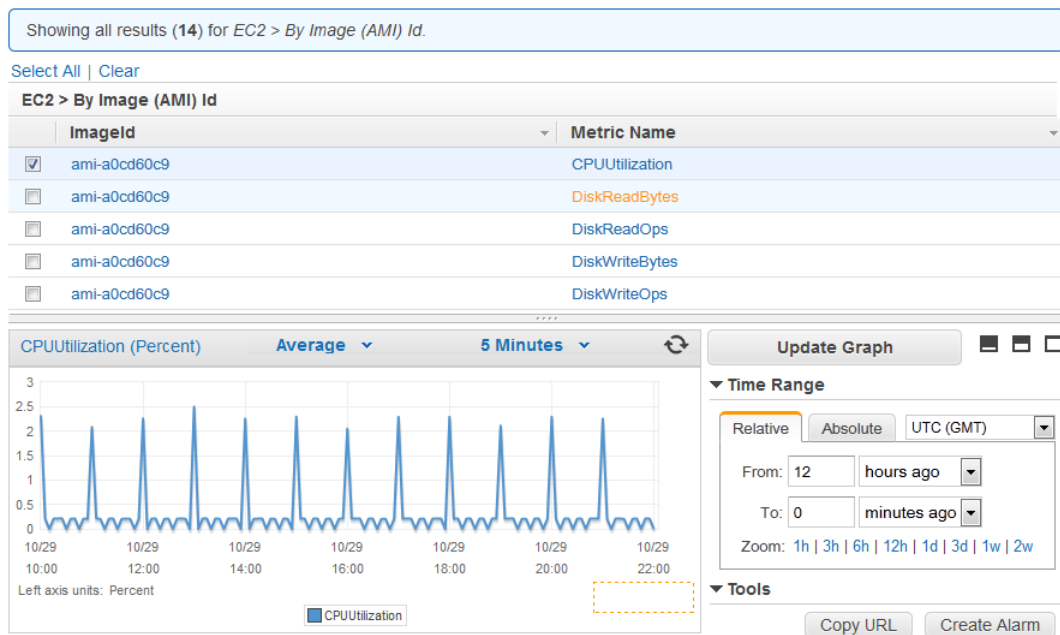
In this scenario, the EC2 instances are running an image ID of `ami-c5e40dac`.

AWS Management Console

To display the average CPU utilization for an image ID

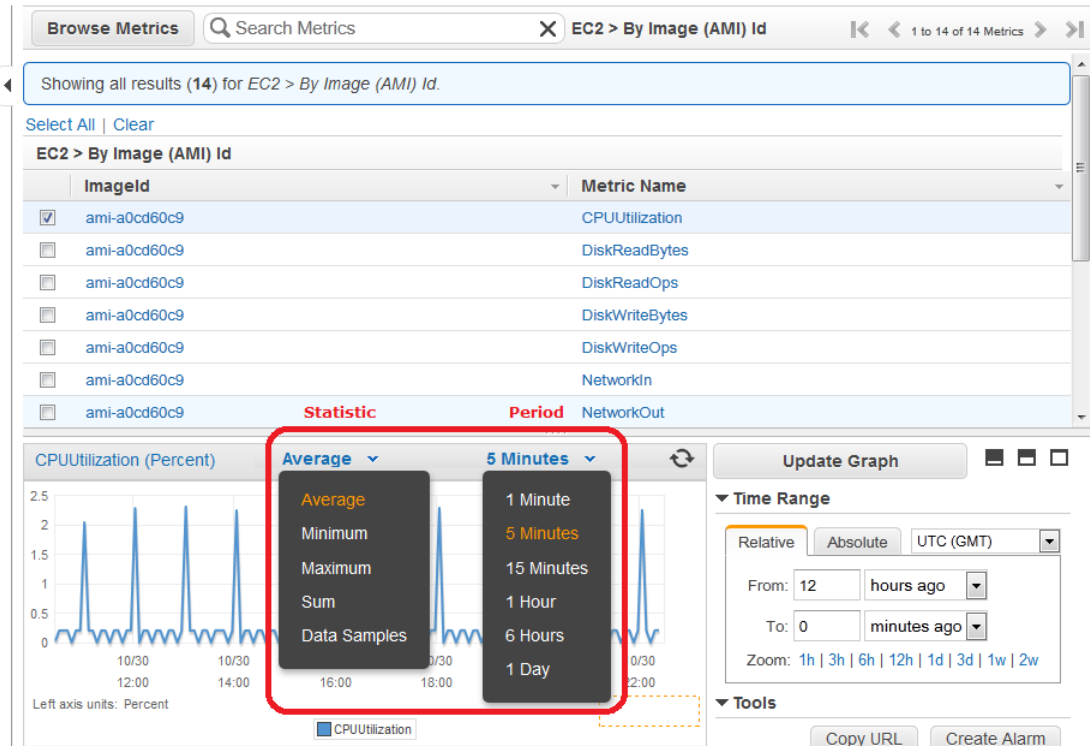
1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Image (AMI) Id**.
5. Select a row that contains **CPUUtilization** and an image ID.

A graph showing average `CPUUtilization` for all EC2 instances based on the `ami-c5e40dac` image ID appears in the details pane.



6. To change the statistic or period for the metric, choose a different value from the list.

Amazon Elastic Compute Cloud User Guide for Linux Instances Get Statistics for Metrics



AWS Command Line Interface

To get the average CPU utilization for an image ID

Use the `get-metric-statistics` command as follows.

```
$ aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --period 3600 --statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-time 2014-02-10T00:00:00 --end-time 2014-02-11T00:00:00
```

The following is example output:

```
{
  "Datapoints": [
    {
      "Timestamp": "2014-02-10T07:00:00Z",
      "Average": 0.041000000000000009,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T14:00:00Z",
      "Average": 0.079579831932773085,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T06:00:00Z",
      "Average": 0.0360000000000000011,
      "Unit": "Percent"
    }
  ]
}
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Get Statistics for Metrics**

```
    },
    {
      "Timestamp": "2014-02-10T13:00:00Z",
      "Average": 0.037625000000000013,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T18:00:00Z",
      "Average": 0.042750000000000003,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T21:00:00Z",
      "Average": 0.039705882352941188,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T20:00:00Z",
      "Average": 0.039375000000000007,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T02:00:00Z",
      "Average": 0.041041666666666671,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T01:00:00Z",
      "Average": 0.041083333333333354,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T23:00:00Z",
      "Average": 0.038016877637130804,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T15:00:00Z",
      "Average": 0.037666666666666668,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T12:00:00Z",
      "Average": 0.039291666666666676,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T03:00:00Z",
      "Average": 0.036000000000000004,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T04:00:00Z",
      "Average": 0.042666666666666672,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T19:00:00Z",
      "Average": 0.038305084745762719,
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Get Statistics for Metrics**

```
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T22:00:00Z",
        "Average": 0.039291666666666676,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T09:00:00Z",
        "Average": 0.17126050420168065,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T08:00:00Z",
        "Average": 0.041166666666666678,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T11:00:00Z",
        "Average": 0.082374999999999962,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T17:00:00Z",
        "Average": 0.037625000000000013,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T10:00:00Z",
        "Average": 0.039458333333333345,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T05:00:00Z",
        "Average": 0.039250000000000007,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T00:00:00Z",
        "Average": 0.037625000000000013,
        "Unit": "Percent"
      },
      {
        "Timestamp": "2014-02-10T16:00:00Z",
        "Average": 0.041512605042016815,
        "Unit": "Percent"
      }
    ],
    "Label": "CPUUtilization"
  }
}
```

The operation returns statistics that are one-minute values for the one-day interval. Each value represents an average CPU utilization percentage for EC2 instances running the specified machine image.

Graph Metrics for Your Instances

After you launch an instance, you can go to the Amazon EC2 console and view the instance's monitoring graphs. They're displayed when you select the instance on the **Instances** page in the EC2 Dashboard. A **Monitoring** tab is displayed next to the instance's **Description** tab. The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

Each graph is based on one of the available Amazon EC2 metrics. For more information about the metrics and the data they provide to the graphs, see [List the Available CloudWatch Metrics for Your Instances \(p. 421\)](#).

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services to make it easier to see what's going on. You can use the following procedures to graph metrics in CloudWatch.

Contents

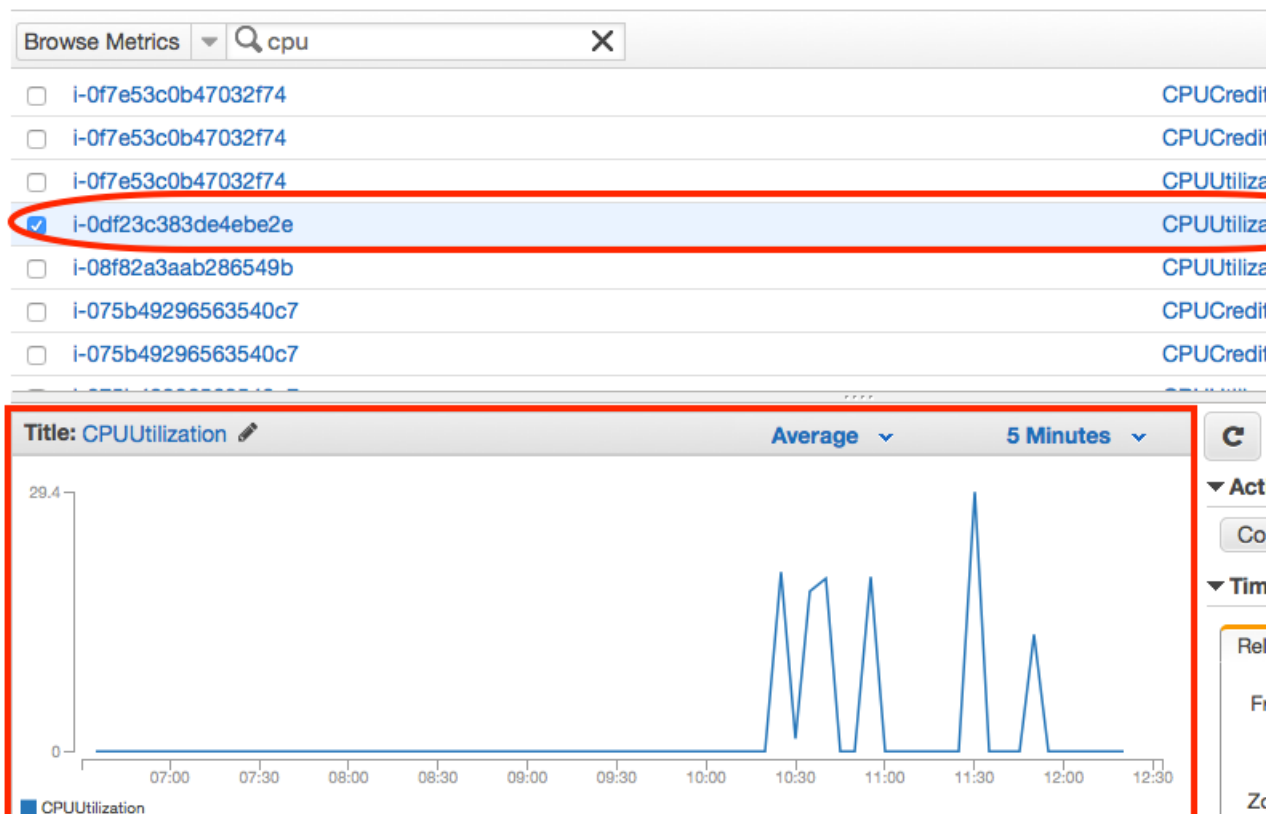
- [Graph a Metric \(p. 441\)](#)
- [Graph a Metric Across Resources \(p. 442\)](#)

Graph a Metric

You can select a metric and create a graph of the data in CloudWatch. For example, you can select the CPUUtilization metric for an Amazon EC2 instance and display a graph of CPU usage over time for that instance.

To graph a metric

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, use the **Search Metrics** box and categories to find a metric by metric name, AWS resource, or other metadata.
5. Use the scroll bar and next and previous arrows above the metrics list to page through the full list of metrics
6. Select the metric to view, for example, CPUUtilization. A graph appears in the details pane.



7. To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Graph a Metric Across Resources

You can graph a metric across all resources to see everything on one graph. For example, you can graph the CPUUtilization metric for all Amazon EC2 instances on one graph.

To graph a metric across resources

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Metrics**.

Amazon Elastic Compute Cloud User Guide for Linux Instances Graph Metrics

CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: **1,014**

Billing Metrics : 35 Total Estimated Charge : 1 By Service : 13 By Linked Account : 3 By Linked Account and Service : 18	DynamoDB Metrics : 4 Table Metrics : 4	EBS Metrics : 80 Per-Volume Metrics : 80
EC2 Metrics : 272 Per-Instance Metrics : 181 By Auto Scaling Group : 56 By Image (AMI) Id : 14 Aggregated by Instance Type : 14 Across All Instances : 7	ELB Metrics : 95 Per-LB Metrics : 29 Per LB, per AZ Metrics : 37 By Availability Zone : 20 Across All LBs : 9	ElastiCache Metrics : 87 Cache Node Metrics : 87
OpsWorks Metrics : 45 Instance Metrics : 15 Layer Metrics : 15 Stack Metrics : 15	RDS Metrics : 104 Per-Database Metrics : 52 By Database Class : 26 By Database Engine : 13 Across All Databases : 13	Redshift Metrics : 26 Node Metrics : 13 Aggregated by Cluster : 13

- In the **CloudWatch Metrics by Category** pane, select a metric category. For example, under **EC2 Metrics**, select **Per-Instance Metrics**.

EC2 > Per-Instance Metrics

	InstanceId	InstanceName	Metric Name
<input type="checkbox"/>	i-0314eff14c446eb49		CPUUtilization
<input type="checkbox"/>	i-0314eff14c446eb49		DiskReadBytes
<input type="checkbox"/>	i-0314eff14c446eb49		DiskReadOps
<input type="checkbox"/>	i-0314eff14c446eb49		DiskWriteBytes

Select a metric above to view graph

Click a checkbox to select a metric
Click on text to add to search

Update Graph ☰ ☱ ☲

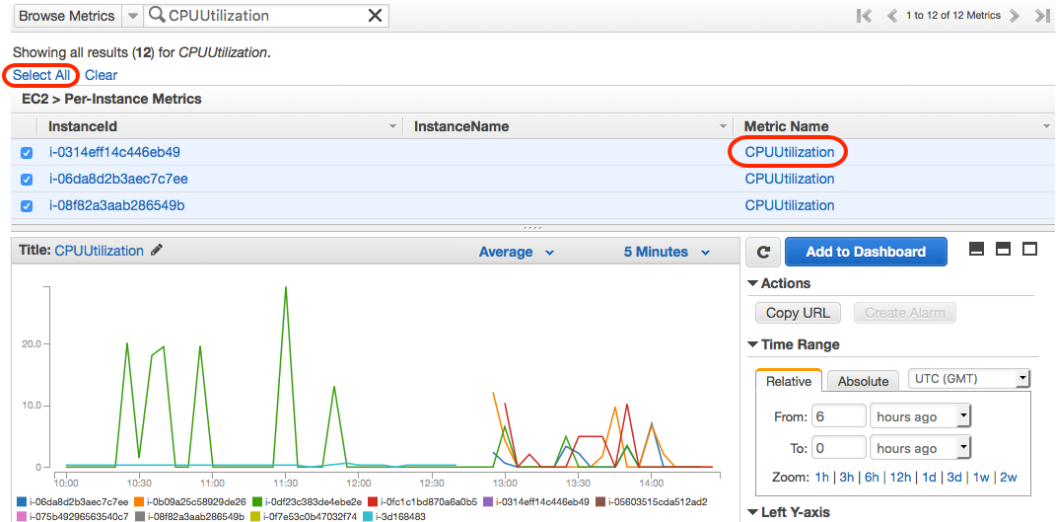
▼ Time Range

Relative Absolute UTC (GMT) ▼
 From: hours ago
 To: minutes ago
 Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

- In the metric list, in the **Metric Name** column, click a metric. For example **CPUUtilization**.
- At the top of the metric list, click **Select All**.

The graph shows all data for all occurrences of the selected metric. In the example below, CPUUtilization for all Amazon EC2 instances is shown.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create a CloudWatch Alarm



- To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Create a CloudWatch Alarm

You can create an Amazon CloudWatch alarm that monitors any one of your Amazon EC2 instance's CloudWatch metrics. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm on the Amazon EC2 console of the AWS Management Console, or you can use the CloudWatch console and configure more advanced options.

Contents

- [Send Email Based on CPU Usage Alarm \(p. 444\)](#)
- [Send Email Based on Load Balancer Alarm \(p. 446\)](#)
- [Send Email Based on Storage Throughput Alarm \(p. 448\)](#)

Send Email Based on CPU Usage Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to create an Amazon CloudWatch alarm that sends an Amazon Simple Notification Service email message when the alarm changes state from OK to ALARM.

In this scenario, you configure the alarm to change to the ALARM state when the average CPU use of an EC2 instance exceeds 70 percent for two consecutive five-minute periods.

AWS Management Console

To create an alarm that sends email based on CPU usage

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- If necessary, from the navigation bar, change the region to match the region where you have your instances.
- In the navigation pane, click **Alarms**.
- Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, select a metric category, for example, **EC2 Metrics**.
- In the list of metrics, select a row that contains **CPUUtilization** for a specific instance ID.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create a CloudWatch Alarm

A graph showing average CPUUtilization for a single instance appears in the lower pane.

The screenshot shows the 'Create Alarm' wizard in AWS CloudWatch. It is currently on the '1. Select Metric' step. A search for 'CPUUtilization' has been performed, and a list of EC2 instances is displayed. The instance 'i-0df23c383de4ebe2e' is selected. Below the list, a graph shows the 'Average' statistic over a '5 Minutes' period. The graph shows a baseline near 0% utilization with a sharp spike to approximately 28% at 11:00. The 'Time Range' is set to 'Relative' from 12.1 hours ago to 0 hours ago. The 'Zoom' is set to '1h'. The 'Left Y-axis' is labeled 'CPUUtilization'. At the bottom, there are buttons for 'Cancel', 'Previous', 'Next', and 'Create Alarm'.

6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **5 minutes**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **myHighCpuAlarm**.
9. In the **Description** field, enter a description of the alarm, for example: **CPU usage exceeds 70 percent**.
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter **70** and in the **for** field, enter **10**.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **myHighCpuAlarm**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the **ALARM** state.

15. Click **Create Alarm** to complete the alarm creation process.

AWS Command Line Interface

To send an Amazon Simple Notification Service email message when CPU utilization exceeds 70 percent

1. Set up an Amazon Simple Notification Service topic or retrieve the Topic Resource Name of the topic you intend to use. For help on setting up an Amazon Simple Notification Service topic, see [Set Up Amazon Simple Notification Service](#).
2. Create an alarm using the `put-metric-alarm` command as follows.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create a CloudWatch Alarm

```
$ aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-  
description "Alarm when CPU exceeds 70%" --metric-name CPUUtilization  
--namespace AWS/EC2 --statistic Average --period 300 --threshold  
70 --comparison-operator GreaterThanThreshold --dimensions  
Name=InstanceId,Value=i-1234567890abcdef0 --evaluation-periods 2 --alarm-  
actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

3. Test the alarm by forcing an alarm state change with the `set-alarm-state` command.
 - a. Change the alarm state from `INSUFFICIENT_DATA` to `OK`:

```
$ aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason  
"initializing" --state-value OK
```

- b. Change the alarm state from `OK` to `ALARM`:

```
$ aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason  
"initializing" --state-value ALARM
```

- c. Check that an email has been received.

Send Email Based on Load Balancer Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and configure an alarm that monitors load balancer latency exceeding 100 ms.

AWS Management Console

To create a load balancer alarm that sends email

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **ELB Metrics**.
5. In the list of metrics, select a row that contains **Latency** for a specific load balancer.

A graph showing average `Latency` for a single load balancer appears in the lower pane.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create a CloudWatch Alarm

The screenshot displays the 'Create Alarm' wizard in the AWS console. It is currently on the '1. Select Metric' step. A list of metrics is shown, with 'Latency' selected for the metric 'awseb-e-2-AWSEBLoa-1K2GC5PYQHIV'. Below the list, a graph visualizes the 'Average' of 'Latency (Seconds)' over a '5 Minutes' period. The graph shows data points for various times on 10/27, 10/28, 10/29, and 10/30. The y-axis ranges from 0.006 to 0.031. To the right of the graph, the 'Time Range' is set to 'Relative', starting '3 days ago' and ending '0 minutes ago'. At the bottom of the wizard, there are 'Cancel', 'Back', 'Next', and 'Create Alarm' buttons.

6. Select **Average** from the **Statistic** drop-down list.
7. Select **1 Minute** from the **Period** drop-down list.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `myHighCpuAlarm`.
9. In the **Description** field, enter a description of the alarm, for example: **Alarm when Latency exceeds 100ms**.
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter `0.1` and in the **for** field, enter `3`.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `myHighCpuAlarm`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

15. Click **Create Alarm** to complete the alarm creation process.

AWS Command Line Interface

To send an Amazon Simple Notification Service email message when LoadBalancer Latency Exceeds 100 milliseconds

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in [Set Up Amazon Simple Notification Service](#).
2. Use the `put-metric-alarm` command as follows.

```
$ aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm when Latency exceeds 100ms" --metric-name Latency --namespace AWS/ELB --statistic Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-topic --unit Milliseconds
```

3. Test the alarm.
 - Force an alarm state change to ALARM:

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --state-value OK
```

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --state-value ALARM
```

- Check that you have received an email notification about the alarm.

Send Email Based on Storage Throughput Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and to configure an alarm that sends email when EBS exceeds 100 MB throughput.

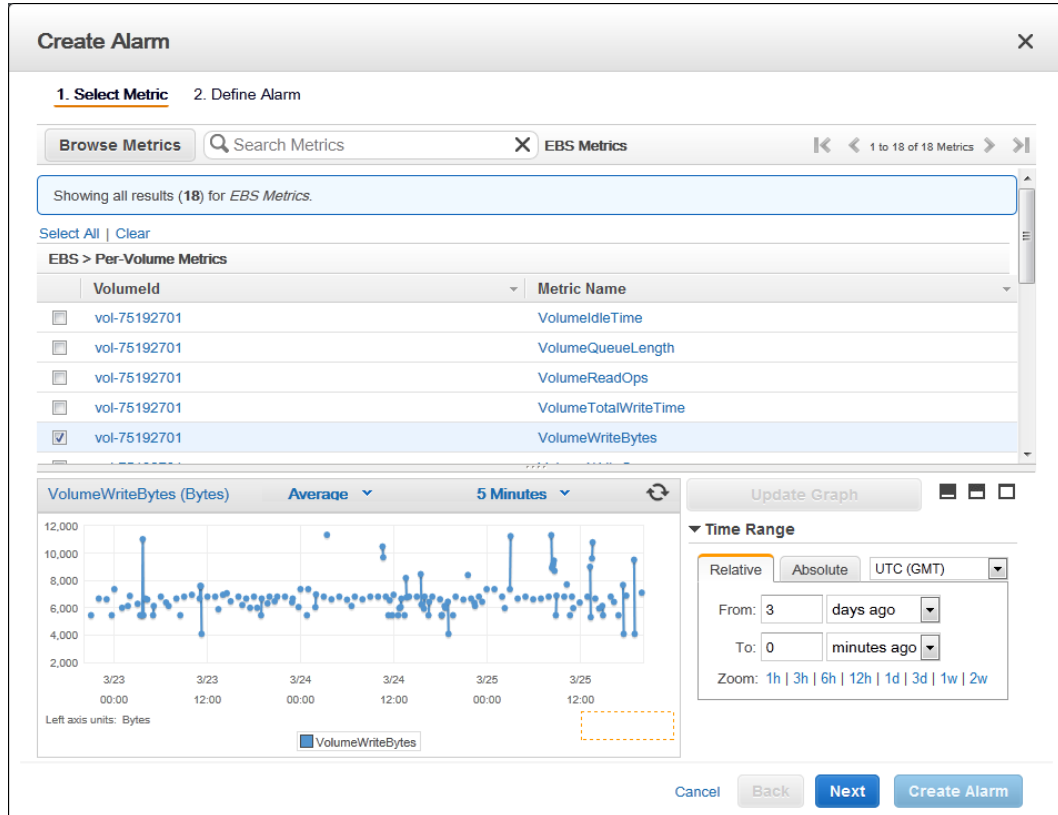
AWS Management Console

To create a storage throughput alarm that sends email

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, from the navigation bar, change the region to match the region where you have your instances.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **EBS Metrics**.
5. In the list of metrics, select a row that contains **VolumeWriteBytes** for a specific VolumeId.

A graph showing average `VolumeWriteBytes` for a single volume appears in the lower pane.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create a CloudWatch Alarm



6. Select **Average** from the **Statistic** drop-down list.
7. Select **5 Minutes** from the **Period** drop-down list.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `myHighWriteAlarm`.
9. In the **Description** field, enter a description of the alarm, for example: `VolumeWriteBytes exceeds 100,000 KiB/s`.
10. In the **is** drop-down list, select `>`.
11. In the box next to the **is** drop-down list, enter `100000` and in the **for** field, enter `15`.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `myHighCpuAlarm`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

15. Click **Create Alarm** to complete the alarm creation process.

AWS Command Line Interface

To send an Amazon Simple Notification Service email message when EBS exceeds 100 MB throughput

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in [Set Up Amazon Simple Notification Service](#).
2. Use the `put-metric-alarm` command as follows.

```
$ aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-  
description "Alarm when EBS volume exceeds 100MB throughput" --metric-  
name VolumeReadBytes --namespace AWS/EBS --statistic Average --period  
300 --threshold 100000000 --comparison-operator GreaterThanThreshold  
--dimensions Name=VolumeId,Value=my-volume-id --evaluation-periods  
3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-alarm-topic  
--insufficient-data-actions arn:aws:sns:us-east-1:1234567890:my-  
insufficient-data-topic
```

3. Test the alarm.
 - Force an alarm state change to ALARM.

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason  
"initializing" --state-value OK
```

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason  
"initializing" --state-value ALARM
```

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason  
"initializing" --state-value INSUFFICIENT_DATA
```

- Check that you have received two email notifications about the alarm.

Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your Amazon Elastic Compute Cloud (Amazon EC2) instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

Every alarm action you create uses alarm action ARNs. One set of ARNs is more secure because it requires you to have the EC2ActionsAccess IAM role in your account. This IAM role enables you to perform stop, terminate, or reboot actions--previously you could not execute an action if you were using an IAM role. Existing alarms that use the previous alarm action ARNs do not require this IAM role, however it is recommended that you change the ARN and add the role when you edit an existing alarm that uses these ARNs.

The EC2ActionsAccess IAM role enables AWS to perform alarm actions on your behalf. When you create an alarm action for the first time using the Amazon EC2 or Amazon CloudWatch consoles, AWS automatically creates this role for you. In addition, you must create the EC2ActionsAccess role using either console before it's available for use from the CLI.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can create an alarm that automatically recovers an Amazon EC2 instance when the instance becomes impaired due to an underlying hardware failure a problem that requires AWS involvement to repair. Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important

The recover action is only supported on:

- C3, C4, M3, M4, R3, T2, and X1 instance types.
- Instances in a VPC. If your instance has a public IP address, it will retain the same public IP address after recovery.
- Instances with shared tenancy (where the tenancy attribute of the instance is set to `default`).
- Instances that use Amazon EBS storage exclusively.

Currently, the recover action is not supported for EC2-Classic instances, dedicated tenancy instances, instances running on Dedicated hosts, and instances that use any instance store volumes.

You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the AWS/EC2 namespace), as well as any custom metrics that include the "InstanceId=" dimension, as long as the InstanceId value refers to a valid running Amazon EC2 instance.

Contents

- [Adding Stop Actions to Amazon CloudWatch Alarms \(p. 451\)](#)
- [Adding Terminate Actions to Amazon CloudWatch Alarms \(p. 454\)](#)
- [Adding Reboot Actions to Amazon CloudWatch Alarms \(p. 456\)](#)
- [Adding Recover Actions to Amazon CloudWatch Alarms \(p. 459\)](#)
- [Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions \(p. 461\)](#)
- [Using the CLI or the API to Create an Alarm to Stop, Terminate, Reboot, or Recover an Instance \(p. 462\)](#)
- [Amazon CloudWatch Alarm Action Scenarios \(p. 467\)](#)

Adding Stop Actions to Amazon CloudWatch Alarms

You can configure the stop alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS

Amazon Elastic Compute Cloud User Guide for Linux Instances Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see [Sample Code & Libraries](#).

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Amazon EC2 instances that use an Amazon Elastic Block Store volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Using the Amazon EC2 Console to Create an Alarm to Stop an Instance

To create an alarm to stop an idle instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the contents pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

Or, you can also select the instance, and then in the lower pane on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

7. Select the **Take the action** check box, and then choose the **Stop this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
10. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
11. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
12. To change the name of the alarm, in the **Name this alarm** box, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm that Stops an Instance

To create an alarm to stop an idle instance

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **1 Day**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **stop EC2 instance**.
9. In the **Description** field, enter a description of the alarm, for example: **stop EC2 instance when CPU is idle for too long**.
10. In the **is** drop-down list, select **<**.
11. In the box next to the **is** drop-down list, enter **10** and in the **for** field, enter **1440**.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Stop this instance**.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

15. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
16. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
17. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `stop_EC2_Instance`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

Important

If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

18. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Terminate Actions to Amazon CloudWatch Alarms

You can configure the terminate alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see [Sample Code & Libraries](#).

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on enabling and disabling termination protection for an instance, see [Enabling Termination Protection for an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Using the Amazon EC2 Console to Create an Alarm that Terminates an Instance

To create an alarm to terminate an idle instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

7. Select the **Take the action** check box, and then choose the **Terminate this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
10. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
11. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
12. To change the name of the alarm, in the **Name this alarm** box, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm to Terminate an Idle Instance

To create an alarm to terminate an idle instance

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, click **Alarms**.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **1 Day**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **Terminate EC2 instance**.
9. In the **Description** field, enter a description of the alarm, for example: **Terminate EC2 instance when CPU is idle for too long**.
10. In the **is** drop-down list, select **<**.
11. In the box next to the **is** drop-down list, enter **10** and in the **for** field, enter **1440**.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Terminate this instance**.
15. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
16. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
17. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **Terminate_EC2_Instance**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the **ALARM** state.

Important

If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

18. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Reboot Actions to Amazon CloudWatch Alarms

You can configure the reboot alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see [Sample Code & Libraries](#).

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance. For more information about rebooting an instance, see [Reboot Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Important

To avoid a race condition between the reboot and recover actions, we recommend that you set the alarm threshold to **3** for **1** minute when creating alarms that reboot an Amazon EC2 instance.

Using the Amazon EC2 Console to Create an Alarm to Reboot an Instance

To create an alarm to reboot an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

7. Select the **Take the action** check box, and then choose the **Reboot this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.

9. In the **Whenever** box, choose Status Check Failed (Instance).
10. In the **For at least** field, enter **2**.
11. In the **consecutive period(s) of** box, select **1 minute**.
12. To change the name of the alarm, in the **Name of alarm** box, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm to Reboot an Instance

To create an alarm to reboot an instance

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and `StatusCheckFailed_Instance` metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Minimum** from the **Statistic** drop-down list.

Note

This is the only statistic that is currently supported.

7. Select a period from the **Period** drop-down list, for example: **1 Minute**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **Reboot EC2 instance**.
9. In the **Description** field, enter a description of the alarm, for example: **Reboot EC2 instance when health checks fail**.
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter **0** and in the **for** field, enter **2**.

A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Reboot this instance**.
15. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
16. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **Reboot_EC2_Instance**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

Important

If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

17. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Recover Actions to Amazon CloudWatch Alarms

You can configure the recover alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see [Sample Code & Libraries](#).

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important

The recover action is only supported on:

- C3, C4, M3, M4, R3, T2, and X1 instance types.
- Instances in a VPC. Dedicated instances or instances running on Dedicated hosts are not supported. If your instance has a public IP address, it will retain the same public IP address after recovery.
- Instances that use EBS-backed storage. Instance storage is not supported. Automatic recovery of the instance will fail if any instance storage is attached.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2

Amazon Elastic Compute Cloud User Guide for Linux Instances Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Important

To avoid a race condition between the reboot and recover actions, we recommend that you set the alarm threshold to **2** for **1** minute when creating alarms that recover an Amazon EC2 instance.

Create an Alarm from the Amazon EC2 Console

To create an alarm to recover an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

7. Select the **Take the action** check box, and then choose the **Recover this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** box, choose Status Check Failed (System).
10. In the **For at least** field, enter **2**.
11. In the **consecutive period(s) of** box, select **1 minute**.
12. To change the name of the alarm, in the **Name of alarm** box, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

13. Click **Create Alarm**.

Create an Alarm from the Amazon CloudWatch Console

To create an alarm to recover an instance

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and `StatusCheckFailed_System` metric you want to create an alarm for. You can also type an instance ID in the search box to go to the instance that you want.
6. Select **Minimum** from the **Statistic** drop-down list.

Note

This is the only statistic that is currently supported.

7. Select a period from the **Period** drop-down list, for example: **1 Minute**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **Recover EC2 instance**.
9. In the **Description** field, enter a description of the alarm, for example: **Recover EC2 instance when health checks fail**.
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter 0 and in the **for** field, enter 2.

A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Recover this instance**.
15. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
16. To create a new Amazon SNS topic, select **New list**.

In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **Recover_EC2_Instance**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

Important

If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

17. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, click **Alarms**.
4. In the upper pane, select the alarm with the history that you want to view.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

5. In the lower pane, the **Details** tab shows the most recent state transition along with the time and metric values.
6. Click the **History** tab to view the most recent history entries.

Using the CLI or the API to Create an Alarm to Stop, Terminate, Reboot, or Recover an Instance

If you are using either the AWS CLI or the Amazon CloudWatch API, or if you are using the AWS SDKs with the API, you can create a CloudWatch alarm using an Amazon EC2 per-instance metric, and then add an action using the action's dedicated Amazon Resource Name (ARN). You can add the action to any alarm state, and you can specify the region for each action. The region must match the region to which you send the put-metric-alarm request.

Action	ARN (with region)	ARN (for use with IAM role)
<i>Stop</i>	arn:aws:automate:us-east-1:ec2:stop	arn:aws:swf:us-east-1:{ <i>customer-account</i> }:action/actions/AWS_EC2.InstanceId.Stop/1.0 Note You must create at least one stop alarm using the Amazon EC2 or CloudWatch console to create the EC2ActionsAccess IAM role. After this IAM role is created, you can create stop alarms using the CLI.
<i>Terminate</i>	arn:aws:automate:us-east-1:ec2:terminate	arn:aws:swf:us-east-1:{ <i>customer-account</i> }:action/actions/AWS_EC2.InstanceId.Terminate/1.0 Note You must create at least one terminate alarm using the Amazon EC2 or CloudWatch console to create the EC2ActionsAccess IAM role. After this IAM role is created, you can create terminate alarms using the CLI.
<i>Reboot</i>	n/a	arn:aws:swf:us-east-1:{ <i>customer-account</i> }:action/actions/AWS_EC2.InstanceId.Reboot/1.0 Note You must create at least one reboot alarm

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

Action	ARN (with region)	ARN (for use with IAM role)
		using the Amazon EC2 or CloudWatch console to create the EC2ActionsAccess IAM role. After this IAM role is created, you can create reboot alarms using the CLI.
<i>Recover</i>	arn:aws:automate:us-east-1:ec2:recover	n/a

For information about using the Amazon CloudWatch API with the AWS SDKs, see [Sample Code & Libraries](#).

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to stop an instance using the CLI

You can use the `arn:aws:automate:us-east-1:ec2:stop` ARN to stop an Amazon EC2 instance. The following example shows how to stop an instance if the average CPU utilization is less than 10 percent over a 24 hour period.

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-
description "Stop the instance when it is idle for a day" --namespace
"AWS/EC2" --dimensions Name=InstanceId,Value="i-1234567890abcdef0" --
statistic Average --metric-name CPUUtilization --comparison-operator
LessThanThreshold --threshold 10 --period 21600 --evaluation-periods 4 --
alarm-actions arn:aws:automate:us-east-1:ec2:stop
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

To create an alarm to terminate an instance using the CLI

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-  
description "Terminate the instance when it is idle for a day" --namespace  
"AWS/EC2" --dimensions Name=InstanceId,Value="i-1234567890abcdef0" --  
statistic Average --metric-name CPUUtilization --comparison-operator  
LessThanThreshold --threshold 1 --period 21600 --evaluation-periods 4 --  
alarm-actions arn:aws:automate:us-east-1:ec2:terminate
```

To create an alarm to reboot an instance using the CLI

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-  
description "Reboot the instance" --namespace "AWS/EC2" --dimensions  
Name=InstanceId,Value="i-1234567890abcdef0" --statistic Minimum  
--metric-name StatusCheckFailed_Instance --comparison-operator  
GreaterThanThreshold --threshold 0 --period 60 --evaluation-periods 2  
--alarm-actions arn:aws:swf:us-east-1:{customer-account}:action/actions/  
AWS_EC2.InstanceId.Reboot/1.0
```

To create an alarm to recover an instance using the CLI

Note

Terminated instances cannot be recovered.

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-  
description "Recover the instance" --namespace "AWS/EC2" --dimensions  
Name=InstanceId,Value="i-1234567890abcdef0" --statistic Average  
--metric-name StatusCheckFailed_System --comparison-operator  
GreaterThanThreshold --threshold 0 --period 60 --evaluation-periods 2 --  
alarm-actions arn:aws:automate:us-east-1:ec2:recover
```

To create an alarm to stop an instance using the API

The following example request shows how to create an alarm that stops an Amazon EC2 instance:

- ```
http://monitoring.amazonaws.com/

?SignatureVersion=2

&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id
```

**Amazon Elastic Compute Cloud  
User Guide for Linux Instances  
Create Alarms That Stop, Terminate,  
Reboot, or Recover an Instance**

```
&Dimension.member.1.Value=i-1234567890abcdef0
&Period=21600
&Statistic=Average
&AlarmName=Stop-EC2-Instance
&ComparisonOperator=LessThanThreshold
&Threshold=10
&EvaluationPeriods=4
&StartTime=2009-01-16T00:00:00
&EndTime=2009-01-16T00:02:00
&Timestamp=2009-01-08-18
&AWSSecretAccessKeyId=XXX YOUR ACCESS KEY XXX
&Signature=%XXX YOUR SIGNATURE XXX%3D
&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:stop
```

**To create an alarm to terminate an instance using the API**

The following example request shows how to create an alarm that terminates an Amazon EC2 instance:

- ```
http://monitoring.amazonaws.com/
?SignatureVersion=2
&Action=PutMetricAlarm
&Version=2009-05-15
&Namespace=AWS/EC2
&MetricName=CPUUtilization
&Dimension.member.1.Name=instance-id
&Dimension.member.1.Value=i-1234567890abcdef0
&Period=21600
&Statistic=Average
&AlarmName=Terminate-EC2-Instance
&ComparisonOperator=LessThanThreshold
&Threshold=10
```


**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

```
&EvaluationPeriods=4
&StartTime=2009-01-16T00:00:00
&EndTime=2009-01-16T00:02:00
&Timestamp=2009-01-08-18
&AWSSecretAccessKeyId=XXX YOUR ACCESS KEY XXX
&Signature=%XXX YOUR SIGNATURE XXX%3D
&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:terminate
```

To create an alarm to reboot an instance using the API

The following example request shows how to create an alarm that reboots an Amazon EC2 instance:

- ```
http://monitoring.amazonaws.com/
?SignatureVersion=2
&Action=PutMetricAlarm
&Version=2009-05-15
&Namespace=AWS/EC2
&MetricName=StatusCheckFailed_Instance
&Dimension.member.1.Name=instance-id
&Dimension.member.1.Value=i-1234567890abcdef0
&Period=60
&Statistic=Average
&AlarmName=Reboot-EC2-Instance
&ComparisonOperator=GreaterThanThreshold
&Threshold=0
&EvaluationPeriods=2
&StartTime=2009-01-16T00:00:00
&EndTime=2009-01-16T00:02:00
&Timestamp=2009-01-08-18
&AWSSecretAccessKeyId=XXX YOUR ACCESS KEY XXX
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

**Amazon Elastic Compute Cloud  
User Guide for Linux Instances  
Create Alarms That Stop, Terminate,  
Reboot, or Recover an Instance**

```
&AlarmActions.member.1=arn:aws:aws:swf:us-east-1:{customer-
account}:action/actions/AWS_EC2.InstanceId.Reboot/1.0
```

### To create an alarm to recover an instance using the API

The following example request shows how to create an alarm that recovers an Amazon EC2 instance:

- ```
http://monitoring.amazonaws.com/  
  
?SignatureVersion=2  
  
&Action=PutMetricAlarm  
  
&Version=2009-05-15  
  
&Namespace=AWS/EC2  
  
&MetricName=StatusCheckFailed_System  
  
&Dimension.member.1.Name=instance-id  
&Dimension.member.1.Value=i-1234567890abcdef0  
  
&Period=60  
  
&Statistic=Average  
  
&AlarmName=Terminate-EC2-Instance  
  
&ComparisonOperator=GreaterThanThreshold  
  
&Threshold=0  
  
&EvaluationPeriods=2  
  
&StartTime=2009-01-16T00:00:00  
  
&EndTime=2009-01-16T00:02:00  
  
&Timestamp=2009-01-08-18  
  
&AWSSecretAccessKeyId=XXX YOUR ACCESS KEY XXX  
  
&Signature=%XXX YOUR SIGNATURE XXX%3D  
  
&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:recover
```

Amazon CloudWatch Alarm Action Scenarios

You can use the Amazon EC2 console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.

Amazon Elastic Compute Cloud User Guide for Linux Instances Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

Create Alarm
✕

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

1 **Send a notification to:** [create topic](#)

Take the action:

- Recover this instance ⓘ
- Stop this instance ⓘ
- Terminate this instance ⓘ
- Reboot this instance ⓘ

AWS will create the following IAM role in your account so that AWS can perform this action. [Learn more.](#)

Create IAM role: **EC2ActionsAccess** (show IAM policy document)

Whenever: of

Is: Percent

For at least: consecutive period(s) of

Name of alarm:

CPU Utilization Percent

75
50
25
0

7/21 22:00 7/22 00:00 7/22 02:00

i-0a723c3830e4e62e

Cancel Create Alarm

Scenario 1: Stop Idle Development and Test Instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
	Stop
	Maximum
	CPUUtilization
	<=
	10%
	60 minutes
	1

Scenario 2: Stop Idle Instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance**

Setting	Value
	Stop and email
	Average
	CPUUtilization
	<=
	5%
	60 minutes
	24

Scenario 3: Send Email About Web Servers with Unusually High Traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Setting	Value
	Email
	Sum
	NetworkOut
	>
	10 GB
	1 day
	1

Scenario 4: Stop Web Servers with Unusually High Traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
	Stop and send SMS
	Sum
	NetworkOut
	>
	1 GB
	1 hour
	1

Scenario 5: Stop an Instance Experiencing a Memory Leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90%, so that application logs can be retrieved for troubleshooting.

Note

The MemoryUtilization metric is a custom metric. In order to use the MemoryUtilization metric, you must install the Perl scripts for Linux instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).

Setting	Value
	Stop
	Maximum
	MemoryUtilization
	>=
	90%
	1 minute
	1

Scenario 6: Stop an Impaired Instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

Setting	Value
	Stop
	Average
	StatusCheckFailed_System
	>=
	1
	15 minutes
	1

Scenario 7: Terminate Instances When Batch Processing Jobs Are Complete

Create an alarm that terminates an instance that runs batch jobs when it is no longer sending results data.

Setting	Value
	Terminate
	Maximum

Setting	Value
	NetworkOut
	<=
	100,000 bytes
	5 minutes
	1

Scenarios Using the CloudWatch Console

The previous scenarios can also be performed using the Amazon CloudWatch console. We've numbered the settings on the console to match the numbered settings in the Amazon EC2 console and the scenarios that we covered earlier, so you can make a comparison and create an alarm with the appropriate actions.

Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

The Amazon CloudWatch Monitoring Scripts for Amazon Elastic Compute Cloud (Amazon EC2) Linux-based instances demonstrate how to produce and consume Amazon CloudWatch custom metrics.

These sample Perl scripts comprise a fully functional example that reports memory, swap, and disk space utilization metrics for a Linux instance. You can download the [Amazon CloudWatch Monitoring Scripts for Linux](#) from the AWS sample code library.

Important

These scripts are examples only. They are provided "as is" and are not supported.

These monitoring scripts are intended for use with Amazon EC2 instances running Linux operating systems. The scripts have been tested on the following Amazon Machine Images (AMIs) for both 32-bit and 64-bit versions:

- Amazon Linux 2014.09.2
- Red Hat Enterprise Linux 6.6
- SUSE Linux Enterprise Server 12
- Ubuntu Server 14.04

Note

Standard Amazon CloudWatch free tier quantities and usage charges for custom metrics apply to your use of these scripts. For more information, see the [Amazon CloudWatch pricing page](#).

You can use EC2Config on Amazon EC2 instances running Windows to monitor memory and disk metrics by sending this data to CloudWatch Logs. To get started with CloudWatch Logs on an Amazon EC2 instance running Windows, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [Prerequisites \(p. 472\)](#)
- [Getting Started \(p. 474\)](#)
- [mon-put-instance-data.pl \(p. 475\)](#)
- [mon-get-instance-stats.pl \(p. 477\)](#)
- [Viewing Your Custom Metrics in the Console \(p. 479\)](#)

Prerequisites

You must perform additional steps on some versions of Linux.

Note

The CloudWatchClient.pm module included in the script package locally caches instance metadata. If you create an AMI from an instance where you have run the scripts, any instances launched from this AMI within the cache TTL (default: six hours, 24 hours for Auto Scaling groups) will emit metrics using the original instance's ID. After the cache TTL time period passes, the script will retrieve fresh data and the scripts will use the current instance's ID. To immediately correct this, remove the cached data using: `$ rm /var/tmp/aws-mon/instance-id.`

Amazon Linux AMI

If you are running Amazon Linux AMI, version 2014.03 or later, you'll need to add some additional Perl modules in order for the monitoring scripts to work. Use the following procedures to configure your server.

To install the scripts for the first time

1. Log on to your Amazon Linux AMI instance.

Amazon Elastic Compute Cloud User Guide for Linux Instances Prerequisites

2. At a command prompt, install the following package:

```
$ sudo yum install perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https
```

Red Hat Enterprise Linux

If you are running Red Hat Enterprise Linux, you'll need to add some additional Perl modules in order for the monitoring scripts to work. Use the following procedures to configure your server.

To install the scripts for the first time

1. Log on to your Red Hat Enterprise Linux instance.
2. At a command prompt, install the following packages:

```
$ sudo yum install perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA -y
```

```
$ sudo yum install zip unzip
```

SUSE Linux Enterprise Server

If you are running SUSE Linux Enterprise Server, you'll need to add some additional Perl modules in order for the monitoring scripts to work. Use the following procedures to configure your server.

To install the scripts for the first time

1. Log on to your SUSE Linux Enterprise Server instance.
2. At a command prompt, install the following packages:

```
$ sudo zypper install perl-Switch perl-DateTime
```

```
$ sudo zypper install -y "perl(LWP::Protocol::https)"
```

Ubuntu Server

If you are running Ubuntu Server, use the following procedures to configure your server.

To install the scripts for the first time

1. Log on to your Ubuntu Server instance.
2. At a command prompt, install the following packages:

```
$ sudo apt-get update
```

```
$ sudo apt-get install unzip
```

```
$ sudo apt-get install libwww-perl libdatetime-perl
```


For information about connecting to Amazon EC2 Linux instances, see [Connect to Your Linux Instance](#) (p. 289).

Getting Started

The following steps show you how to download, uncompress, and configure the CloudWatch Monitoring Scripts on an EC2 Linux instance.

To download, install, and configure the script

1. Open a command prompt, move to a folder where you want to store the scripts and then type the following:

```
curl http://aws-cloudwatch.s3.amazonaws.com/downloads/
CloudWatchMonitoringScripts-1.2.1.zip -O
unzip CloudWatchMonitoringScripts-1.2.1.zip
rm CloudWatchMonitoringScripts-1.2.1.zip
cd aws-scripts-mon
```

The `CloudWatchMonitoringScripts-1.2.1.zip` package contains these files:

- **CloudWatchClient.pm**—Shared Perl module that simplifies calling Amazon CloudWatch from other scripts.
 - **mon-put-instance-data.pl**—Collects system metrics on an Amazon EC2 instance (memory, swap, disk space utilization) and sends them to Amazon CloudWatch.
 - **mon-get-instance-stats.pl**—Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
 - **awscreds.template**—File template for AWS credentials that stores your access key ID and secret access key.
 - **LICENSE.txt**—Text file containing the Apache 2.0 license.
 - **NOTICE.txt**—copyright notice.
2. If you already have an AWS Identity and Access Management (IAM) role associated with your instance, make sure that it has permissions to perform the following operations:
 - `cloudwatch:PutMetricData`
 - `cloudwatch:GetMetricStatistics`
 - `cloudwatch:ListMetrics`
 - `ec2:DescribeTags`

Otherwise, you can create a new IAM role with permissions to perform CloudWatch operations and associate that role when you launch a new instance. For more information, see [Controlling User Access to Your AWS Account](#).

3. Optional: If you aren't using an IAM role, update the `awscreds.template` file that you downloaded earlier. The content of this file should use the following format:

```
AWSAccessKeyId=YourAccessKeyID
```

```
AWSecretKey=YourSecretAccessKey
```

Note

This step is optional if you have already created a file for credentials. You can use an existing file by specifying its location on the command line when you call the scripts. Alternatively, you can set the environment variable `AWS_CREDENTIAL_FILE` to point to the file with your AWS credentials.

For instructions on how to access your credentials, see [Creating, Modifying, and Viewing User Security Credentials](#) in the *IAM User Guide*.

mon-put-instance-data.pl

This script collects memory, swap, and disk space utilization data on the current system. It then makes a remote call to Amazon CloudWatch to report the collected data as custom metrics.

Options

Name	Description
<code>--mem-util</code>	Collects and sends the MemoryUtilization metrics in percentages. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
<code>--mem-used</code>	Collects and sends the MemoryUsed metrics, reported in megabytes. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
<code>--mem-avail</code>	Collects and sends the MemoryAvailable metrics, reported in megabytes. This option reports memory available for use by applications and the operating system.
<code>--swap-util</code>	Collects and sends SwapUtilization metrics, reported in percentages.
<code>--swap-used</code>	Collects and sends SwapUsed metrics, reported in megabytes.
<code>--disk-path=PATH</code>	Selects the disk on which to report. PATH can specify a mount point or any file located on a mount point for the filesystem that needs to be reported. For selecting multiple disks, specify a <code>--disk-path=PATH</code> for each one of them. To select a disk for the filesystems mounted on <code>/</code> and <code>/home</code> , use the following parameters: <code>--disk-path=/</code> <code>--disk-path=/home</code>
<code>--disk-space-util</code>	Collects and sends the DiskSpaceUtilization metric for the selected disks. The metric is reported in percentages.
<code>--disk-space-used</code>	Collects and sends the DiskSpaceUsed metric for the selected disks. The metric is reported by default in gigabytes. Due to reserved disk space in Linux operating systems, disk space used and disk space available might not accurately add up to the amount of total disk space.
<code>--disk-space-avail</code>	Collects and sends the DiskSpaceAvailable metric for the selected disks. The metric is reported in gigabytes. Due to reserved disk space in the Linux operating systems, disk space used and disk space available might not accurately add up to the amount of total disk space.
<code>--memory-units=UNITS</code>	Specifies units in which to report memory usage. If not specified, memory is reported in megabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
mon-put-instance-data.pl**

Name	Description
<code>--disk-space-units=UNITS</code>	Specifies units in which to report disk space usage. If not specified, disk space is reported in gigabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
<code>--aws-credential-file=PATH</code>	Provides the location of the file containing AWS credentials. This parameter cannot be used with the <code>--aws-access-key-id</code> and <code>--aws-secret-key</code> parameters.
<code>--aws-access-key-id=VALUE</code>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>--aws-secret-key</code> option. Do not use this option with the <code>--aws-credential-file</code> parameter.
<code>--aws-secret-key=VALUE</code>	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the <code>--aws-access-key-id</code> option. Do not use this option with <code>--aws-credential-file</code> parameter.
<code>--aws-iam-role=VALUE</code>	Specifies the IAM role used to provide AWS credentials. The value <code>=VALUE</code> is required. If no credentials are specified, the default IAM role associated with the EC2 instance is applied. Only one IAM role can be used. If no IAM roles are found, or if more than one IAM role is found, the script will return an error. Do not use this option with the <code>--aws-credential-file</code> , <code>--aws-access-key-id</code> , or <code>--aws-secret-key</code> parameters.
<code>--aggregated[=only]</code>	Adds aggregated metrics for instance type, AMI ID, and overall for the region. The value <code>=only</code> is optional; if specified, the script reports only aggregated metrics.
<code>--auto-scaling[=only]</code>	Adds aggregated metrics for the Auto Scaling group. The value <code>=only</code> is optional; if specified, the script reports only Auto Scaling metrics. The IAM policy associated with the IAM account or role using the scripts need to have permissions to call the EC2 action DescribeTags .
<code>--verify</code>	Performs a test run of the script that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch to report the data. This option also checks that credentials are provided. When run in verbose mode, this option outputs the metrics that will be sent to CloudWatch.
<code>--from-cron</code>	Use this option when calling the script from cron. When this option is used, all diagnostic output is suppressed, but error messages are sent to the local system log of the user account.
<code>--verbose</code>	Displays detailed information about what the script is doing.
<code>--help</code>	Displays usage information.
<code>--version</code>	Displays the version number of the script.

Examples

The following examples assume that you have already updated the `awscreds.template` file with valid AWS credentials. If you are not using the `awscreds.template` file, provide credentials using the `--aws-access-key-id` and `--aws-secret-key` arguments.

To perform a simple test run without posting data to CloudWatch

- Run the following command:

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

To collect all available memory metrics and send them to CloudWatch

- Run the following command:

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail
```

To set a cron schedule for metrics reported to CloudWatch

1. Start editing the crontab using the following command:

```
crontab -e
```

2. Add the following command to report memory and disk space utilization to CloudWatch every five minutes:

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-util --disk-  
space-util --disk-path=/ --from-cron
```

If the script encounters an error, the script will write the error message in the system log.

To collect aggregated metrics for an Auto Scaling group and send them to Amazon CloudWatch without reporting individual instance metrics

- Run the following command:

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-  
scaling=only
```

To collect aggregated metrics for instance type, AMI ID and region, and send them to Amazon CloudWatch without reporting individual instance metrics

- Run the following command:

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --  
aggregated=only
```

mon-get-instance-stats.pl

This script queries CloudWatch for statistics on memory, swap, and disk space metrics within the time interval provided using the number of most recent hours. This data is provided for the Amazon EC2 instance on which this script is executed.

Options

Name	Description
<code>--recent-hours=N</code>	Specifies the number of recent hours to report on, as represented by <code>N</code> where <code>N</code> is an integer.
<code>--aws-credential-file=PATH</code>	Provides the location of the file containing AWS credentials.
<code>--aws-access-key-id=VALUE</code>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>--aws-secret-key</code> option. Do not use this option with the <code>--aws-credential-file</code> option.
<code>--aws-secret-key=VALUE</code>	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the <code>--aws-access-key-id</code> option. Do not use this option with <code>--aws-credential-file</code> option.
<code>--aws-iam-role=VALUE</code>	Specifies the IAM role used to provide AWS credentials. The value <code>=VALUE</code> is required. If no credentials are specified, the default IAM role associated with the EC2 instance is applied. Only one IAM role can be used. If no IAM roles are found, or if more than one IAM role is found, the script will return an error. Do not use this option with the <code>--aws-credential-file</code> , <code>--aws-access-key-id</code> , or <code>--aws-secret-key</code> parameters.
<code>--verify</code>	Performs a test run of the script that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch to report the data. This option also checks that credentials are provided. When run in verbose mode, this option outputs the metrics that will be sent to CloudWatch.
<code>--verbose</code>	Displays detailed information about what the script is doing.
<code>--help</code>	Displays usage information.
<code>--version</code>	Displays the version number of the script.

Examples

To get utilization statistics for the last 12 hours

- Run the following command:

```
./mon-get-instance-stats.pl --recent-hours=12
```

The returned response will be similar to the following example output:

```
Instance metric statistics for the last 12 hours.  
  
CPU Utilization  
Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%  
  
Memory Utilization  
Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%
```

```
Swap Utilization
  Average: N/A, Minimum: N/A, Maximum: N/A

Disk Space Utilization on /dev/xvda1 mounted as /
  Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

Viewing Your Custom Metrics in the Console

If you successfully call the `mon-put-instance-data.pl` script, you can use the AWS Management Console to view your posted custom metrics in the Amazon CloudWatch console.

To view custom metrics

1. Execute `mon-put-instance-data.pl`, as described earlier.
2. Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
3. Click **View Metrics**.
4. In the **Viewing** list, your custom metrics posted by the script are displayed with the prefix `System/Linux`.

Network and Security

Amazon EC2 provides the following network and security features.

Features

- [Amazon EC2 Key Pairs \(p. 481\)](#)
- [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#)
- [Controlling Access to Amazon EC2 Resources \(p. 498\)](#)
- [Amazon EC2 and Amazon Virtual Private Cloud \(p. 543\)](#)
- [Amazon EC2 Instance IP Addressing \(p. 570\)](#)
- [Elastic IP Addresses \(p. 581\)](#)
- [Elastic Network Interfaces \(ENI\) \(p. 588\)](#)
- [Placement Groups \(p. 602\)](#)
- [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 605\)](#)
- [Enhanced Networking on Linux \(p. 608\)](#)

If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key. For more information, see [How Do I Get Security Credentials?](#) in the *Amazon Web Services General Reference*.

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic or a default VPC is automatically assigned a public IP address. An instance that's launched into a nondefault VPC can be assigned a public IP address on launch. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 549\)](#).

Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, you can use an *Elastic IP address*.

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.

Amazon EC2 Key Pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. Linux instances have no password, and you use a key pair to log in using SSH. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see [Creating Your Key Pair Using Amazon EC2 \(p. 482\)](#).

Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see [Importing Your Own Key Pair to Amazon EC2 \(p. 482\)](#).

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name.

Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance.

Note

Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose a private key, there is no way to recover it. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair. If you lose the private key for an EBS-backed Linux instance, you can regain access to your instance. For more information, see [Connecting to Your Linux Instance if You Lose Your Private Key \(p. 486\)](#).

Key Pairs for Multiple Users

If you have several users that require access to a single instance, you can add user accounts to your instance. For more information, see [Managing User Accounts on Your Linux Instance \(p. 320\)](#). You can create a key pair for each user, and add the public key information from each key pair to the `.ssh/authorized_keys` file for each user on your instance. You can then distribute the private key files to your users. That way, you do not have to distribute the same private key file that's used for the root account to multiple users.

Contents

- [Creating Your Key Pair Using Amazon EC2 \(p. 482\)](#)
- [Importing Your Own Key Pair to Amazon EC2 \(p. 482\)](#)
- [Retrieving the Public Key for Your Key Pair on Linux \(p. 484\)](#)
- [Retrieving the Public Key for Your Key Pair on Windows \(p. 485\)](#)

- [Verifying Your Key Pair's Fingerprint](#) (p. 485)
- [Deleting Your Key Pair](#) (p. 486)
- [Connecting to Your Linux Instance if You Lose Your Private Key](#) (p. 486)

Creating Your Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line. After you create a key pair, you can specify it when you launch your instance. You can also add the key pair to a running instance to enable another user to connect to the instance. For more information, see [Managing User Accounts on Your Linux Instance](#) (p. 320).

To create your key pair using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

Tip

The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane.

3. Choose **Create Key Pair**.
4. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**.
5. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

6. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
$ chmod 400 my-key-pair.pem
```

To create your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [create-key-pair](#) (AWS CLI)
- [New-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Importing Your Own Key Pair to Amazon EC2

If you used Amazon EC2 to create your key pair, as described in the previous section, you are ready to launch an instance. Otherwise, instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair. Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

Amazon EC2 accepts the following formats:

- OpenSSH public key format (the format in `~/.ssh/authorized_keys`)
- Base64 encoded DER format
- SSH public key file format as specified in [RFC4716](#)

Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 2048, and 4096.

To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, `~/.ssh/my-key-pair.pub` (Linux) or `C:\keys\my-key-pair.pub` (Windows). The file name extension for this file is not important.
3. Save the private key to a different local file that has the `.pem` extension. For example, `~/.ssh/my-key-pair.pem` (Linux) or `C:\keys\my-key-pair.pem` (Windows). Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Use the following steps to import your key pair using the Amazon EC2 console.

To import the public key

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Import Key Pair**.
4. In the **Import Key Pair** dialog box, choose **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and choose **Import**.

To import your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [import-key-pair](#) (AWS CLI)
- [Import-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

After the public key file is imported, you can verify that the key pair was imported successfully using the Amazon EC2 console as follows.

To verify that your key pair was imported

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you created the key pair.
3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Verify that the key pair that you imported is in the displayed list of key pairs.

To view your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-key-pairs](#) (AWS CLI)
- [Get-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Retrieving the Public Key for Your Key Pair on Linux

On a Linux instance, the public key content is placed in an entry within `~/.ssh/authorized_keys`. This is done at boot time and enables you to securely access your instance without passwords. You can open this file in an editor to view the public key for your key pair. The following is an example entry for the key pair named `my-key-pair`. It consists of the public key followed by the name of the key pair. For example:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXr
lsLnBItnctckiJ7FbtXJMXLvwwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk
+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

You can use **ssh-keygen** to get the public key for your key pair. Run the following command on a computer to which you've downloaded your private key:

```
$ ssh-keygen -y
```

When prompted to enter the file in which the key is, specify the path to your `.pem` file; for example:

```
/path_to_key_pair/my-key-pair.pem
```

The command returns the public key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXr
lsLnBItnctckiJ7FbtXJMXLvwwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk
+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

If this command fails, ensure that you've changed the permissions on your key pair file so that only you can view it by running the following command:

```
$ chmod 400 my-key-pair.pem
```

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

```
$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Retrieving the Public Key for Your Key Pair on Windows

```
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/  
d6RJhJOI0iBXR  
lsLnBItnctckiJ7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUMozce5Pjz5/i8SeJtjnV3iAoG/cQk  
+0FzZ  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi  
+z7wB3Rb  
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

For more information, see [Retrieving Instance Metadata \(p. 338\)](#).

Note that if you change the key pair that you use to connect to the instance, as shown in the last section on this page, we don't update the instance metadata to show the new public key; you'll continue to see the public key for the key pair you specified when you launched the instance in the instance metadata.

Retrieving the Public Key for Your Key Pair on Windows

On Windows, you can use PuTTYgen to get the public key for your key pair. Start PuTTYgen, click **Load**, and select the .ppk or .pem file. PuTTYgen displays the public key.

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

```
$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key  
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/  
d6RJhJOI0iBXR  
lsLnBItnctckiJ7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUMozce5Pjz5/i8SeJtjnV3iAoG/cQk  
+0FzZ  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi  
+z7wB3Rb  
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

For more information, see [Retrieving Instance Metadata \(p. 338\)](#).

Verifying Your Key Pair's Fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key that's stored in AWS.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file:

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -  
nocrypt | openssl sha1 -c
```

If you created your key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file on your local machine:

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

The output should match the fingerprint that's displayed in the console.

Deleting Your Key Pair

When you delete a key pair, you are only deleting Amazon EC2's copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (`.pem`) file.

Note

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the key pair you're deleting is not specified in your launch configuration. Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

You can delete a key pair using the Amazon EC2 console or the command line.

To delete your key pair using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Select the key pair and choose **Delete**.
4. When prompted, choose **Yes**.

To delete your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-key-pair](#) (AWS CLI)
- [Remove-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Note

If you create a Linux AMI from an instance, and then use the AMI to launch a new instance in a different region or account, the new instance includes the public key from the original instance. This enables you to connect to the new instance using the same private key file as your original instance. You can remove this public key from your instance by removing its entry from the `.ssh/authorized_keys` file using a text editor of your choice. For more information about managing users on your instance and providing remote access using a specific key pair, see [Managing User Accounts on Your Linux Instance \(p. 320\)](#).

Connecting to Your Linux Instance if You Lose Your Private Key

If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the `authorized_keys` file, move the volume back to the original instance, and restart the

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Connecting to Your Linux Instance
if You Lose Your Private Key**

instance. For more information about launching, connecting to, and stopping instances, see [Instance Lifecycle](#) (p. 275).

This procedure isn't supported for instance store-backed instances. To determine the root device type of your instance, open the Amazon EC2 console, choose **Instances**, select the instance, and check the value of **Root device type** in the details pane. The value is either `ebs` or `instance store`. If the root device is an instance store volume, you must have the private key in order to connect to the instance.

Prerequisites

Create a new key pair using either the Amazon EC2 console or a third-party tool.

To connect to an EBS-backed instance with a different key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** in the navigation pane, and then select the instance that you'd like to connect to. (We'll refer to this as the original instance.)
3. Save the following information that you'll need to complete this procedure.
 - Write down the instance ID, AMI ID, and Availability Zone of the original instance.
 - Choose the entry for `sda1` (the root volume) under **Block devices** in the details pane and write down the volume ID in the **EBS ID** field (vol-xxxxxxx).
 - [EC2-Classic] If the original instance has an associated Elastic IP address, write down the Elastic IP address shown in the **Elastic IP** field in the details pane.
4. Choose **Actions**, select **Instance State**, and then select **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

5. Choose **Launch Instance**, and then use the launch wizard to launch a temporary instance with the following options:
 - On the **Choose an AMI** page, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89).
 - On the **Choose an Instance Type** page, leave the default instance type that the wizard selects for you.
 - On the **Configure Instance Details** page, specify the same Availability Zone as the instance you'd like to connect to. If you're launching an instance in a VPC, select a subnet in this Availability Zone.
 - On the **Tag Instance** page, add the tag `Name=Temporary` to the instance to indicate that this is a temporary instance.
 - On the **Review** page, choose **Launch**. Create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.
6. In the navigation pane, choose **Volumes** and select the root device volume for the original instance (you wrote down its volume ID in a previous step). Choose **Actions**, and then select **Detach Volume**. Wait for the state of the volume to become `available`. (You might need to choose the **Refresh** icon.)
7. With the volume still selected, choose **Actions**, and then select **Attach Volume**. Select the instance ID of the temporary instance, write down the device name specified under **Device** (for example, `/dev/sdf`), and then choose **Yes, Attach**.
8. Connect to the temporary instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Connecting to Your Linux Instance
if You Lose Your Private Key**

9. From the temporary instance, mount the volume that you attached to the instance so that you can access its file system. For example, if the device name is `/dev/sdf`, use the following commands to mount the volume as `/mnt/tempvol`.

Note

The device name may appear differently on your instance. For example, devices mounted as `/dev/sdf` may show up as `/dev/xvdf` on the instance. Some versions of Red Hat (or its variants, such as CentOS) may even increment the trailing letter by 4 characters, where `/dev/sdf` becomes `/dev/xvdk`.

- a. Use the **lsblk** command to determine if the volume is partitioned.

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
##xvda1    202:1    0   8G  0 part /
xvdf        202:80   0  101G  0 disk
##xvdf1    202:81   0  101G  0 part
xvdg        202:96   0   30G  0 disk
```

In the above example, `/dev/xvda` and `/dev/xvdf` are partitioned volumes, and `/dev/xvdg` is not. If your volume is partitioned, you mount the partition (`/dev/xvdf1`) instead of the raw device (`/dev/xvdf`) in the next steps.

- b. Create a temporary directory to mount the volume.

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

- c. Mount the volume (or partition) at the temporary mount point, using the volume name or device name you identified earlier.

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

10. From the temporary instance, use the following command to update `authorized_keys` on the mounted volume with the new public key from the `authorized_keys` for the temporary instance (you may need to substitute a different user name in the following command, such as `ubuntu` for Ubuntu instances):

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

If this copy succeeded, you can go to the next step.

(Optional) Otherwise, if you don't have permission to edit files in `/mnt/tempvol`, you'll need to update the file using **sudo** and then check the permissions on the file to verify that you'll be able to log into the original instance. Use the following command to check the permissions on the file:

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
total 4
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

In this example output, `222` is the user ID and `500` is the group ID. Next, use **sudo** to re-run the copy command that failed:

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```


Run the following command again to determine whether the permissions changed:

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

If the user ID and group ID have changed, use the following command to restore them:

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/  
authorized_keys
```

11. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at `/mnt/tempvol`:

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

12. From the Amazon EC2 console, select the volume with the volume ID that you wrote down, choose **Actions**, and then select **Detach Volume**. Wait for the state of the volume to become `available`. (You might need to choose the **Refresh** icon.)
13. With the volume still selected, choose **Actions**, and then select **Attach Volume**. Select the instance ID of the original instance, specify the device name `/dev/sda1`, and then choose **Yes, Attach**.

Warning

If you don't specify `sda1` as the device name, you'll be unable to start the original instance. This is because Amazon EC2 expects the root device volume at `sda1`.

14. Select the original instance, choose **Actions**, select **Instance State**, and then choose **Start**. After the instance enters the `running` state, you can connect to it using the private key file for your new key pair.
15. [EC2-Classic] If the original instance had an associated Elastic IP address before you stopped it, you must re-associate it with the instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down at the beginning of this procedure.
 - c. Choose **Actions**, and then select **Associate Address**.
 - d. Select the ID of the original instance, and then choose **Associate**.
16. (Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, choose **Actions**, select **Instance State**, and then choose **Terminate**.

Amazon EC2 Security Groups for Linux Instances

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

If you need to allow traffic to a Windows instance, see [Amazon EC2 Security Groups for Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Topics

- [Security Groups for EC2-Classic](#) (p. 490)
- [Security Groups for EC2-VPC](#) (p. 490)
- [Security Group Rules](#) (p. 491)
- [Default Security Groups](#) (p. 492)
- [Custom Security Groups](#) (p. 493)
- [Creating a Security Group](#) (p. 494)
- [Describing Your Security Groups](#) (p. 495)
- [Adding Rules to a Security Group](#) (p. 495)
- [Deleting Rules from a Security Group](#) (p. 496)
- [Deleting a Security Group](#) (p. 496)
- [API and Command Overview](#) (p. 497)

If you have requirements that aren't met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Security Groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group.

Note

In EC2-Classic, you can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

Security Groups for EC2-VPC

If you're using EC2-VPC, you must use security groups created specifically for your VPC. When you launch an instance in a VPC, you must specify a security group for that VPC. You can't specify a security group that you created for EC2-Classic when you launch an instance in a VPC.

After you launch an instance in a VPC, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*. You can also change the security groups associated with any other network interface. For more information, see [Changing the Security Group of an Elastic Network Interface](#) (p. 598).

You can change the rules of a security group, and those changes are automatically applied to all instances that are associated with the security group.

Note

In EC2-VPC, you can associate a network interface with up to 5 security groups and add up to 50 rules to a security group.

When you specify a security group for a nondefault VPC to the CLI or the API actions, you must use the security group ID and not the security group name to identify the security group.

Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic. For more information about security groups for EC2-VPC, see [Security Groups for Your](#)

VPC, and [Differences Between Security Groups for EC2-Classic and EC2-VPC](#) in the *Amazon VPC User Guide*.

Security Group Rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups allow all outbound traffic.
- Security group rules are always permissive; you can't create rules that deny access.
- You can add and remove rules at any time. You can't change the outbound rules for EC2-Classic. If you're using the Amazon EC2 console, you can modify existing rules, and you can copy the rules from an existing security group to a new security group.
- When you add or remove rules, your changes are automatically applied to the instances associated with the security group after a short period, depending on the connection tracking for the traffic. For more information, see [Connection Tracking](#) (p. 492).
- Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see [Connection Tracking](#) (p. 492).

Note

If your instance (host A) initiates traffic to host B and uses a protocol other than TCP, UDP, or ICMP, your instance's firewall only tracks the IP address and protocol number for the purpose of allowing response traffic from host B. If host B initiates traffic to your instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules, because it's regarded as response traffic. For VPC security groups, you can control this by modifying your security group's outbound rules to permit only certain types of outbound traffic. Alternatively, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

For each rule, you specify the following:

- The protocol to allow (such as TCP, UDP, or ICMP)
- TCP and UDP, or a custom protocol: The range of ports to allow
- ICMP: The ICMP type and code
- One of the following options for the source (inbound rules) or destination (outbound rules):
 - An individual IP address, in CIDR notation. Be sure to use the `/32` prefix after the IP address; if you use the `/0` prefix after the IP address, this opens the port to everyone. For example, specify the IP address `203.0.113.1` as `203.0.113.1/32`.
 - An IP address range, in CIDR notation (for example, `203.0.113.0/24`).
 - The name (EC2-Classic) or ID (EC2-Classic or EC2-VPC) of a security group. This allows instances associated with the specified security group to access instances associated with this security group. (Note that this does not add rules from the source security group to this security group.) You can specify one of the following security groups:
 - The current security group.
 - EC2-Classic: A different security group for EC2-Classic in the same region.
 - EC2-Classic: A security group for another AWS account in the same region (add the AWS account ID as a prefix; for example, `111122223333/sg-edcd9784`).
 - EC2-VPC: A different security group for the same VPC or a peer VPC.

When you specify a security group as the source or destination for a rule, the rule affects all instances associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see [Amazon EC2 Instance IP Addressing \(p. 570\)](#). If your security group rule references a security group in a peer VPC, and the referenced security group or VPC peering connection is deleted, the rule is marked as stale. For more information, see [Working with Stale Security Group Rules](#) in the *Amazon VPC Peering Guide*.

If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.1 and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.

When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.

Caution

Because you can assign multiple security groups to an instance, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. Therefore, we recommend that you condense your rules as much as possible.

For more information about creating security group rules to ensure that Path MTU Discovery can function correctly, see [Path MTU Discovery \(p. 606\)](#).

Connection Tracking

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. This allows security groups to be stateful — responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa. For example, if you initiate an ICMP `ping` command to your instance from your home computer, and your inbound security group rules allow ICMP traffic, information about the connection (including the port information) is tracked. Response traffic from the instance for the `ping` command is not tracked as new request, but rather as an established connection and is allowed to flow out of the instance, even if your outbound security group rules restrict outbound ICMP traffic.

Not all flows of traffic are tracked. If a security group rule permits TCP or UDP flows for all traffic (0.0.0.0/0) and there is a corresponding rule in the other direction that permits the response traffic, then that flow of traffic is not tracked. The response traffic is therefore allowed to flow based on the inbound or outbound rule that permits the response traffic, and not on tracking information.

An existing flow of traffic that is tracked may not be interrupted when you remove the security group rule that enables that flow. Instead, the flow is interrupted when it's stopped by you or the other host for at least a few minutes (or up to 5 days for established TCP connections). For UDP, this may require terminating actions on the remote side of the flow. An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you remove a rule that allows all inbound SSH traffic (0.0.0.0/0) to the instance, then your existing SSH connections to the instance are immediately dropped.

If you want to ensure that traffic is immediately interrupted when you remove a security group rule, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

Default Security Groups

Your AWS account automatically has a *default security group* per region for EC2-Classic. When you create a VPC, we automatically create a default security group for the VPC. If you don't specify a

different security group when you launch an instance, the instance is automatically associated with the appropriate default security group.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the initial settings for each default security group:

- Allow inbound traffic only from other instances associated with the default security group
- Allow all outbound traffic from the instance

The default security group specifies itself as a source security group in its inbound rules. This is what allows instances associated with the default security group to communicate with other instances associated with the default security group.

You can change the rules for a default security group. For example, you can add an inbound rule to allow SSH connections so that specific hosts can manage the instance.

You can't delete a default security group. If you try to delete the EC2-Classic default security group, you'll get the following error: `Client.InvalidGroup.Reserved: The security group 'default' is reserved.` If you try to delete a VPC default security group, you'll get the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.`

Custom Security Groups

If you don't want all your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server. For instructions that help you create security groups for web servers and database servers, see [Recommended Security Groups](#) in the *Amazon VPC User Guide*.

Note

In EC2-Classic, you can create up to 500 security groups in each region for each account. In EC2-VPC, you can create up to 500 security groups per VPC. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and `. _ - / () # , @ [] + = & ; { } ! $ *`

AWS assigns each security group a unique ID in the form `sg-xxxxxxx`. The following are the initial settings for a security group that you create:

- Allow no inbound traffic
- Allow all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. In EC2-VPC, you can also change its outbound rules.

To allow instances that have the same security group to communicate with each other, you must explicitly add rules for this. The following table describes the rules that you must add to your security group to enable instances in EC2-Classic to communicate with each other.

Inbound			
Source	Protocol	Port Range	Comments
The ID of the security group	ICMP	All	Allow inbound ICMP access from other instances associated with this security group
The ID of the security group	TCP	0 - 65535	Allow inbound TCP access from other instances associated with this security group
The ID of the security group	UDP	0 - 65535	Allow inbound UDP access from other instances associated with this security group

The following table describes the rules that you must add to your security group to enable instances in a VPC to communicate with each other.

Inbound			
Source	Protocol	Port Range	Comments
The ID of the security group	All	All	Allow inbound traffic from other instances associated with this security group

Creating a Security Group

To create a new security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. Specify a name and description for the security group. For **VPC**, choose **No VPC** to create a security group for EC2-Classic, or choose a VPC ID to create a security group for that VPC.
5. You can start adding rules, or you can choose **Create** to create the security group now (you can always add rules later). For more information about adding rules, see [Adding Rules to a Security Group](#) (p. 495).

To copy a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group you want to copy, choose **Actions**, and then **Copy to new**.
4. The **Create Security Group** dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. In the **VPC** list, choose **No VPC** to create a security group for EC2-Classic, or choose a VPC ID to create a security group for that VPC. When you are done, choose **Create**.

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Describing Your Security Groups

To describe your security groups for EC2-Classic

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-Classic**.
4. Select a security group. We display general information in the **Description** tab and inbound rules on the **Inbound** tab.

To describe your security groups for EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-VPC**.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, and outbound rules on the **Outbound** tab.

Adding Rules to a Security Group

When you add a rule to a security group, the new rule is automatically applied to any instances associated with the security group.

To add rules to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose the security group.
4. You can allow web servers to receive all inbound HTTP and HTTPS traffic. On the **Inbound** tab, choose **Edit**. In the dialog, choose **Add Rule**. Select **HTTP** from the **Type** list, and leave the source as **Anywhere** (0.0.0.0/0). Add a similar rule for the HTTPS protocol.
5. To connect to a Linux instance, you need to allow SSH traffic. Choose **Add Rule**, and then select **SSH** from the **Type** list.

In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is 203.0.113.25, specify 203.0.113.25/32 to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24. You can select **My IP** to from the **Source** list to let us automatically populate the field with your computer's IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Caution

If you use 0.0.0.0/0, you enable all IP addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

6. You can allow communication between all instances associated with this security group, or between instances associated with another security group and instances associated with this security group. Choose **Add Rule**, select **All ICMP**, then start entering the ID of the security group

in **Source**; this provides you with a list of security groups. Select the security group from the list. Repeat the steps for the TCP and UDP protocols. Choose **Save** when you are done.

Type ⓘ	Protocol ⓘ	Port Range ⓘ
HTTP	TCP	80
HTTPS	TCP	443
All ICMP	ICMP	0 - 65535
All TCP	TCP	0 - 65535
All UDP	UDP	0 - 65535

Add Rule

7. If you are creating a security group for a VPC, you can also specify outbound rules. For an example, see [Adding and Removing Rules](#) in the *Amazon VPC User Guide*.

Deleting Rules from a Security Group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

To delete a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group.
4. Choose **Edit**, and then choose **Delete** (a cross icon) next to each rule that you need to delete.
5. Choose **Save**.

Deleting a Security Group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

To delete a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Security Groups**.
3. Select a security group and choose **Delete**.
4. Choose **Yes, Delete**.

API and Command Overview

You can perform the tasks described on this page using the command line or an API. For more information about the command line interfaces and a list of available APIs, see [Accessing Amazon EC2](#) (p. 3).

Create a security group

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Add one or more ingress rules to a security group

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Add one or more egress rules to a security group

- [authorize-security-group-egress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

Describe one or more security groups

- [describe-security-groups](#) (AWS CLI)
- [Get-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Modify the security groups for an instance

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Remove one or more ingress rules from a security group

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Remove one or more egress rules from a security group

- [revoke-security-group-egress](#)(AWS CLI)
- [Revoke-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Delete a security group

- [delete-security-group](#) (AWS CLI)
- [Remove-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Controlling Access to Amazon EC2 Resources

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Contents

- [Network Access to Your Instance](#) (p. 498)
- [Amazon EC2 Permission Attributes](#) (p. 498)
- [IAM and Amazon EC2](#) (p. 498)
- [IAM Policies for Amazon EC2](#) (p. 500)
- [IAM Roles for Amazon EC2](#) (p. 536)
- [Authorizing Inbound Traffic for Your Linux Instances](#) (p. 541)

Network Access to Your Instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see [Authorizing Inbound Traffic for Your Linux Instances](#) (p. 541).

Amazon EC2 Permission Attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict permissions for specific users within the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see [Making an AMI Public](#) (p. 78).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see [Sharing an Amazon EBS Snapshot](#) (p. 691).

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources

- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Creating an IAM Group and Users

To create an IAM group

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, enter a name for your group, and then choose **Next Step**.
4. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
5. Choose **Next Step** and then choose **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, choose **Users** and then choose **Create New Users**.
2. In box **1**, enter a user name and then choose **Create**.
3. Choose **Download Credentials** and save your access key in a secure place. You will need your access key for programmatic access to AWS using the AWS CLI, the AWS SDKs, or the HTTP APIs.

Note

You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

After you have downloaded your access key, choose **Close**.

4. Under **User Name**, choose the name of the user you just created.
5. In the **Groups** tab, choose **Add User to Groups**.
6. Select the group you created earlier, and then choose **Add to Groups**.
7. Choose **Security Credentials** and then **Manage Password**.

8. Select **Assign a custom password** and then enter and confirm a password. When you are finished, choose **Apply**.
9. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group

Related Topics

For more information about IAM, see the following:

- [IAM Policies for Amazon EC2 \(p. 500\)](#)
- [IAM Roles for Amazon EC2 \(p. 536\)](#)
- [Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

IAM Policies for Amazon EC2

By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see [Permissions and Policies](#) in the *IAM User Guide*. For more information about managing and creating custom IAM policies, see [Managing IAM Policies](#).

Getting Started

An IAM policy must grant or deny permission to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy Syntax (p. 501)
Define actions in your policy	Actions for Amazon EC2 (p. 502)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 502)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 504)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 507)
Test your policy	Checking that Users Have the Required Permissions (p. 507)
Example policies for a CLI or SDK	Example Policies for Working With the AWS CLI or an AWS SDK (p. 520)

Task	Topic
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 530)

Policy Structure

The following topics explain the structure of an IAM policy.

Topics

- [Policy Syntax \(p. 501\)](#)
- [Actions for Amazon EC2 \(p. 502\)](#)
- [Amazon Resource Names for Amazon EC2 \(p. 502\)](#)
- [Condition Keys for Amazon EC2 \(p. 504\)](#)
- [Checking that Users Have the Required Permissions \(p. 507\)](#)

Policy Syntax

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows:

```
{
  "Statement": [ {
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }
]
}
```

There are various elements that make up a statement:

- **Effect:** The *effect* can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action:** The *action* is the specific API action for which you are granting or denying permission. To learn about specifying *action*, see [Actions for Amazon EC2 \(p. 502\)](#).
- **Resource:** The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the *arn* value, see [Amazon Resource Names for Amazon EC2 \(p. 502\)](#). For more information about which API actions support which ARNs, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 507\)](#). If the API action does not support ARNs, use the * wildcard to specify that all resources can be affected by the action.
- **Condition:** Conditions are optional. They can be used to control when your policy will be in effect. For more information about specifying conditions for Amazon EC2, see [Condition Keys for Amazon EC2 \(p. 504\)](#).

For more information about example IAM policy statements for Amazon EC2, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) (p. 520).

Actions for Amazon EC2

In an IAM policy statement, you can specify any API action from any service that supports IAM. For Amazon EC2, use the following prefix with the name of the API action: `ec2:`. For example: `ec2:RunInstances` and `ec2:CreateImage`.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": ["ec2:action1", "ec2:action2"]
```

You can also specify multiple actions using wildcards. For example, you can specify all actions whose name begins with the word "Describe" as follows:

```
"Action": "ec2:Describe*"
```

To specify all Amazon EC2 API actions, use the `*` wildcard as follows:

```
"Action": "ec2:*"
```

For a list of Amazon EC2 actions, see [Actions](#) in the *Amazon EC2 API Reference*.

Amazon Resource Names for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

Important

Currently, not all API actions support individual ARNs; we'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 507).

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

The service (for example, `ec2`).

region

The region for the resource (for example, `us-east-1`).

account

The AWS account ID, with no hyphens (for example, `123456789012`).

resourceType

The type of resource (for example, `instance`).

resourcePath

A path that identifies the resource. You can use the `*` wildcard in your paths.

For example, you can indicate a specific instance (`i-1234567890abcdef0`) in your statement using its ARN as follows:

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

You can also specify all instances that belong to a specific account by using the * wildcard as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the Resource element as follows:

```
"Resource": "*" 
```

The following table describes the ARNs for each type of resource used by the Amazon EC2 API actions.

Resource Type	ARN
All Amazon EC2 resources	arn:aws:ec2:*
All Amazon EC2 resources owned by the specified account in the specified region	arn:aws:ec2:region:account:*
Customer gateway	arn:aws:ec2:region:account:customer-gateway/cgw-id Where <i>cgw-id</i> is cgw-xxxxxxx
DHCP options set	arn:aws:ec2:region:account:dhcp-options/dhcp-options-id Where <i>dhcp-options-id</i> is dopt-xxxxxxx
Image	arn:aws:ec2:region::image/image-id Where <i>image-id</i> is the ID of the AMI, AKI, or ARI, and <i>account</i> isn't used
Instance	arn:aws:ec2:region:account:instance/instance-id Where <i>instance-id</i> is i-xxxxxxx or i-xxxxxxxxxxxxxxxxxxx
Instance profile	arn:aws:iam::account:instance-profile/instance-profile-name Where <i>instance-profile-name</i> is the name of the instance profile, and <i>region</i> isn't used
Internet gateway	arn:aws:ec2:region:account:internet-gateway/igw-id Where <i>igw-id</i> is igw-xxxxxxx
Key pair	arn:aws:ec2:region:account:key-pair/key-pair-name Where <i>key-pair-name</i> is the key pair name (for example, gsg-keypair)
Network ACL	arn:aws:ec2:region:account:network-acl/nacl-id Where <i>nacl-id</i> is acl-xxxxxxx
Network interface	arn:aws:ec2:region:account:network-interface/eni-id Where <i>eni-id</i> is eni-xxxxxxx

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Resource Type	ARN
Placement group	<i>arn:aws:ec2:region:account:placement-group/placement-group-name</i> Where <i>placement-group-name</i> is the placement group name (for example, <i>my-cluster</i>)
Route table	<i>arn:aws:ec2:region:account:route-table/route-table-id</i> Where <i>route-table-id</i> is <i>rtb-xxxxxxx</i>
Security group	<i>arn:aws:ec2:region:account:security-group/security-group-id</i> Where <i>security-group-id</i> is <i>sg-xxxxxxx</i>
Snapshot	<i>arn:aws:ec2:region::snapshot/snapshot-id</i> Where <i>snapshot-id</i> is <i>snap-xxxxxxx</i> or <i>snap-xxxxxxxxxxxxxxxxxxx</i> , and <i>account</i> isn't used
Subnet	<i>arn:aws:ec2:region:account:subnet/subnet-id</i> Where <i>subnet-id</i> is <i>subnet-xxxxxxx</i>
Volume	<i>arn:aws:ec2:region:account:volume/volume-id</i> Where <i>volume-id</i> is <i>vol-xxxxxxx</i> or <i>vol-xxxxxxxxxxxxxxxxxxx</i>
VPC	<i>arn:aws:ec2:region:account:vpc/vpc-id</i> Where <i>vpc-id</i> is <i>vpc-xxxxxxx</i>
VPC peering connection	<i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i> Where <i>vpc-peering connection-id</i> is <i>pcx-xxxxxxx</i>

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permission to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows:

```
"Resource": [ "arn1", "arn2" ]
```

For more general information about ARNs, see [Amazon Resource Names \(ARN\) and AWS Service Namespaces](#) in the *Amazon Web Services General Reference*. For more information about the resources that are created or modified by the Amazon EC2 actions, and the ARNs that you can use in your IAM policy statements, see [Granting IAM Users Required Permissions for Amazon EC2 Resources](#) in the *Amazon EC2 API Reference*.

Condition Keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permission to be granted, all conditions must be met.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

You can also use placeholders when you specify conditions. For example, you can grant an IAM user permission to use resources with a tag that specifies his or her IAM user name. For more information, see [Policy Variables](#) in the *IAM User Guide*.

Amazon EC2 implements the AWS-wide condition keys (see [Available Keys](#)), plus the following service-specific condition keys. (We'll add support for additional service-specific condition keys for Amazon EC2 later.)

Important

Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the `Resource` element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the `Action` element of your policy to include multiple API actions, then you must use the `...IfExists` condition type to ensure that the condition key is ignored for resources that do not use it. For more information, see [...IfExists Conditions](#) in the *IAM User Guide*.

Condition Key	Key/Value Pair	Evaluation Types
ec2:AccepterVpc	"ec2:AccepterVpc": " <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN for the peer VPC	ARN, Null
ec2:AvailabilityZone	"ec2:AvailabilityZone": " <i>az-api-name</i> " Where <i>az-api-name</i> is the name of the Availability Zone (for example, <i>us-west-2a</i>) To list your Availability Zones, use describe-availability-zones	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized": " <i>optimized-flag</i> " Where <i>optimized-flag</i> is <code>true</code> <code>false</code>	Boolean, Null
ec2:ImageType	"ec2:ImageType": " <i>image-type-api-name</i> " Where <i>image-type-api-name</i> is <code>ami</code> <code>aki</code> <code>ari</code>	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile": " <i>instance-profile-arn</i> " Where <i>instance-profile-arn</i> is the instance profile ARN	ARN, Null
ec2:InstanceType	"ec2:InstanceType": " <i>instance-type-api-name</i> " Where <i>instance-type-api-name</i> is the name of the instance type.	String, Null
ec2:Owner	"ec2:Owner": " <i>account-id</i> " Where <i>account-id</i> is <code>amazon</code> <code>aws-marketplace</code> <code>aws-account-id</code>	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot": " <i>snapshot-arn</i> " Where <i>snapshot-arn</i> is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume": " <i>volume-arn</i> " Where <i>volume-arn</i> is the volume ARN	ARN, Null

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Condition Key	Key/Value Pair	Evaluation Types
ec2:PlacementGroup	"ec2:PlacementGroup":" <i>placement-group-arn</i> " Where <i>placement-group-arn</i> is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":" <i>placement-group-strategy</i> " Where <i>placement-group-strategy</i> is <code>cluster</code>	String, Null
ec2:ProductCode	"ec2:ProductCode":" <i>product-code</i> " Where <i>product-code</i> is the product code	String, Null
ec2:Public	"ec2:Public":" <i>public-flag</i> " Where <i>public-flag</i> for an AMI is <code>true</code> <code>false</code>	Boolean, Null
ec2:Region	"ec2:Region":" <i>region-name</i> " Where <i>region-name</i> is the name of the region (for example, <code>us-west-2</code>). To list your regions, use describe-regions .	String, Null
ec2:RequesterVpc	"ec2:RequesterVpc":" <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN for the requester's VPC	ARN, Null
ec2:ResourceTag/ <i>tag-key</i>	"ec2:ResourceTag/ <i>tag-key</i> ":" <i>tag-value</i> " Where <i>tag-key</i> and <i>tag-value</i> are the tag-key pair	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType":" <i>root-device-type-name</i> " Where <i>root-device-type-name</i> is <code>ebs</code> <code>instance-store</code>	String, Null
ec2:Subnet	"ec2:Subnet":" <i>subnet-arn</i> " Where <i>subnet-arn</i> is the subnet ARN	ARN, Null
ec2:Tenancy	"ec2:Tenancy":" <i>tenancy-attribute</i> " Where <i>tenancy-attribute</i> is <code>default</code> <code>dedicated</code> <code>host</code>	String, Null
ec2:VolumeIops	"ec2:VolumeIops":" <i>volume-iops</i> " Where <i>volume-iops</i> is the input/output operations per second (IOPS); the range is 100 to 20,000	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize":" <i>volume-size</i> " Where <i>volume-size</i> is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType":" <i>volume-type-name</i> " Where <i>volume-type-name</i> is <code>gp2</code> for General Purpose SSD volumes, <code>io1</code> for Provisioned IOPS SSD volumes, <code>st1</code> for Throughput Optimized HDD volumes, <code>sc1</code> for Cold HDD volumes, or <code>standard</code> for Magnetic volumes.	String, Null

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Condition Key	Key/Value Pair	Evaluation Types
ec2:Vpc	"ec2:Vpc": "vpc-arn" Where <i>vpc-arn</i> is the VPC ARN	ARN, Null

For information about which condition keys you can use with which Amazon EC2 resources, on an action-by-action basis, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 507). For example policy statements for Amazon EC2, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) (p. 520).

Checking that Users Have the Required Permissions

After you've created an IAM policy, we recommend that you check whether it grants users the permissions to use the particular API actions and resources they need before you put the policy into production.

First, create an IAM user for testing purposes, and then attach the IAM policy that you created to the test user. Then, make a request as the test user.

If the action that you are testing creates or modifies a resource, you should make the request using the `DryRun` parameter (or run the CLI command with the `--auth-dry-run` option). In this case, the call completes the authorization check, but does not complete the operation. For example, you can check whether the user can terminate a particular instance without actually terminating it. If the test user has the required permissions, the request returns `DryRunOperation`; otherwise, it returns `UnauthorizedOperation`.

If the policy doesn't grant the user the permissions that you expected, or is overly permissive, you can adjust the policy as needed and retest until you get the desired results.

Important

It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

If an authorization check fails, the request returns an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` action. For more information, see [DecodeAuthorizationMessage](#) in the *AWS Security Token Service API Reference*, and [decode-authorization-message](#) in the *AWS Command Line Interface Reference*.

Supported Resource-Level Permissions for Amazon EC2 API Actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permission to launch instances, but only of a specific type, and only using a specific AMI.

The following table describes the Amazon EC2 API actions that currently support resource-level permissions, as well as the supported resources (and their ARNs) and condition keys for each action. When specifying an ARN, you can use the `*` wildcard in your paths; for example, when you cannot or do not want to specify exact resource IDs. For examples of using wildcards, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) (p. 520).

Important

If an Amazon EC2 API action is not listed in this table, then it does not support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

can grant users permission to use the action, but you have to specify a * for the resource element of your policy statement. For an example of how to do this, see [1: Read-only access \(p. 520\)](#). We'll add support for additional actions, ARNs, and condition keys later. For a list of Amazon EC2 API actions that currently do not support resource-level permissions, see [Unsupported Resource-Level Permissions](#) in the *Amazon EC2 API Reference*.

API Action	Resources	Condition Keys
AcceptVpcPeeringConnections	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is a VPC owned by the acceptor.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
AttachClassicLinkVpc	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group-id Where the security group is the security group for the VPC.	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
AttachVolume	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
CreateVpcPeeringConnection	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is a requester VPC.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/*	ec2:AccepterVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway arn:aws:ec2:region:account:customer-gateway/* arn:aws:ec2:region:account:customer-gateway/cgw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteDhcpOptions	DHCP options set arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options-id	ec2:Region ec2:ResourceTag/tag-key
DeleteInternetGateway	Internet gateway arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteNetworkAcl	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteNetworkAclEntry	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRoute	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
DeleteRouteTable	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteSecurityGroup	Security group arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteVolume	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AccepterVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
DetachClassicLinkVpc	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
DetachVolume	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DisableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
EnableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
GetConsoleScreenshot	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RebootInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC peering connection arn:aws:ec2:region:account:vpc- peering-connection/* arn:aws:ec2:region:account:vpc- peering-connection/vpc-peering- connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
RevokeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RevokeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
RunInstances	Image <i>arn:aws:ec2:region::image/*</i> <i>arn:aws:ec2:region::image/image-id</i>	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	Instance <i>arn:aws:ec2:region:account:instance/*</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	Key pair <i>arn:aws:ec2:region:account:key-pair/*</i> <i>arn:aws:ec2:region:account:key-pair/key-pair-name</i>	ec2:Region
	Network interface <i>arn:aws:ec2:region:account:network-interface/*</i> <i>arn:aws:ec2:region:account:network-interface/eni-id</i>	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	Placement group <i>arn:aws:ec2:region:account:placement-group/*</i> <i>arn:aws:ec2:region:account:placement-group/placement-group-name</i>	ec2:Region ec2:PlacementGroupStrategy
	Security group <i>arn:aws:ec2:region:account:security-group/*</i> <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
	Snapshot arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/snapshot- <i>id</i>	ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/tag-key ec2:VolumeSize
	Subnet arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/ <i>subnet-id</i>	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	Volume arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
StartInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

API Action	Resources	Condition Keys
StopInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
TerminateInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Resource-Level Permissions for RunInstances

The `RunInstances` API action launches one or more instances, and creates and uses a number of Amazon EC2 resources. The action requires an AMI and creates an instance; and the instance must be associated with a security group. Launching into a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. The user must have permission to use these resources, so they must be specified in the `Resource` element of any policy that uses resource-level permissions for the `ec2:RunInstances` action. If you don't intend to use resource-level permissions with the `ec2:RunInstances` action, you can specify the `*` wildcard in the `Resource` element of your statement instead of individual ARNs.

If you are using resource-level permissions, the following table describes the minimum resources required to use the `ec2:RunInstances` action.

Type of launch	Resources required	Condition keys
Launching into EC2-Classic using an instance store-backed AMI	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Type of launch	Resources required	Condition keys
		ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
Launching into EC2-Classic using an Amazon EBS-backed AMI	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Type of launch	Resources required	Condition keys
	<code>arn:aws:ec2:region:account:volume*</code>	<ul style="list-style-type: none"> ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
Launching into a VPC using an instance store-backed AMI	<code>arn:aws:ec2:region:account:instance*</code>	<ul style="list-style-type: none"> ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	<code>arn:aws:ec2:region::image/* (or a specific AMI ID)</code>	<ul style="list-style-type: none"> ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	<code>arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)</code>	<ul style="list-style-type: none"> ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	<code>arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)</code>	<ul style="list-style-type: none"> ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

Type of launch	Resources required	Condition keys
	arn:aws:ec2:region:account:subnet/* (or a specific subnet ID)	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
Launching into a VPC using an Amazon EBS-backed AMI	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc

Type of launch	Resources required	Condition keys
	<code>arn:aws:ec2:region:account:volumes*</code>	<code>ec2:AvailabilityZone</code> <code>ec2:ParentSnapshot</code> <code>ec2:Region</code> <code>ec2:Volumeops</code> <code>ec2:VolumeSize</code> <code>ec2:VolumeType</code>
	<code>arn:aws:ec2:region:account:subnets*</code> (or a specific subnet ID)	<code>ec2:AvailabilityZone</code> <code>ec2:Region</code> <code>ec2:ResourceTag/tag-key</code> <code>ec2:Vpc</code>

We recommend that you also specify the key pair resource in your policy — even though it's not required to launch an instance, you cannot connect to your instance without a key pair. For examples of using resource-level permissions with the `ec2:RunInstances` action, see [4: Launching instances \(RunInstances\)](#) (p. 523).

For additional information about resource-level permissions in Amazon EC2, see the following AWS Security Blog post: [Demystifying EC2 Resource-Level Permissions](#).

Example Policies for Working With the AWS CLI or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI or an AWS SDK. For example policies for working in the Amazon EC2 console, see [Example Policies for Working in the Amazon EC2 Console](#) (p. 530). For examples of IAM policies specific to Amazon VPC, see [Controlling Access to Amazon VPC Resources](#)

- [1: Read-only access](#) (p. 520)
- [2: Working with instances](#) (p. 521)
- [3: Working with volumes](#) (p. 522)
- [4: Launching instances \(RunInstances\)](#) (p. 523)
- [5: Working with ClassicLink](#) (p. 527)
- [6: Working with Reserved Instances](#) (p. 529)

Example 1: Read-only access

The following policy grants users permission to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 507).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:Describe*",
    "Resource": "*"
  }]
}
```

Example 2: Working with instances

a. Describe, launch, stop, start, and terminate all instances

The following policy grants users permission to use the API actions specified in the `Action` element. The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 507\)](#).

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
      "ec2:DescribeAvailabilityZones",
      "ec2:RunInstances", "ec2:TerminateInstances",
      "ec2:StopInstances", "ec2:StartInstances"
    ],
    "Resource": "*"
  }]
}
```

b. Describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances `i-1234567890abcdef0` and `i-0598c7d356eba48d7`, and to terminate only instances in the US East (N. Virginia) Region (`us-east-1`) with the resource tag `"purpose=test"`.

The first statement uses a `*` wildcard for the `Resource` element to indicate that users can specify all resources with the action; in this case, they can list all instances. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, `ec2:DescribeInstances`). For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 507\)](#).

The second statement uses resource-level permissions for the `StopInstances` and `StartInstances` actions. The specific instances are indicated by their ARNs in the `Resource` element.

The third statement allows users to terminate all instances in the US East (N. Virginia) Region (`us-east-1`) that belong to the specified AWS account, but only where the instance has the tag `"purpose=test"`. The `Condition` element qualifies when the policy statement is in effect.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
      }
    }
  ]
}
```

Example 3. Working with volumes

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a `Condition` element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag `volume_user=iam-user-name` to instances with the tag `department=dev`, and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/department": "dev"
      }
    }
  ]
}
```

```
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/volume_user": "${aws:username}"
      }
    }
  }
]
```

Example 4: Launching instances (RunInstances)

The `RunInstances` API action launches one or more instances. `RunInstances` requires an AMI and creates an instance; and users can specify a key pair and security group in the request. Launching into EC2-VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permission to use these Amazon EC2 resources. The caller can also configure the instance using optional parameters to `RunInstances`, such as the instance type and a subnet. You can create a policy statement that requires users to specify an optional parameter, or restricts users to particular values for a parameter. The examples in this section demonstrate some of the many possible ways that you can control the configuration of an instance that a user can launch.

Note that by default, users don't have permission to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see [2: Working with instances \(p. 521\)](#).

a. AMI

The following policy allows users to launch instances using only the AMIs that have the specified tag, `department=dev`, associated with them. The users can't launch instances using other AMIs because the `Condition` element of the first statement requires that users specify an AMI that has this tag. The users also can't launch into a subnet, as the policy does not grant permissions for the subnet and network interface resources. They can, however, launch into EC2-Classical. The second statement uses a wildcard to enable users to create instance resources, and requires users to specify the key pair `project_keypair` and the security group `sg-1a2b3c4d`. Users are still able to launch instances without a key pair.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/department": "dev"
      }
    }
  ]
}
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

```
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/project_keypair",
      "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
    ]
  }
]
```

Alternatively, the following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so), and the users can't launch an instance into a subnet.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-9e1670f7",
      "arn:aws:ec2:region::image/ami-45cf5c3c",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The `Condition` element of the first statement tests whether `ec2:Owner` is `amazon`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so). The users are able to launch an instance into a subnet.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:Owner": "amazon"
      }
    }
  }
],
  {
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

```
"Effect": "Allow",
"Action": "ec2:RunInstances",
"Resource": [
  "arn:aws:ec2:region:account:instance/*",
  "arn:aws:ec2:region:account:subnet/*",
  "arn:aws:ec2:region:account:volume/*",
  "arn:aws:ec2:region:account:network-interface/*",
  "arn:aws:ec2:region:account:key-pair/*",
  "arn:aws:ec2:region:account:security-group/*"
]
}
```

b. Instance type

The following policy allows users to launch instances using only the `t2.micro` or `t2.small` instance type, which you might do to control costs. The users can't launch larger instances because the Condition element of the first statement tests whether `ec2:InstanceType` is either `t2.micro` or `t2.small`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:InstanceType": ["t2.micro", "t2.small"]
      }
    }
  }],
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
```

Alternatively, you can create a policy that denies users permission to launch any instances except `t2.micro` and `t2.small` instance types.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
IAM Policies**

```
"Resource": [
  "arn:aws:ec2:region:account:instance/*"
],
"Condition": {
  "StringNotEquals": {
    "ec2:InstanceType": ["t2.micro", "t2.small"]
  }
}
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:region::image/ami-*",
    "arn:aws:ec2:region:account:network-interface/*",
    "arn:aws:ec2:region:account:instance/*",
    "arn:aws:ec2:region:account:subnet/*",
    "arn:aws:ec2:region:account:volume/*",
    "arn:aws:ec2:region:account:key-pair/*",
    "arn:aws:ec2:region:account:security-group*"
  ]
}
]
```

c. Subnet

The following policy allows users to launch instances using only the specified subnet, `subnet-12345678`. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so). Users are still able to launch instances into EC2-Classic.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:subnet/subnet-12345678",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }
]
```

Alternatively, you could create a policy that denies users permission to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet `subnet-12345678` is specified. This denial overrides any other policies that are created to allow launching instances into other subnets. Users are still able to launch instances into EC2-Classic.

```
{
  "Version": "2012-10-17",
```

```
"Statement": [{
  "Effect": "Deny",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:region:account:network-interface/*"
  ],
  "Condition": {
    "ArnNotEquals": {
      "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:region::image/ami-*",
    "arn:aws:ec2:region:account:network-interface/*",
    "arn:aws:ec2:region:account:instance/*",
    "arn:aws:ec2:region:account:subnet/*",
    "arn:aws:ec2:region:account:volume/*",
    "arn:aws:ec2:region:account:key-pair/*",
    "arn:aws:ec2:region:account:security-group*"
  ]
}
]
```

Example 5. Working with ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc`, and `ec2:DetachClassicLinkVpc` actions to control how users are able to use those actions. Resource-level permissions are not supported for `ec2:Describe*` actions.

a. Full permission to work with ClassicLink

The following policy grants users permission to view ClassicLink-enabled VPCs and linked EC2-Classic instances, to enable and disable a VPC for ClassicLink, and to link and unlink instances from a ClassicLink-enabled VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",
      "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",
      "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"
    ],
    "Resource": "*"
  }
]
```

b. Enable and disable a VPC for ClassicLink

The following policy allows user to enable and disable VPCs for ClassicLink that have the specific tag 'purpose=classiclink'. Users cannot enable or disable any other VPCs for ClassicLink.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcClassicLink",
      "Resource": "arn:aws:ec2:region:account:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "classiclink"
        }
      }
    }
  ]
}
```

c. Link instances

The following policy grants users permission to link instances to a VPC only if the instance is an `m3.large` instance type. The second statement allows users to use the VPC and security group resources, which are required to link an instance to a VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": "arn:aws:ec2:region:account:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:InstanceType": "m3.large"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:vpc/*",
        "arn:aws:ec2:region:account:security-group*"
      ]
    }
  ]
}
```

The following policy grants users permission to link instances to a specific VPC (`vpc-1a2b3c4d`) only, and to associate only specific security groups from the VPC to the instance (`sg-1122aabb` and `sg-aabb2233`). Users cannot link an instance to any other VPC, and they cannot specify any other of the VPC security groups to associate with the instance in the request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
"Effect": "Allow",
"Action": "ec2:AttachClassicLinkVpc",
"Resource": [
  "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",
  "arn:aws:ec2:region:account:instance/*",
  "arn:aws:ec2:region:account:security-group/sg-1122aabb",
  "arn:aws:ec2:region:account:security-group/sg-aabb2233"
]
}
```

d. Unlink instances

The following grants users permission to unlink any linked EC2-Classical instance from a VPC, but only if the instance has the tag "unlink=true". The second statement grants users permission to use the VPC resource, which is required to unlink an instance from a VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:DetachClassicLinkVpc",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/unlink": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DetachClassicLinkVpc",
    "Resource": [
      "arn:aws:ec2:region:account:vpc/*"
    ]
  }
]
```

Example 6. Working with Reserved Instances

The following policy gives users permission to view, modify, and purchase Reserved Instances in your account.

It is not possible to set resource-level permissions for individual Reserved Instances. This policy means that users have access to all the Reserved Instances in the account.

The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with the action; in this case, they can list and modify all Reserved Instances in the account. They can also purchase Reserved Instances using the account credentials. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions.

```
{
  "Version": "2012-10-17",
  "Statement": [{
```



```
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
      "ec2:PurchaseReservedInstancesOffering",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeReservedInstancesOfferings"
    ],
    "Resource": "*"
  }
]
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
      "ec2:DescribeAvailabilityZones"
    ],
    "Resource": "*"
  }]
}
```

Example Policies for Working in the Amazon EC2 Console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

- [1: Read-only access \(p. 531\)](#)
- [2: Using the EC2 launch wizard \(p. 532\)](#)
- [3: Working with volumes \(p. 533\)](#)
- [4: Working with security groups \(p. 534\)](#)
- [5: Working with Elastic IP addresses \(p. 535\)](#)
- [6: Working with Reserved Instances \(p. 536\)](#)

Note

To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#). If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the [DecodeAuthorizationMessage](#) API action for AWS STS, or the `decode-authorization-message` command in the AWS CLI.

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: [Granting Users Permission to Work in the Amazon EC2 Console](#).

Example 1: Read-only access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: [1: Read-only access \(p. 520\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

a. View instances, AMIs, and snapshots

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the `ec2:Describe` API action with specific `ec2:Describe` actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The `ec2:DescribeTags` action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action if you want users to view only private AMIs.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeTags", "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
  }]
}
```

Note

Currently, the Amazon EC2 `ec2:Describe*` API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the `Resource` element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 507\)](#).

b. View instances and CloudWatch metrics

The following policy allows users to view instances in the Amazon EC2 console, as well as CloudWatch alarms and metrics in the **Monitoring** tab of the **Instances** page. The Amazon EC2 console uses the CloudWatch API to display the alarms and metrics, so you must grant users permission to use the `cloudwatch:DescribeAlarms` and `cloudwatch:GetMetricStatistics` actions.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:GetMetricStatistics"
    ],
    "Resource": "*"
  }]
}
```

instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

```
Statement: [
  {
    Action: [
      "ec2:DescribeInstances",
      "ec2:DescribeImages",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    Effect: "Allow"
  }
]
```

Example 2: Using the EC2 Launch wizard

The first statement grants users permission to provide the options in the launch wizard as demonstrated in the example above. The second statement grants users permission to use the network interface, volume, elastic block store, and subnet resources for EC2-Classic to view and select a specific resource to attach to an instance. For more information about using the `ec2:RunInstances` action, see [Using Amazon EC2 to launch instances in a VPC](#). The third and fourth statements grant users permission to view and select existing network interfaces for the selected subnet. The fifth statement grants users permission to view and select existing AMI resources and only grant a specific permission for the AMI ID. Users can only launch in the sa-east-1 region. If users select a different region, or select a different instance type, AMI, or subnet in the launch wizard, the wizard's suggested `launch-wizard-x` security group. However, this action alone only creates the security group; it does not add or modify any rules. To add inbound rules, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To add inbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To modify or delete existing rules, users must have permission to use the `ec2:RevokeSecurityGroupIngress` API action.

Important: Subnet resources are not available in EC2-Classic. To launch instances in EC2-Classic, you must specify a VPC ID. The launch wizard does not support launching into a VPC to view and select existing network interfaces for the selected subnet.

- `ec2:CreateTags`: To add a tag to the instance. By default, the launch wizard attempts to add a tag with a key of `aws:ec2:instance-type` and a value of the instance type. Users that do not have permission to use this action will encounter a warning that this tag could not be applied to an instance; however, this does not affect the success of the launch, so you should only grant users permission to use this action if it is absolutely necessary.

```
{
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1:111122223333:instance/*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:InstanceType": "m1.small"
    }
  }
},
{
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1::image/ami-*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:Owner": "amazon"
    }
  }
}
]
```

Example 3: Working with volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "purpose=test", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the `ec2:DescribeInstances` action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the `ec2:DescribeVolumeStatus` and `ec2:DescribeAvailabilityZones` actions are necessary to ensure that volumes display correctly in the console.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",
      "ec2:DescribeAvailabilityZones", "ec2:CreateVolume",
      "ec2:DescribeInstances"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/purpose": "test"
      }
    }
  }
],
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:volume/*"
  }
]
}
```

Department: .
},
{
a. View security groups and add and remove rules

```
"Action": [
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:RevokeSecurityGroupIngress",
    "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
],
"Resource": [
    "arn:aws:ec2:region:111122223333:security-group/*"
]
```

Example 4: Working with security groups

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at the least the following API actions:

- **ec2:CreateSecurityGroup**: To create a new security group.

b. Working with the Create Security Group dialog box

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- **ec2:AuthorizeSecurityGroupIngress**: To add inbound rules.
- **ec2:AuthorizeSecurityGroupEgress**: To add outbound rules to VPC security groups.
- **ec2:RevokeSecurityGroupIngress**: To modify or delete existing inbound rules. This is useful if you want to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- **ec2:RevokeSecurityGroupEgress**: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- **ec2>DeleteSecurityGroup**: To cater for scenarios where invalid rules cannot be saved. If a user creates a security group with an invalid rule, the console first creates the security group, then attempts to add the rules to it. After that fails, the security group is deleted. The user remains in the **Create Security Group** dialog box, where an error is displayed. The rules remain listed, so the user can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user is returned to the console.

The following policy grants users permission to use the **Create Security Group** dialog box and to create inbound and outbound rules for security groups that are associated with a specific VPC (vpc-1a2b3c4d). Users can create security groups for EC2-Classical or another VPC, but they cannot currently add rules to them. Similarly, users cannot add rules to any other resource-level permissions based on VPC IDs. This policy also grants users permission to delete security groups and the console. This makes it easier for users to actions to security groups, such as create, add inbound rules. This policy also grants users permission to delete security groups that are associated with VPC vpc-1a2b3c4d.

```
"Action": [
    "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup",
    "ec2:DescribeVpcs"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2>DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
    "Condition": {
        "ArnEquals": {
            "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
        }
    }
}
]
```

Example 5: Working with Elastic IP addresses

The following policy grants users permission to view Elastic IP addresses in the Amazon EC2 console. The console uses the `ec2:DescribeInstances` action to display information about instances with which the Elastic IP addresses are associated. If users are not granted permission to use this action, the Elastic IP addresses page cannot load properly.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeAddresses", "ec2:DescribeInstances"
    ],
    "Resource": "*"
  }
]
```

To allow users to work with Elastic IP addresses, you can add the following actions to your policy

- `ec2:AllocateAddress`: To allocate an address for use in VPC or EC2-Classic.
- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:DescribeNetworkInterfaces`: To work with the **Associate Address** dialog box. The dialog box displays the available network interfaces to which you can associate an Elastic IP address, and will not open if users are not granted permission to use this action. However, this only applies to EC2-VPC; this action is not required for associating an Elastic IP address to an instance in EC2-Classic.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface,

Example 6: Working with Reserved Instances

The following policy can be attached to an IAM user. It gives the user access to view and modify Reserved Instances in your account, as well as purchase new Reserved Instances in the AWS Management Console.

This policy allows users to view all the Reserved Instances, as well as On-Demand Instances, in the account. It's not possible to set resource-level permissions for individual Reserved Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
      "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeInstances",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeReservedInstancesOfferings"
    ],
    "Resource": "*"
  }
]
```

The `ec2:DescribeAvailabilityZones` action is necessary to ensure that the Amazon EC2 console can display information about the Availability Zones in which you can purchase Reserved Instances. The `ec2:DescribeInstances` action is not required, but ensures that the user can view the instances in the account and purchase reservations to match the correct specifications.

You can adjust the API actions to limit user access, for example removing `ec2:DescribeInstances` and `ec2:DescribeAvailabilityZones` means the user has read-only access.

IAM Roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting them from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instances.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that needs to use a bucket in Amazon S3.

Note

Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the console, the console creates an instance profile automatically and gives it the same name as the role it corresponds to. If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, and you might give them different names. To launch an instance with an IAM role, you specify the name of its instance profile. When you launch an instance using the Amazon EC2 console, you can select a role to associate with the instance; however, the list that's displayed is actually a list of instance profile names. For more information, see [Instance Profiles](#) in the *IAM User Guide*.

You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you make a change to a role, the change is propagated to all instances, simplifying credential management.

Note

You can't assign a role to an existing instance; you can only specify a role when you launch a new instance.

For more information about creating and using IAM roles, see [Roles](#) in the *IAM User Guide*.

Topics

- [Retrieving Security Credentials from Instance Metadata](#) (p. 537)
- [Granting an IAM User Permission to Launch an Instance with an IAM Role](#) (p. 538)
- [Creating an IAM Role Using the Console](#) (p. 538)
- [Launching an Instance with an IAM Role Using the Console](#) (p. 539)
- [Creating an IAM Role Using the AWS CLI](#) (p. 539)
- [Launching an Instance with an IAM Role Using the AWS CLI](#) (p. 541)

Retrieving Security Credentials from Instance Metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/role-name`. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes prior to the expiration of the old credentials.

Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2012-04-27T22:39:16Z"
```



```
}
```

For applications, AWS CLI, and Tools for Windows PowerShell commands that run on the instance, you do not have to explicitly get the temporary security credentials — the AWS SDKs, AWS CLI, and Tools for Windows PowerShell automatically get the credentials from the EC2 instance metadata service and use them. To make a call outside of the instance using temporary security credentials (for example, to test IAM policies), you must provide the access key, secret key, and the session token. For more information, see [Using Temporary Security Credentials to Request Access to AWS Resources](#) in the *IAM User Guide*.

For more information about instance metadata, see [Instance Metadata and User Data](#) (p. 337).

Granting an IAM User Permission to Launch an Instance with an IAM Role

To enable an IAM user to launch an instance with an IAM role, you must grant the user permission to pass the role to the instance.

For example, the following IAM policy grants users permission to launch an instance with the IAM role named `s3access`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/s3access"
  }]
}
```

Alternatively, you could grant IAM users access to all your roles by specifying the resource as `***` in this policy. However, consider whether users who launch instances with your roles (ones that exist or that you'll create later on) might be granted permissions that they don't need or shouldn't have.

For more information, see [Permissions Required for Using Roles with Amazon EC2](#) in the *IAM User Guide*.

Creating an IAM Role Using the Console

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the IAM console

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. On the **Set Role Name** page, enter a name for the role and choose **Next Step**.
4. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.
5. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
6. Review the role information, edit the role as needed, and then choose **Create Role**.

Launching an Instance with an IAM Role Using the Console

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the *IAM User Guide*.

To launch an instance with an IAM role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. Select an AMI, then select an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select the IAM role you created from the **IAM role** list.

Note

The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or choose **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.
6. Review your settings, then choose **Launch** to choose a key pair and launch your instance.
7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Creating an IAM Role Using the AWS CLI

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the AWS CLI

- Create an IAM role with a policy that allows the role to use an Amazon S3 bucket.
 - a. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
}
```

- b. Create the `s3access` role and specify the trust policy that you created.

```
$ aws iam create-role --role-name s3access --assume-role-policy-  
document file://ec2-role-trust-policy.json  
{  
  "Role": {  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": "sts:AssumeRole",  
          "Effect": "Allow",  
          "Principal": {  
            "Service": "ec2.amazonaws.com"  
          }  
        }  
      ]  
    },  
    "RoleId": "AROAIIZKPBKS2LEXAMPLE",  
    "CreateDate": "2013-12-12T23:46:37.247Z",  
    "RoleName": "s3access",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:role/s3access"  
  }  
}
```

- c. Create an access policy and save it in a text file named `ec2-role-access-policy.json`. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["s3:*"],  
      "Resource": ["*"]  
    }  
  ]  
}
```

- d. Attach the access policy to the role.

```
$ aws iam put-role-policy --role-name s3access --policy-name S3-  
Permissions --policy-document file://ec2-role-access-policy.json
```

- e. Create an instance profile named `s3access-profile`.

```
$ aws iam create-instance-profile --instance-profile-name s3access-  
profile  
{  
  "InstanceProfile": {  
    "InstanceProfileId": "AIPAJTLPJLEGREXAMPLE",  
    "Roles": [],  
    "CreateDate": "2013-12-12T23:53:34.093Z",  
    "InstanceProfileName": "s3access-profile",
```

```
"Path": "/",  
"Arn": "arn:aws:iam::123456789012:instance-profile/s3access-  
profile"  
  }  
}
```

- f. Add the `s3access` role to the `s3access-profile` instance profile.

```
$ aws iam add-role-to-instance-profile --instance-profile-name  
s3access-profile --role-name s3access
```

For more information about these commands, see [create-role](#), [put-role-policy](#), and [create-instance-profile](#) in the *AWS Command Line Interface Reference*.

Launching an Instance with an IAM Role Using the AWS CLI

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the *IAM User Guide*.

To launch an instance with an IAM role using the AWS CLI

1. Launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
$ aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile  
Name="s3access-profile" --key-name my-key-pair --security-groups my-  
security-group --subnet-id subnet-1a2b3c4d
```

For more information, see [run-instances](#) in the *AWS Command Line Interface Reference*.

2. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-  
credentials/role_name
```

Authorizing Inbound Traffic for Your Linux Instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance using SSH. If your instance is a web server, you can allow all IP addresses to access your instance via HTTP, so that external users can browse the content on your web server.

To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize SSH traffic from your computer's public IP address. To allow SSH traffic from additional IP address ranges, add another rule for each range you need to authorize.

If you need to enable network access to a Windows instance, see [Authorizing Inbound Traffic for Your Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Before You Start

Decide who requires access to your instance; for example, a single host or a specific network that you trust. In this case, we use your local system's public IP address. You can get the public IP address of your local computer using a service. For example, we provide the following service: <http://checkip.amazonaws.com>. To locate another service that provides your IP address, use the search phrase "what is my IP address". If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Caution

If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

For more information about security groups, see [Amazon EC2 Security Groups for Linux Instances](#) (p. 489).

Adding a Rule for Inbound SSH Traffic to a Linux Instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your Linux instance from your IP address using SSH.

To add a rule to a security group for inbound SSH traffic using the console

1. In the navigation pane of the Amazon EC2 console, choose **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Choose **view rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, choose **Security Groups**. Select one of the security groups associated with your instance.
3. In the details pane, on the **Inbound** tab, choose **Edit**. In the dialog, choose **Add Rule**, and then choose **SSH** from the **Type** list.
4. In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

For information about finding your IP address, see [Before You Start](#) (p. 542).

5. Choose **Save**.

To add a rule to a security group using the command line

You can use one of the following commands. Be sure to run this command on your local system, not on the instance itself. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

Assigning a Security Group to an Instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch your AWS resources, such as instances, into your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using AWS's scalable infrastructure. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the Internet. You can connect your VPC to your own corporate data center, making the AWS cloud an extension of your data center. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see the [Amazon VPC User Guide](#).

Your account may support both the EC2-VPC and EC2-Classic platforms, on a region-by-region basis. If you created your account after 2013-12-04, it supports EC2-VPC only. To find out which platforms your account supports, see [Supported Platforms](#) (p. 549). If your account supports EC2-VPC only, we create a *default VPC* for you. A default VPC is a VPC that is already configured and ready for you to use. You can launch instances into your default VPC immediately. For more information, see [Your Default VPC and Subnets](#) in the *Amazon VPC User Guide*. If your account supports EC2-Classic and EC2-VPC, you can launch instances into either platform. Regardless of which platforms your account supports, you can create your own *nondefault VPC*, and configure it as you need.

Contents

- [Benefits of Using a VPC](#) (p. 543)
- [Differences Between EC2-Classic and EC2-VPC](#) (p. 544)
- [Sharing and Accessing Resources Between EC2-Classic and EC2-VPC](#) (p. 546)
- [Instance Types Available Only in a VPC](#) (p. 548)
- [Amazon VPC Documentation](#) (p. 548)
- [Supported Platforms](#) (p. 549)
- [ClassicLink](#) (p. 550)
- [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC](#) (p. 560)

Benefits of Using a VPC

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IP addresses to your instances that persist across starts and stops

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Differences Between EC2-Classic and EC2-VPC**

- Assign multiple IP addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware

Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between instances launched in EC2-Classic, instances launched in a default VPC, and instances launched in a nondefault VPC.

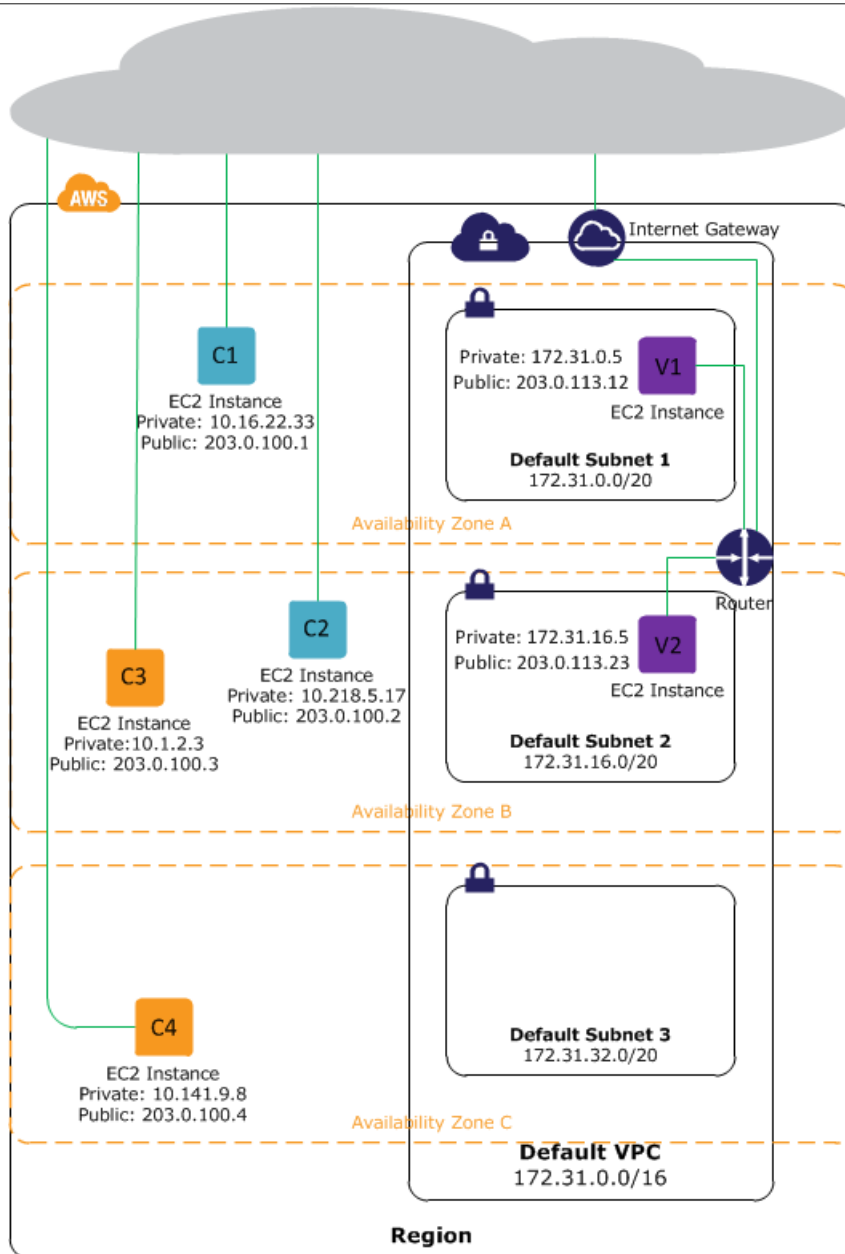
Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IP address (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance launched in a default subnet receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.
Multiple private IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts. You can create up to 500 security groups in each region.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.
Security group association	You can assign an unlimited number of security groups to an instance when you launch it.	You can assign up to 5 security groups to an instance. You can assign security groups to your instance	You can assign up to 5 security groups to an instance. You can assign security groups to your instance

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Differences Between EC2-Classic and EC2-VPC**

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
	You can't change the security groups of your running instance. You can either modify the rules of the assigned security groups, or replace the instance with a new one (create an AMI from the instance, launch a new instance from this AMI with the security groups that you need, disassociate any Elastic IP address from the original instance and associate it with the new instance, and then terminate the original instance).	when you launch it and while it's running.	when you launch it and while it's running.
Security group rules	You can add rules for inbound traffic only. You can add up to 100 rules to a security group.	You can add rules for inbound and outbound traffic. You can add up to 50 rules to a security group.	You can add rules for inbound and outbound traffic. You can add up to 50 rules to a security group.
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.
Accessing the Internet	Your instance can access the Internet. Your instance automatically receives a public IP address, and can access the Internet directly through the AWS network edge.	By default, your instance can access the Internet. Your instance receives a public IP address by default. An Internet gateway is attached to your default VPC, and your default subnet has a route to the Internet gateway.	By default, your instance cannot access the Internet. Your instance doesn't receive a public IP address by default. Your VPC may have an Internet gateway, depending on how it was created.

The following diagram shows instances in each platform. Note the following:

- Instances C1, C2, C3, and C4 are in the EC2-Classic platform. C1 and C2 were launched by one account, and C3 and C4 were launched by a different account. These instances can communicate with each other, can access the Internet directly.
- Instances V1 and V2 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access instances in EC2-Classic and the Internet through the Internet gateway.



Sharing and Accessing Resources Between EC2-Classic and EC2-VPC

Some resources and features in your AWS account can be shared or accessed between the EC2-Classic and EC2-VPC platforms, for example, through ClassicLink. For more information about ClassicLink, see [ClassicLink](#) (p. 550).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC. For more information about migrating from EC2-Classic to a VPC, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC](#) (p. 560).

The following resources can be shared or accessed between EC2-Classic and a VPC.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Sharing and Accessing Resources
Between EC2-Classic and EC2-VPC**

Resource	Notes
AMI	
Bundle task	
EBS volume	
Elastic IP address	You can migrate an Elastic IP address from EC2-Classic to EC2-VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 583) .
Instance	<p>An EC2-Classic instance can communicate with instances in a VPC using public IP addresses, or you can use ClassicLink to enable communication over private IP addresses.</p> <p>You can't migrate an instance from EC2-Classic to a VPC. However, you can migrate your application from an instance in EC2-Classic to an instance in a VPC. For more information, see Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC (p. 560).</p>
Key pair	
Load balancer	<p>If you're using ClassicLink, you can register a linked EC2-Classic instance with a load balancer in a VPC, provided that the VPC has a subnet in the same Availability Zone as the instance.</p> <p>You can't migrate a load balancer from EC2-Classic to a VPC. You can't register an instance in a VPC with a load balancer in EC2-Classic.</p>
Placement group	
Reserved Instance	You can change the network platform for your Reserved Instances from EC2-Classic to EC2-VPC. For more information, see Modifying Your Reserved Instances (p. 210) .
Security group	<p>A linked EC2-Classic instance can use a VPC security groups through ClassicLink to control traffic to and from the VPC. VPC instances can't use EC2-Classic security groups.</p> <p>You can't migrate a security group from EC2-Classic to a VPC. You can copy rules from a security group in EC2-Classic to a security group in a VPC. For more information, see Creating a Security Group (p. 494).</p>
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot instances

Instance Types Available Only in a VPC

Instances of the following instance types are not supported in EC2-Classic and must be launched in a VPC:

- C4
- M4
- T2
- X1

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch a VPC-only instance:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or AWS SDK to launch a VPC-only instance. For more information, see [Create a Virtual Private Cloud \(VPC\)](#) (p. 23).
- Launch your VPC-only instance using the Amazon EC2 console. The Amazon EC2 console creates a nondefault VPC in your account and launches the instance into the subnet in the first Availability Zone. Note that the console creates the VPC with the following attributes:
 - One subnet in each Availability Zone, with the public IP addressing attribute set to `true` so that instances receive a public IP address. For more information, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.
 - An Internet gateway, and a main route table that routes traffic in the VPC to the Internet gateway. This enables the instances you launch in the VPC to communicate over the Internet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.
 - A default security group for the VPC and a default network ACL that is associated with each subnet. For more information, see [Security in Your VPC](#) in the *Amazon VPC User Guide*.

If you have other resources in EC2-Classic, you can take steps to migrate them to EC2-VPC. For more information, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC](#) (p. 560).

Amazon VPC Documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC Getting Started Guide	Provides a hands-on introduction to Amazon VPC.
Amazon VPC User Guide	Provides detailed information about how to use Amazon VPC.
Amazon VPC Network Administrator Guide	Helps network administrators configure your customer gateway.

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

For more information about the availability of either platform in your account, see [Availability](#) in the *Amazon VPC User Guide*. For more information about the differences between EC2-Classic and EC2-VPC, see [Differences Between EC2-Classic and EC2-VPC](#) (p. 544).

Supported Platforms in the Amazon EC2 Console

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**. If there are two values, `EC2` and `VPC`, you can launch instances into either platform. If there is one value, `VPC`, you can launch instances only into EC2-VPC.

If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier `vpc-1a2b3c4d`.


Supported Platforms

VPC

Default VPC

`vpc-1a2b3c4d`

If your account supports only EC2-VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list when you launch an instance using the launch wizard.

Network ⓘ	<input type="text" value="vpc-1a2b3c4d (172.31.0.0/16) (default)"/>	 Create new VPC
Subnet ⓘ	<input type="text" value="No preference (default subnet in any Availability Zor"/>	Create new subnet

EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

Supported Platforms

EC2

VPC

If your account supports EC2-Classic and EC2-VPC, you can launch into EC2-Classic using the launch wizard by selecting **Launch into EC2-Classic** from the **Network** list. To launch into a VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list.

Related Topic

For more information about how you can tell which platforms you can launch instances into, see [Detecting Your Supported Platforms](#) in the *Amazon VPC User Guide*.

ClassicLink

ClassicLink allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. ClassicLink removes the need to make use of public IP addresses or Elastic IP addresses to enable communication between instances in these platforms. For more information about private and public IP addresses, see [IP Addressing in Your VPC](#).

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. To find out which platform your account supports, see [Supported Platforms](#) (p. 549). For more information about the benefits of using a VPC, see [Amazon EC2 and Amazon Virtual Private Cloud](#) (p. 543). For more information about migrating your resources to a VPC, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC](#) (p. 560).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance hour usage apply.

Topics

- [ClassicLink Basics](#) (p. 550)
- [ClassicLink Limitations](#) (p. 553)
- [Working with ClassicLink](#) (p. 553)
- [API and CLI Overview](#) (p. 557)
- [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application](#) (p. 558)

ClassicLink Basics

There are two steps to linking an EC2-Classic instance to a VPC using ClassicLink. First, you must enable the VPC for ClassicLink. By default, all VPCs in your account are not enabled for ClassicLink, to maintain their isolation. After you've enabled the VPC for ClassicLink, you can then link any running EC2-Classic instance in the same region in your account to that VPC. Linking your instance includes selecting security groups from the VPC to associate with your EC2-Classic instance. After you've linked the instance, it can communicate with instances in your VPC using their private IP addresses, provided the VPC security groups allow it. Your EC2-Classic instance does not lose its private IP address when linked to the VPC.

Note

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

A linked EC2-Classic instance can communicate with instances in a VPC, but it does not form part of the VPC. If you list your instances and filter by VPC, for example, through the `DescribeInstances` API request, or by using the **Instances** screen in the Amazon EC2 console, the results do not return any EC2-Classic instances that are linked to the VPC. For more information about viewing your linked EC2-Classic instances, see [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances](#) (p. 555).

By default, if you use a public DNS hostname to address an instance in a VPC from a linked EC2-Classic instance, the hostname resolves to the instance's public IP address. The same occurs if you

use a public DNS hostname to address a linked EC2-Classic instance from an instance in the VPC. If you want the public DNS hostname to resolve to the private IP address, you can enable ClassicLink DNS support for the VPC. For more information, see [Enabling ClassicLink DNS Support \(p. 555\)](#).

If you no longer require a ClassicLink connection between your instance and the VPC, you can unlink the EC2-Classic instance from the VPC. This disassociates the VPC security groups from the EC2-Classic instance. A linked EC2-Classic instance is automatically unlinked from a VPC when it's stopped. After you've unlinked all linked EC2-Classic instances from the VPC, you can disable ClassicLink for the VPC.

Using Other AWS Services in Your VPC With ClassicLink

Linked EC2-Classic instances can access the following AWS services in the VPC: Amazon Redshift, Amazon ElastiCache, Elastic Load Balancing, and Amazon RDS. However, instances in the VPC cannot access the AWS services provisioned by the EC2-Classic platform using ClassicLink.

If you use Elastic Load Balancing in your VPC, you can register your linked EC2-Classic instance with the load balancer, provided that the instance is in an Availability Zone in which your VPC has a subnet. If you terminate the linked EC2-Classic instance, the load balancer deregisters the instance. For more information about working with load balancers in a VPC, see [Elastic Load Balancing in Amazon VPC in the *Elastic Load Balancing User Guide*](#).

If you use Auto Scaling, you can create an Auto Scaling group with instances that are automatically linked to a specified ClassicLink-enabled VPC at launch. For more information, see [Linking EC2-Classic Instances to a VPC](#) in the *Auto Scaling User Guide*.

If you use Amazon RDS instances or Amazon Redshift clusters in your VPC, and they are publicly accessible (accessible from the Internet), the endpoint you use to address those resources from a linked EC2-Classic instance by default resolves to a public IP address. If those resources are not publicly accessible, the endpoint resolves to a private IP address. To address a publicly accessible RDS instance or Redshift cluster over private IP using ClassicLink, you must use their private IP address or private DNS hostname, or you must enable ClassicLink DNS support for the VPC.

If you use a private DNS hostname or a private IP address to address an RDS instance, the linked EC2-Classic instance cannot use the failover support available for Multi-AZ deployments.

You can use the Amazon EC2 console to find the private IP addresses of your Amazon Redshift, Amazon ElastiCache, or Amazon RDS resources.

To locate the private IP addresses of AWS resources in your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Check the descriptions of the network interfaces in the **Description** column. A network interface that's used by Amazon Redshift, Amazon ElastiCache, or Amazon RDS will have the name of the service in the description. For example, a network interface that's attached to an Amazon RDS instance will have the following description: `RDSNetworkInterface`.
4. Select the required network interface.
5. In the details pane, get the private IP address from the **Primary private IP** field.

Controlling the Use of ClassicLink

By default, IAM users do not have permission to work with ClassicLink. You can create an IAM policy that grants users permissions to enable or disable a VPC for ClassicLink, link or unlink an instance to a ClassicLink-enabled VPC, and to view ClassicLink-enabled VPCs and linked EC2-Classic instances. For more information about IAM policies for Amazon EC2, see [IAM Policies for Amazon EC2 \(p. 500\)](#).

For more information about policies for working with ClassicLink, see the following example: [5. Working with ClassicLink \(p. 527\)](#).

Security Groups in ClassicLink

Linking your EC2-Classic instance to a VPC does not affect your EC2-Classic security groups. They continue to control all traffic to and from the instance. This excludes traffic to and from instances in the VPC, which is controlled by the VPC security groups that you associated with the EC2-Classic instance. EC2-Classic instances that are linked to the same VPC cannot communicate with each other through the VPC; regardless of whether they are associated with the same VPC security group. Communication between EC2-Classic instances is controlled by the EC2-Classic security groups associated with those instances. For an example of a security group configuration, see [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application \(p. 558\)](#).

After you've linked your instance to a VPC, you cannot change which VPC security groups are associated with the instance. To associate different security groups with your instance, you must first unlink the instance, and then link it to the VPC again, choosing the required security groups.

Routing for ClassicLink

When you enable a VPC for ClassicLink, a static route is added to all of the VPC route tables with a destination of `10.0.0.0/8` and a target of `local`. This allows communication between instances in the VPC and any EC2-Classic instances that are then linked to the VPC. If you add a custom route table to a ClassicLink-enabled VPC, a static route is automatically added with a destination of `10.0.0.0/8` and a target of `local`. When you disable ClassicLink for a VPC, this route is automatically deleted in all of the VPC route tables.

VPCs that are in the `10.0.0.0/16` and `10.1.0.0/16` IP address ranges can be enabled for ClassicLink only if they do not have any existing static routes in route tables in the `10.0.0.0/8` IP address range, excluding the local routes that were automatically added when the VPC was created. Similarly, if you've enabled a VPC for ClassicLink, you may not be able to add any more specific routes to your route tables within the `10.0.0.0/8` IP address range.

Important

If your VPC CIDR block is a publicly routable IP address range, consider the security implications before you link an EC2-Classic instance to your VPC. For example, if your linked EC2-Classic instance receives an incoming Denial of Service (DoS) request flood attack from a source IP address that falls within the VPC's IP address range, the response traffic is sent into your VPC. We strongly recommend that you create your VPC using a private IP address range as specified in [RFC 1918](#).

For more information about route tables and routing in your VPC, see [Route Tables](#) in the *Amazon VPC User Guide*.

Enabling a VPC Peering Connection for ClassicLink

If you have a VPC peering connection between two VPCs, and there are one or more EC2-Classic instances that are linked to one or both of the VPCs via ClassicLink, you can extend the VPC peering connection to enable communication between the EC2-Classic instances and the instances in the VPC on the other side of the VPC peering connection. This enables the EC2-Classic instances and the instances in the VPC to communicate using private IP addresses. To do this, you can enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, or you can enable a local linked EC2-Classic instance to communicate with instances in a peer VPC.

If you enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, a static route is automatically added to your route tables with a destination of `10.0.0.0/8` and a target of `local`.

For more information and examples, see [Configurations With ClassicLink](#) in the *Amazon VPC Peering Guide*.

ClassicLink Limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC, and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different region, or a different AWS account.
- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact AWS support to request that your dedicated tenancy VPC be allowed to be enabled for ClassicLink.

Important

EC2-Classic instances are run on shared hardware. If you've set the tenancy of your VPC to `dedicated` because of regulatory or security requirements, then linking an EC2-Classic instance to your VPC may not conform to those requirements, as you will be allowing a shared tenancy resource to address your isolated resources directly using private IP addresses. If you want to enable your dedicated VPC for ClassicLink, provide a detailed motivation in your request to AWS support.

- VPCs with routes that conflict with the EC2-Classic private IP address range of `10/8` cannot be enabled for ClassicLink. This does not include VPCs with `10.0.0.0/16` and `10.1.0.0/16` IP address ranges that already have local routes in their route tables. For more information, see [Routing for ClassicLink \(p. 552\)](#).
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- You can link a running Spot instance to a VPC. To indicate in a Spot instance request that the instance should be linked to a VPC when the request is fulfilled, you must use the launch wizard in the Amazon EC2 console.
- ClassicLink does not support transitive relationships out of the VPC. Your linked EC2-Classic instance will not have access to any VPN connection, VPC endpoint, or Internet gateway associated with the VPC. Similarly, resources on the other side of a VPN connection, or an Internet gateway will not have access to a linked EC2-Classic instance.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see [VPC Peering](#) in the *Amazon VPC User Guide*.
- If you link your EC2-Classic instance to a VPC in the `172.16.0.0/16` range, and you have a DNS server running on the `172.16.0.23/32` IP address within the VPC, then your linked EC2-Classic instance will not be able to access the VPC DNS server. To work around this issue, run your DNS server on a different IP address within the VPC.

Working with ClassicLink

You can use the Amazon EC2 and Amazon VPC consoles to work with the ClassicLink feature. You can enable or disable a VPC for ClassicLink, and link and unlink EC2-Classic instances to a VPC.

Note

The ClassicLink features are only visible in the consoles for accounts and regions that support EC2-Classic.

Topics

- [Enabling a VPC for ClassicLink \(p. 554\)](#)
- [Linking an Instance to a VPC \(p. 554\)](#)
- [Creating a VPC with ClassicLink Enabled \(p. 554\)](#)
- [Linking an EC2-Classic Instance to a VPC at Launch \(p. 555\)](#)

- [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances \(p. 555\)](#)
- [Enabling ClassicLink DNS Support \(p. 555\)](#)
- [Disabling ClassicLink DNS Support \(p. 556\)](#)
- [Unlinking a EC2-Classic Instance from a VPC \(p. 556\)](#)
- [Disabling ClassicLink for a VPC \(p. 556\)](#)

Enabling a VPC for ClassicLink

To link an EC2-Classic instance to a VPC, you must first enable the VPC for ClassicLink. You cannot enable a VPC for ClassicLink if the VPC has routing that conflicts with the EC2-Classic private IP address range. For more information, see [Routing for ClassicLink \(p. 552\)](#).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Choose a VPC, and then choose **Actions, Enable ClassicLink**.
4. In the confirmation dialog box, choose **Yes, Enable**.

Linking an Instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it.

Note

You can only link a running EC2-Classic instance to a VPC. You cannot link an instance that's in the `stopped` state.

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the running EC2-Classic instance, choose **Actions, ClassicLink, Link to VPC**. You can select more than one instance to link to the same VPC.
4. In the dialog box that displays, select a VPC from the list. Only VPCs that have been enabled for ClassicLink are displayed.
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

Creating a VPC with ClassicLink Enabled

You can create a new VPC and immediately enable it for ClassicLink by using the VPC wizard in the Amazon VPC console.

To create a VPC with ClassicLink enabled

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the Amazon VPC dashboard, choose **Start VPC Wizard**.
3. Select one of the VPC configuration options and choose **Select**.
4. On the next page of the wizard, choose **Yes for Enable ClassicLink**. Complete the rest of the steps in the wizard to create your VPC. For more information about using the VPC wizard, see [Scenarios for Amazon VPC](#) in the *Amazon VPC User Guide*.

Linking an EC2-Classic Instance to a VPC at Launch

You can use the launch wizard in the Amazon EC2 console to launch an EC2-Classic instance and immediately link it to a ClassicLink-enabled VPC.

To link an instance to a VPC at launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch Instance**.
3. Select an AMI, and then choose an instance type. On the **Configure Instance Details** page, ensure that you select **Launch into EC2-Classic** from the **Network** list.

Note

Some instance types, such as T2 instance types, can only be launched into a VPC. Ensure that you select an instance type that can be launched into EC2-Classic.

4. In the **Link to VPC (ClassicLink)** section, select a VPC from **Link to VPC**. Only ClassicLink-enabled VPCs are displayed. Select the security groups from the VPC to associate with the instance. Complete the other configuration options on the page, and then complete the rest of the steps in the wizard to launch your instance. For more information about using the launch wizard, see [Launching Your Instance from an AMI \(p. 279\)](#).

Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances

You can view all of your ClassicLink-enabled VPCs in the Amazon VPC console, and your linked EC2-Classic instances in the Amazon EC2 console.

To view your ClassicLink-enabled VPCs

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select a VPC, and in the **Summary** tab, look for the **ClassicLink** field. A value of **Enabled** indicates that the VPC is enabled for ClassicLink.
4. Alternatively, look for the **ClassicLink** column, and view the value that's displayed for each VPC (**Enabled** or **Disabled**). If the column is not visible, choose **Edit Table Columns** (the gear-shaped icon), select the **ClassicLink** attribute, and then choose **Close**.

To view your linked EC2-Classic instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an EC2-Classic instance, and in the **Description** tab, look for the **ClassicLink** field. If the instance is linked to a VPC, the field displays the ID of the VPC to which the instance is linked. If the instance is not linked to any VPC, the field displays **Unlinked**.
4. Alternatively, you can filter your instances to display only linked EC2-Classic instances for a specific VPC or security group. In the search bar, start typing `ClassicLink`, select the relevant ClassicLink resource attribute, and then select the security group ID or the VPC ID.

Enabling ClassicLink DNS Support

You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For this feature to work, your VPC must be enabled for DNS hostnames and DNS resolution.

Note

If you enable ClassicLink DNS support for your VPC, your linked EC2-Classic instance can access any private hosted zone associated with the VPC. For more information, see [Working with Private Hosted Zones](#) in the *Amazon Route 53 Developer Guide*.

To enable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **Yes** to enable ClassicLink DNS support, and choose **Save**.

Disabling ClassicLink DNS Support

You can disable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to public IP addresses and not private IP addresses.

To disable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **No** to disable ClassicLink DNS support, and choose **Save**.

Unlinking a EC2-Classic Instance from a VPC

If you no longer require a ClassicLink connection between your EC2-Classic instance and your VPC, you can unlink the instance from the VPC. Unlinking the instance disassociates the VPC security groups from the instance.

Note

A stopped instance is automatically unlinked from a VPC.

To unlink an instance from a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select your instance.
3. In the **Actions** list, select **ClassicLink, Unlink Instance**. You can select more than one instance to unlink from the same VPC.
4. Choose **Yes** in the confirmation dialog box.

Disabling ClassicLink for a VPC

If you no longer require a connection between EC2-Classic instances and your VPC, you can disable ClassicLink on the VPC. You must first unlink all linked EC2-Classic instances that are linked to the VPC.

To disable ClassicLink for a VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, then choose **Actions, Disable ClassicLink**.

4. In the confirmation dialog box, choose **Yes, Disable**.

API and CLI Overview

You can perform the tasks described on this page using the command line or the Query API. For more information about the command line interfaces and a list of available API actions, see [Accessing Amazon EC2](#) (p. 3).

Enable a VPC for ClassicLink

- [enable-vpc-classic-link](#) (AWS CLI)
- [Enable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLink](#) (Amazon EC2 Query API)

Link (attach) an EC2-Classic instance to a VPC

- [attach-classic-link-vpc](#) (AWS CLI)
- [Add-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [AttachClassicLinkVpc](#) (Amazon EC2 Query API)

Unlink (detach) an EC2-Classic instance from a VPC

- [detach-classic-link-vpc](#) (AWS CLI)
- [Dismount-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [DetachClassicLinkVpc](#) (Amazon EC2 Query API)

Disable ClassicLink for a VPC

- [disable-vpc-classic-link](#) (AWS CLI)
- [Disable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLink](#) (Amazon EC2 Query API)

Describe the ClassicLink status of VPCs

- [describe-vpc-classic-link](#) (AWS CLI)
- [Get-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLink](#) (Amazon EC2 Query API)

Describe linked EC2-Classic instances

- [describe-classic-link-instances](#) (AWS CLI)
- [Get-EC2ClassicLinkInstance](#) (AWS Tools for Windows PowerShell)
- [DescribeClassicLinkInstances](#) (Amazon EC2 Query API)

Enable a VPC peering connection for ClassicLink

- [modify-vpc-peering-connection-options](#) (AWS CLI)
- [Edit-EC2VpcPeeringConnectionOption](#) (AWS Tools for Windows PowerShell)
- [ModifyVpcPeeringConnectionOptions](#)(Amazon EC2 Query API)

Enable a VPC for ClassicLink DNS support

- [enable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Enable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Disable a VPC for ClassicLink DNS support

- [disable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Disable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Describe ClassicLink DNS support for VPCs

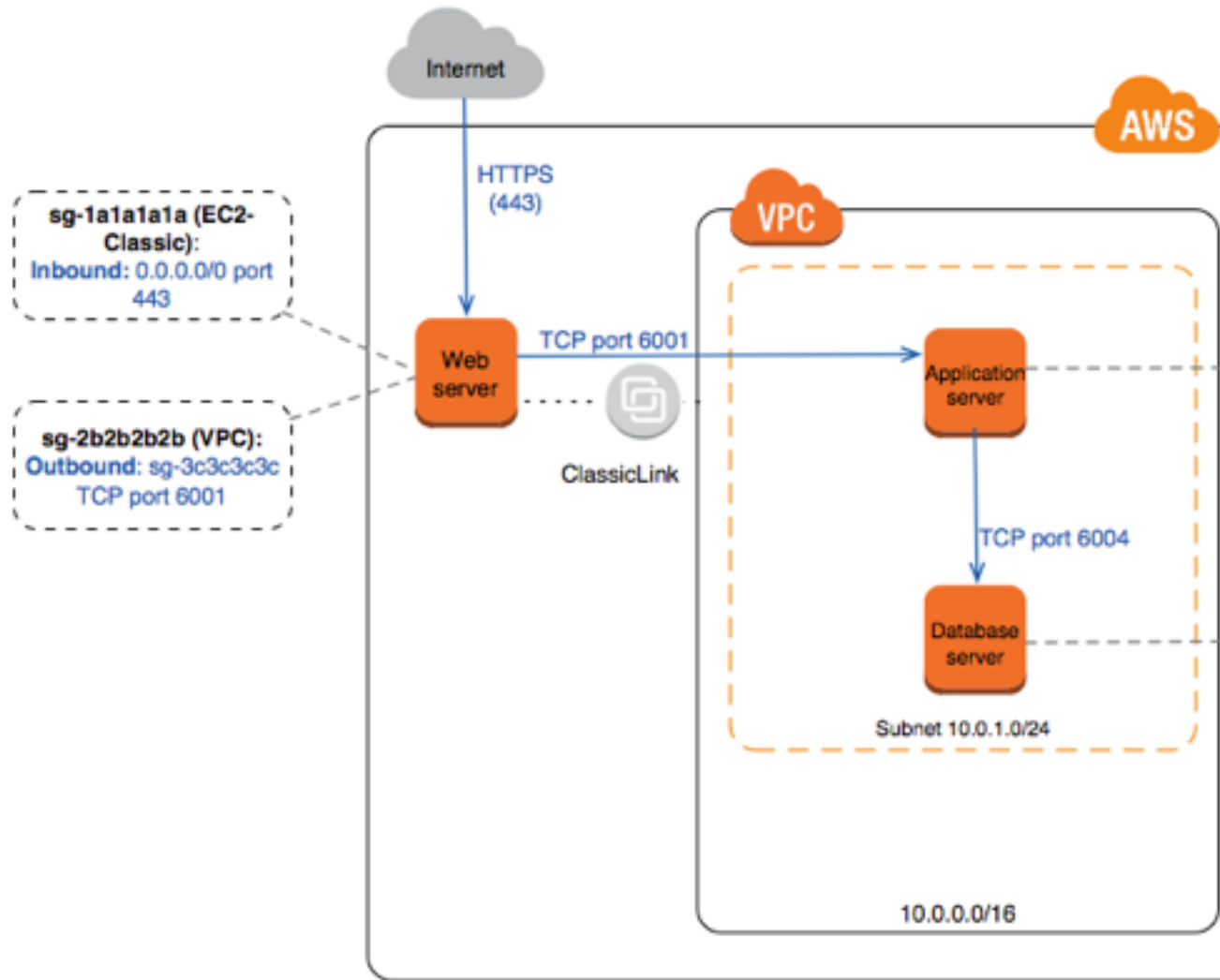
- [describe-vpc-classic-link-dns-support](#) (AWS CLI)
- [Get-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Example: ClassicLink Security Group Configuration for a Three-Tier Web Application

In this example, you have an application with three instances: a public-facing web server, an application server, and a database server. Your web server accepts HTTPS traffic from the Internet, and then communicates with your application server over TCP port 6001. Your application server then communicates with your database server over TCP port 6004. You're in the process of migrating your entire application to a VPC in your account. You've already migrated your application server and your database server to your VPC. Your web server is still in EC2-Classic and linked to your VPC via ClassicLink.

You want a security group configuration that allows traffic to flow only between these instances. You have four security groups: two for your web server (`sg-1a1a1a1a` and `sg-2b2b2b2b`), one for your application server (`sg-3c3c3c3c`), and one for your database server (`sg-4d4d4d4d`).

The following diagram displays the architecture of your instances, and their security group configuration.



Security Groups for Your Web Server (sg-1a1a1a1a and sg-2b2b2b2b)

You have one security group in EC2-Classic, and the other in your VPC. You associated the VPC security group with your web server instance when you linked the instance to your VPC via ClassicLink. The VPC security group enables you to control the outbound traffic from your web server to your application server.

The following are the security group rules for the EC2-Classic security group (sg-1a1a1a1a).

Inbound			
Source	Type	Port Range	Comments
0.0.0.0/0	HTTPS	443	Allows Internet traffic to reach your web server.

The following are the security group rules for the VPC security group (sg-2b2b2b2b).

Outbound			
Destination	Type	Port Range	Comments

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Migrating from EC2-Classic to a VPC**

sg-3c3c3c3c	TCP	6001	Allows outbound traffic from your web server to your application server in your VPC (or to any other instance associated with sg-3c3c3c3c).
-------------	-----	------	---------------------------------------------------------------------------------------------------------------------------------------------

Security Group for Your Application Server (sg-3c3c3c3c)

The following are the security group rules for the VPC security group that's associated with your application server.

Inbound			
Source	Type	Port Range	Comments
sg-2b2b2b2b	TCP	6001	Allows the specified type of traffic from your web server (or any other instance associated with sg-2b2b2b2b) to reach your application server.
Outbound			
Destination	Type	Port Range	Comments
sg-4d4d4d4d	TCP	6004	Allows outbound traffic from the application server to the database server (or to any other instance associated with sg-4d4d4d4d).

Security Group for Your Database Server (sg-4d4d4d4d)

The following are the security group rules for the VPC security group that's associated with your database server.

Inbound			
Source	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6004	Allows the specified type of traffic from your application server (or any other instance associated with sg-3c3c3c3c) to reach your database server.

Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC

Your AWS account might support both EC2-Classic and EC2-VPC, depending on when you created your account and which regions you've used. For more information, and to find out which platform your account supports, see [Supported Platforms \(p. 549\)](#). For more information about the benefits of using a VPC, and the differences between EC2-Classic and EC2-VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 543\)](#).

You create and use resources in your AWS account. Some resources and features, such as enhanced networking and T2 instances, can be used only in a VPC. Some resources can be shared between

EC2-Classic and a VPC, while some can't. For more information, see [Sharing and Accessing Resources Between EC2-Classic and EC2-VPC](#) (p. 546).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC.

There are two ways of migrating to a VPC. You can do a full migration, or you can do an incremental migration over time. The method you choose depends on the size and complexity of your application in EC2-Classic. For example, if your application consists of one or two instances running a static website, and you can afford a short period of downtime, you can do a full migration. If you have a multi-tier application with processes that cannot be interrupted, you can do an incremental migration using ClassicLink. This allows you to transfer functionality one component at a time until your application is running fully in your VPC.

If you need to migrate a Windows instance, see [Migrating a Windows Instance from EC2-Classic to a VPC](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [Full Migration to a VPC](#) (p. 561)
- [Incremental Migration to a VPC Using ClassicLink](#) (p. 567)

Full Migration to a VPC

Complete the following tasks to fully migrate your application from EC2-Classic to a VPC.

Tasks

- [Step 1: Create a VPC](#) (p. 561)
- [Step 2: Configure Your Security Group](#) (p. 562)
- [Step 3: Create an AMI from Your EC2-Classic Instance](#) (p. 562)
- [Step 4: Launch an Instance Into Your VPC](#) (p. 563)
- [Example: Migrating a Simple Web Application](#) (p. 565)

Step 1: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- Use a new, EC2-VPC-only AWS account. Your EC2-VPC-only account comes with a default VPC in each region, which is ready for you to use. Instances that you launch are by default launched into this VPC, unless you specify otherwise. For more information about your default VPC, see [Your Default VPC and Subnets](#). Use this option if you'd prefer not to set up a VPC yourself, or if you do not need specific requirements for your VPC configuration.
- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Note

T2 instance types must be launched into a VPC. If you do not have any VPCs in your EC2-Classic account, and you use the launch wizard in the Amazon EC2 console to launch a

T2 instance, the wizard creates a nondefault VPC for you. For more information about T2 instance types, see [T2 Instances \(p. 160\)](#). Your T2 instance will not be able to communicate with your EC2-Classic instances using private IP addresses. Consider migrating your existing instances to the same VPC using the methods outlined in this topic.

Step 2: Configure Your Security Group

You cannot use the same security groups between EC2-Classic and a VPC. However, if you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

Important

You can only copy security group rules to a new security group in the same AWS account in the same region. If you've created a new AWS account, you cannot use this method to copy your existing security group rules to your new account. You'll have to create a new security group, and add the rules yourself. For more information about creating a new security group, see [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then choose **Actions** and select **Copy to new**.
4. In the **Create Security Group** dialog box, specify a name and description for your new security group. Select your VPC from the **VPC** list.
5. The **Inbound** tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the **Outbound** tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see [Amazon EC2 Security Groups for Linux Instances \(p. 489\)](#).

Note

If you've defined a rule in your EC2-Classic security group that references another security group, you will not be able to use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Choose **Create**.

Step 3: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the [describe-instances](#) AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Important

Some instance types support both PV and HVM virtualization, while others support only one or the other. If you plan to use your AMI to launch a different instance type than your current instance type, check that the instance type supports the type of virtualization that your

AMI offers. If your AMI supports PV virtualization, and you want to use an instance type that supports HVM virtualization, you may have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI Virtualization Types \(p. 74\)](#).

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 89) .
Instance store	Create an instance store-backed AMI from your instance using the AMI tools. For more information, see Creating an Instance Store-Backed Linux AMI (p. 92) .
Instance store	Transfer your instance data to an EBS volume, then take a snapshot of the volume, and create an AMI from the snapshot. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 129) . Note This method converts an instance store-backed instance to an EBS-backed instance.

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- [Amazon EBS Volumes \(p. 636\)](#)
- [Creating an Amazon EBS Volume \(p. 650\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 685\)](#)
- [Creating an Amazon EBS Snapshot \(p. 686\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 652\)](#)

Step 4: Launch an Instance Into Your VPC

After you've created an AMI, you can launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

You can either launch your instance into a VPC that you've created in your existing account, or into a new, VPC-only AWS account.

Using Your Existing EC2-Classic Account

You can use the Amazon EC2 launch wizard to launch an instance into your VPC.

To launch an instance into your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 279\)](#).

Using Your New, VPC-Only Account

To launch an instance in your new AWS account, you'll first have to share the AMI you created with your new account. You can then use the Amazon EC2 launch wizard to launch an instance into your default VPC.

To share an AMI with your new AWS account

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to the account in which you created your AMI.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, ensure **Owned by me** is selected, then select your AMI.
5. In the **Permissions** tab, choose **Edit**. Enter the account number of your new AWS account, choose **Add Permission**, and then choose **Save**.

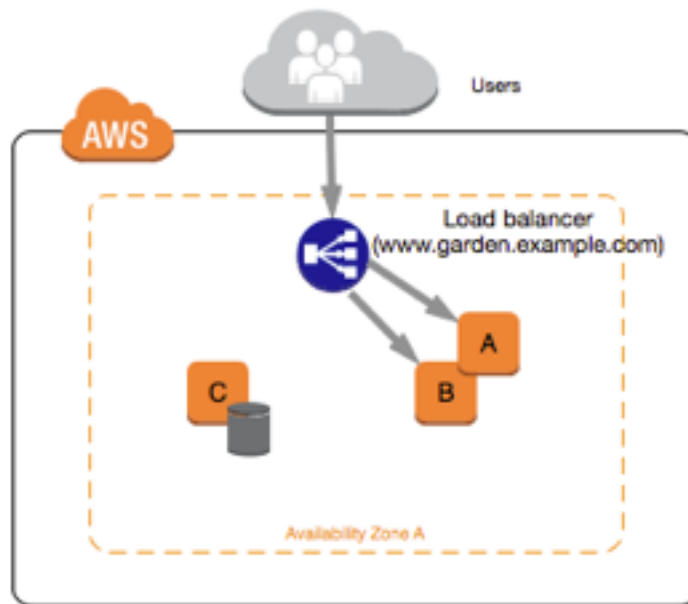
To launch an instance into your default VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to your new AWS account.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, select **Private images**. Select the AMI that you shared from your EC2-Classic account, then choose **Launch**.
5. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, your default VPC should be selected in the **Network** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
7. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
8. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

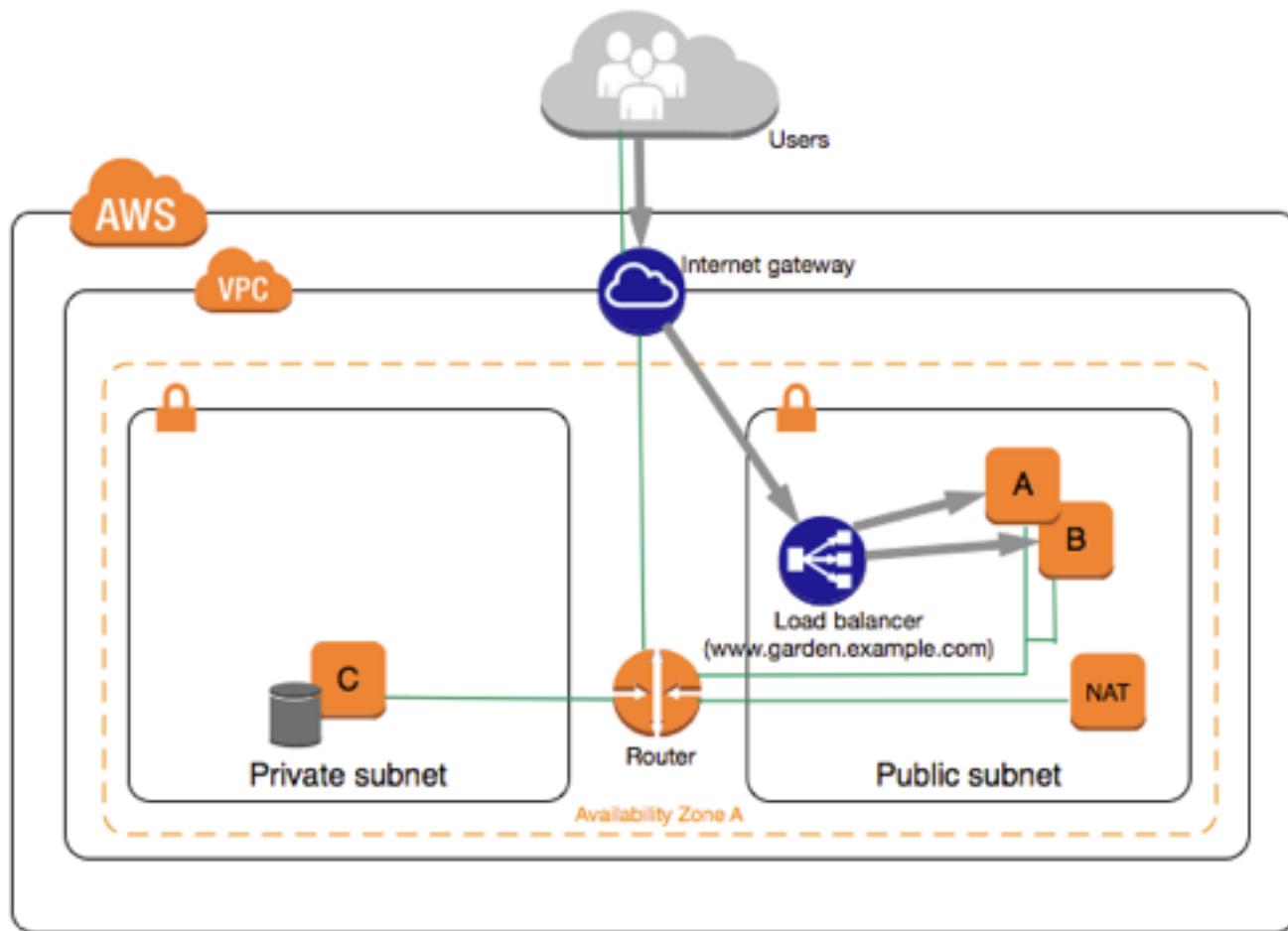
For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 279\)](#).

Example: Migrating a Simple Web Application

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use an Elastic Load Balancer to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name `www.garden.example.com`, and you've used Amazon Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.



The first part of migrating to a VPC is deciding what kind of VPC architecture will suit your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the Internet; however, you can enable Internet access through a Network Address Translation (NAT) device in the public subnet. You may want to set up a NAT device to support periodic updates and patches from the Internet for your database server. You'll migrate your Elastic IP addresses to EC2-VPC, and create an Elastic Load Balancer in your public subnet to load balance the traffic between your web servers.



To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC:** In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT device in your public subnet for you. For more information, see [Scenario 2: VPC with Public and Private Subnets](#) in the *Amazon VPC User Guide*.
- **Create AMIs from your instances:** Create an AMI from one of your web servers, and a second AMI from your database server. For more information, see [Step 3: Create an AMI from Your EC2-Classic Instance](#) (p. 562).
- **Configure your security groups:** In your EC2-Classic environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see [Step 2: Configure Your Security Group](#) (p. 562).

Tip

Create the security groups that are referenced by other security groups first.

- **Launch an instance into your new VPC:** Launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see [Step 4: Launch an Instance Into Your VPC](#) (p. 563).
- **Configure your NAT device:** If you are using a NAT instance, you must create security group for it that allows HTTP and HTTPS traffic from your private subnet. For more information, see [NAT Instances](#). If you are using a NAT gateway, traffic from your private subnet is automatically allowed.

- **Configure your database:** When you created an AMI from your database server in EC2-Classic, all the configuration information that was stored in that instance was copied to the AMI. You may have to connect to your new database server and update the configuration details; for example, if you configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you'll have to update the configuration files to grant the same permissions to your new VPC web servers instead.
- **Configure your web servers:** Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers' configuration settings to point to your new database instance.

Note

By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server may not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Migrate your Elastic IP addresses:** Disassociate your Elastic IP addresses from your web servers in EC2-Classic, and then migrate them to EC2-VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see [Migrating an Elastic IP Address from EC2-Classic to EC2-VPC \(p. 583\)](#).
- **Create a new load balancer:** To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways you can configure your load balancer in VPC. For more information, see [Elastic Load Balancing in Amazon VPC](#).
- **Update your DNS records:** After you've set up your load balancer in your public subnet, ensure that your `www.garden.example.com` domain points to your new load balancer. To do this, you'll need to update your DNS records and update your alias record set in Amazon Route 53. For more information about using Amazon Route 53, see [Getting Started with Amazon Route 53](#).
- **Shut down your EC2-Classic resources:** After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them. Terminate your EC2-Classic instances, and release your EC2-Classic Elastic IP addresses.

Incremental Migration to a VPC Using ClassicLink

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink allows you to link an EC2-Classic instance to a VPC in your account in the same region, allowing your new VPC resources to communicate with the EC2-Classic instance using private IP addresses. You can then migrate functionality to the VPC one step at a time. This topic provides some basic steps for managing an incremental migration from EC2-Classic to a VPC.

For more information about ClassicLink, see [ClassicLink \(p. 550\)](#).

Topics

- [Step 1: Prepare Your Migration Sequence \(p. 568\)](#)
- [Step 2: Create a VPC \(p. 568\)](#)
- [Step 3: Enable Your VPC for ClassicLink \(p. 568\)](#)
- [Step 4: Create an AMI from Your EC2-Classic Instance \(p. 568\)](#)
- [Step 5: Launch an Instance Into Your VPC \(p. 569\)](#)
- [Step 6: Link Your EC2-Classic Instances to Your VPC \(p. 570\)](#)
- [Step 7: Complete the VPC Migration \(p. 570\)](#)

Step 1: Prepare Your Migration Sequence

To use ClassicLink effectively, you must first identify the components of your application that must be migrated to the VPC, and then confirm the order in which to migrate that functionality.

For example, you have an application that relies on a presentation web server, a backend database server, and authentication logic for transactions. You may decide to start the migration process with the authentication logic, then the database server, and finally, the web server.

Step 2: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 3: Enable Your VPC for ClassicLink

After you've created a VPC, you can enable it for ClassicLink. For more information about ClassicLink, see [ClassicLink \(p. 550\)](#).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and then select **Enable ClassicLink** from the **Actions** list.
4. In the confirmation dialog box, choose **Yes, Enable**.

Step 4: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the [describe-instances](#) AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Important

Some instance types support both PV and HVM virtualization, while others support only one or the other. If you plan to use your AMI to launch a different instance type than your current instance type, check that the instance type supports the type of virtualization that your AMI offers. If your AMI supports PV virtualization, and you want to use an instance type that supports HVM virtualization, you may have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI Virtualization Types \(p. 74\)](#).

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 89) .
Instance store	Create an instance store-backed AMI from your instance using the AMI tools. For more information, see Creating an Instance Store-Backed Linux AMI (p. 92) .
Instance store	Transfer your instance data to an EBS volume, then take a snapshot of the volume, and create an AMI from the snapshot. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 129) . Note This method converts an instance store-backed instance to an EBS-backed instance.

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- [Amazon EBS Volumes \(p. 636\)](#)
- [Creating an Amazon EBS Volume \(p. 650\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 685\)](#)
- [Creating an Amazon EBS Snapshot \(p. 686\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 652\)](#)

Step 5: Launch an Instance Into Your VPC

The next step in the migration process is to launch instances into your VPC so that you can start transferring functionality to them. You can use the AMIs that you created in the previous step to launch instances into your VPC. The instances will have the same data and configurations as your existing EC2-Classic instances.

To launch an instance into your VPC using your custom AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.

5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 279\)](#).

After you've launched your instance and it's in the `running` state, you can connect to it and configure it as required.

Step 6: Link Your EC2-Classic Instances to Your VPC

After you've configured your instances and made the functionality of your application available in the VPC, you can use ClassicLink to enable private IP communication between your new VPC instances and your EC2-Classic instances.

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your EC2-Classic instance, then choose **Actions**, **ClassicLink**, and **Link to VPC**.

Note

Ensure that your instance is in the `running` state.

4. In the dialog box, select your ClassicLink-enabled VPC (only VPCs that are enabled for ClassicLink are displayed).
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

Step 7: Complete the VPC Migration

Depending on the size of your application and the functionality that must be migrated, repeat steps 4 to 6 until you've moved all the components of your application from EC2-Classic into your VPC.

After you've enabled internal communication between the EC2-Classic and VPC instances, you must update your application to point to your migrated service in your VPC, instead of your service in the EC2-Classic platform. The exact steps for this depend on your application's design. Generally, this includes updating your destination IP addresses to point to the IP addresses of your VPC instances instead of your EC2-Classic instances. You can migrate your Elastic IP addresses that you are currently using in the EC2-Classic platform to the EC2-VPC platform. For more information, see [Migrating an Elastic IP Address from EC2-Classic to EC2-VPC \(p. 583\)](#).

After you've completed this step and you've tested that the application is functioning from your VPC, you can terminate your EC2-Classic instances, and disable ClassicLink for your VPC. You can also clean up any EC2-Classic resources that you may no longer need to avoid incurring charges for them; for example, you can release Elastic IP addresses, and delete the volumes that were associated with your EC2-Classic instances.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC).

For information about the EC2-Classic and EC2-VPC platforms, see [Supported Platforms \(p. 549\)](#). For information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

Contents

- [Private IP Addresses and Internal DNS Hostnames \(p. 571\)](#)
- [Public IP Addresses and External DNS Hostnames \(p. 572\)](#)
- [Elastic IP Addresses \(p. 573\)](#)
- [Amazon DNS Server \(p. 573\)](#)
- [IP Address Differences Between EC2-Classic and EC2-VPC \(p. 573\)](#)
- [Determining Your Public, Private, and Elastic IP Addresses \(p. 574\)](#)
- [Assigning a Public IP Address \(p. 575\)](#)
- [Multiple Private IP Addresses \(p. 576\)](#)

Private IP Addresses and Internal DNS Hostnames

A private IP address is an IP address that's not reachable over the Internet. You can use private IP addresses for communication between instances in the same network (EC2-Classic or a VPC). For more information about the standards and specifications of private IP addresses, see [RFC 1918](#).

Note

You can create a VPC with a publicly-routable CIDR block that falls outside of the private IP address ranges specified in RFC 1918. However, for the purposes of this documentation, we refer to private IP addresses as the IP addresses that are within the CIDR range of your VPC.

When you launch an instance, we allocate a private IP address for the instance using DHCP. Each instance is also given an internal DNS hostname that resolves to the private IP address of the instance; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance launched in a VPC is given a primary private IP address in the address range of the subnet. For more information, see [Subnet Sizing](#) in the *Amazon VPC User Guide*. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's range for you. Each instance in a VPC has a default network interface (eth0) that is assigned the primary private IP address. You can also specify additional private IP addresses, known as *secondary private IP addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see [Multiple Private IP Addresses \(p. 576\)](#).

For instances launched in EC2-Classic, we release the private IP address when the instance is stopped or terminated. If you restart your stopped instance, it receives a new private IP address.

For instances launched in a VPC, a private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

- **Linux**

```
grep nameserver /etc/resolv.conf
```

Public IP Addresses and External DNS Hostnames

A public IP address is reachable from the Internet. You can use public IP addresses for communication between your instances and the Internet.

Each instance that receives a public IP address is also given an external DNS hostname; for example, `ec2-203-0-113-25.compute-1.amazonaws.com`. We resolve an external DNS hostname to the public IP address of the instance outside the network of the instance, and to the private IP address of the instance from within the network of the instance. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information about NAT, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

When you launch an instance in EC2-Classic, we automatically assign a public IP address to the instance from the EC2-Classic public IP address pool. You cannot modify this behavior. When you launch an instance into a VPC, your subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the EC2-VPC public IP address pool. By default, we assign a public IP address to instances launched in a default VPC, and we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance in a VPC receives a public IP address by doing the following:

- Modifying the public IP addressing attribute of your subnet. For more information, see [Modifying Your Subnet's Public IP Addressing Behavior](#) in the *Amazon VPC User Guide*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see [Assigning a Public IP Address \(p. 575\)](#).

A public IP address is assigned to your instance from Amazon's pool of public IP addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IP address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release the public IP address for your instance when it's stopped or terminated. Your stopped instance receives a new public IP address when it's restarted.
- We release the public IP address for your instance when you associate an Elastic IP address with your instance, or when you associate an Elastic IP address with the primary network interface (eth0) of your instance in a VPC. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead. For example, if you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see [Elastic IP Addresses \(p. 581\)](#).

If your instance is in a VPC and you assign it an Elastic IP address, it receives a DNS hostname if DNS hostnames are enabled. For more information, see [Using DNS with Your VPC](#) in the *Amazon VPC User Guide*.

Note

Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same region.

Elastic IP Addresses

An Elastic IP address is a public IP address that you can allocate to your account. You can associate it to and from instances as you require, and it's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see [Elastic IP Addresses](#) (p. 581).

Amazon DNS Server

Amazon provides a DNS server that resolves DNS hostnames to IP addresses. In EC2-Classic, the Amazon DNS server is located at 172.16.0.23. In EC2-VPC, the Amazon DNS server is located at the base of your VPC network range plus two. For more information, see [Amazon DNS Server](#) in the *Amazon VPC User Guide*.

IP Address Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between IP addresses for instances launched in EC2-Classic, instances launched in a default subnet, and instances launched in a nondefault subnet.

Characteristic	EC2-Classic	Default Subnet	Nondefault Subnet
Public IP address (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default subnet.	Your instance receives a static private IP address from the address range of your subnet.
Multiple IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Network interfaces	IP addresses are associated with the instance; network interfaces aren't supported.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.
Elastic IP address	An Elastic IP address is disassociated from your instance when you stop it.	An Elastic IP address remains associated with	An Elastic IP address remains associated with

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Determining Your Public,
Private, and Elastic IP Addresses**

Characteristic	EC2-Classic	Default Subnet	Nondefault Subnet
		your instance when you stop it.	your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default, except if you've created your VPC using the VPC wizard in the Amazon VPC console.

Determining Your Public, Private, and Elastic IP Addresses

You can use the Amazon EC2 console to determine the private IP addresses, public IP addresses, and Elastic IP addresses of your instances. You can also determine the public and private IP addresses of your instance from within your instance by using instance metadata. For more information, see [Instance Metadata and User Data \(p. 337\)](#).

To determine your instance's private IP addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the private IP address from the **Private IPs** field, and get the internal DNS hostname from the **Private DNS** field.
4. (VPC only) If you have one or more secondary private IP addresses assigned to network interfaces that are attached to your instance, get those IP addresses from the **Secondary private IPs** field.
5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select the a network interface that's associated with your instance.
6. Get the primary private IP address from the **Primary private IP** field, and the internal DNS hostname from the **Private DNS** field.
7. If you've assigned secondary private IP addresses to the network interface, get those IP addresses from the **Secondary private IPs** field.

To determine your instance's public IP addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the public IP address from the **Public IP** field, and get the external DNS hostname from the **Public DNS** field.
4. If an Elastic IP address has been associated with the instance, get the Elastic IP address from the **Elastic IP** field.

Note

If you've associated an Elastic IP address with your instance, the **Public IP** field also displays the Elastic IP address.

5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select a network interface that's associated with your instance.
6. Get the public IP address from the **Public IPs** field. An asterisk (*) indicates the public IP address or Elastic IP address that's mapped to the primary private IP address.

Note

The public IP address is displayed as a property of the network interface in the console, but it's mapped to the primary private IP address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IP address is not displayed. To determine your instance's public IP address from within the instance, you can use instance metadata.

To determine your instance's IP addresses using instance metadata

1. Connect to your instance.
2. Use the following command to access the private IP address:

- **Linux**

```
$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

- **Windows**

```
$ wget http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

- **Linux**

```
$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

- **Windows**

```
$ wget http://169.254.169.254/latest/meta-data/public-ipv4
```

Note that if an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assigning a Public IP Address

If you launch an instance in EC2-Classic, it is assigned a public IP address by default. You can't modify this behavior.

In a VPC, all subnets have an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IP addressing feature is also available for you to control whether your instance is assigned a public IP address; you can override the default behavior of the subnet's IP addressing attribute. The public IP address is assigned from Amazon's pool of public IP addresses, and is assigned to the network interface with the device index of `eth0`. This feature depends on certain conditions at the time you launch your instance.

Important

You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see [Public IP Addresses and External DNS Hostnames \(p. 572\)](#). If you require a persistent public IP address that you can associate or disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see [Elastic IP Addresses \(p. 581\)](#).

To access the public IP addressing feature when launching an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI and an instance type, and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list. An **Auto-assign Public IP** list is displayed. Choose **Enable** or **Disable** to override the default setting for the subnet.

Important

You cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.

5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see [Launching an Instance \(p. 279\)](#). On the final **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance.
6. On the **Instances** page, select your new instance and view its public IP address in **Public IP** field in the details pane.

The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see [Elastic IP Addresses \(p. 581\)](#). You can also modify your subnet's public IP addressing behavior. For more information, see [Modifying Your Subnet's Public IP Addressing Behavior](#).

API and Command Line Tools for Public IP Addressing

To enable or disable the public IP addressing feature, use one of the methods in the table below. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

Method	Parameter
AWS CLI	Use the <code>--associate-public-ip-address</code> or the <code>--no-associate-public-ip-address</code> option with the <code>run-instances</code> command.
AWS Tools for Windows PowerShell	Use the <code>-AssociatePublicIp</code> parameter with the <code>New-EC2Instance</code> command.
Query API	Use the <code>NetworkInterface.n.AssociatePublicIpAddress</code> parameter with the <code>RunInstances</code> request.

Multiple Private IP Addresses

In EC2-VPC, you can specify multiple private IP addresses for your instances. The number of network interfaces and private IP addresses that you can specify for an instance depends on the instance type. For more information, see [Private IP Addresses Per Network Interface Per Instance Type \(p. 589\)](#).

It can be useful to assign multiple private IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple private IP addresses for each network interface.

- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary private IP address to the standby instance.

Contents

- [How Multiple IP Addresses Work \(p. 577\)](#)
- [Assigning a Secondary Private IP Address \(p. 577\)](#)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address \(p. 579\)](#)
- [Associating an Elastic IP Address with the Secondary Private IP Address \(p. 580\)](#)
- [Viewing Your Secondary Private IP Addresses \(p. 580\)](#)
- [Unassigning a Secondary Private IP Address \(p. 580\)](#)

How Multiple IP Addresses Work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IP address to any network interface. The network interface can be attached to or detached from the instance.
- You must choose a secondary private IP address that's in the CIDR block range of the subnet for the network interface.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- Secondary private IP addresses can be assigned and unassigned to elastic network interfaces attached to running or stopped instances.
- Secondary private IP addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- When assigning multiple secondary private IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the secondary private IP addresses can't be assigned.
- Primary private IP addresses, secondary private IP addresses, and any associated Elastic IP addresses remain with the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IP address of the primary network interface to another network interface.
- You can move any additional network interface from one instance to another.

The following list explains how multiple IP addresses work with Elastic IP addresses:

- Each private IP address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IP address is reassigned to another interface, the secondary private IP address retains its association with an Elastic IP address.
- When a secondary private IP address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IP address.

Assigning a Secondary Private IP Address

You can assign the secondary private IP address to the network interface for an instance as you launch the instance, or after the instance is running. This section includes the following procedures.

- [To assign a secondary private IP address when launching an instance in EC2-VPC \(p. 578\)](#)
- [To assign a secondary IP address during launch using the command line \(p. 578\)](#)
- [To assign a secondary private IP to an existing instance \(p. 579\)](#)
- [To assign a secondary private IP to an existing instance using the command line \(p. 579\)](#)

To assign a secondary private IP address when launching an instance in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - a. Choose **Add Device** to add another network interface. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [Private IP Addresses Per Network Interface Per Instance Type \(p. 589\)](#).

Important

When you add a second network interface, the system can no longer auto-assign a public IP address. You will not be able to connect to the instance unless you assign an Elastic IP address to the primary network interface (eth0). You can assign the Elastic IP address after you complete the Launch wizard. For more information, see [Working with Elastic IP Addresses \(p. 584\)](#).

- b. For each network interface, you can specify a primary private IP address, and one or more secondary private IP addresses. For this example, however, accept the IP address that we automatically assign.
 - c. Under **Secondary IP addresses**, choose **Add IP**, and then enter a private IP address in the subnet range, or accept the default, `Auto-assign`, to let us select an address.
- #### Important
- After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see [Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address \(p. 579\)](#).
6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
 7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
 8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
 9. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To assign a secondary IP address during launch using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- The `--secondary-private-ip-addresses` option with the `run-instances` command (AWS CLI)

- Define `-NetworkInterface` and specify the `PrivateIpAddresses` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

To assign a secondary private IP to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**, and then select the network interface attached to the instance.
3. Choose **Actions, Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, do the following:
 - a. Choose **Assign new IP**.
 - b. Enter a specific IP address that's within the subnet range for the instance, or leave the field blank and we select an IP address for you.
 - c. (Optional) Choose **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
 - d. Choose **Yes, Update, Close**.

Note that alternatively, you can assign a secondary private IP address to an instance. Choose **Instances** in the navigation pane, select the instance, and then choose **Actions, Networking, Manage Private IP Addresses**. You can configure the same information in the dialog as you did in the steps above.

To assign a secondary private IP to an existing instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [assign-private-ip-addresses](#) (AWS CLI)
- [Register-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address

After you assign a secondary private IP address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

- If you are using Amazon Linux, the `ec2-net-utils` package can take care of this step for you. It configures additional network interfaces that you attach while the instance is running, refreshes secondary IP addresses during DHCP lease renewal, and updates the related routing rules. You can immediately refresh the list of interfaces by using the command `sudo service network restart` and then view the up-to-date list using `ip addr li`. If you require manual control over your network configuration, you can remove the `ec2-net-utils` package. For more information, see [Configuring Your Network Interface Using ec2-net-utils \(p. 593\)](#).
- If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IP addresses. If the instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.

For information about configuring a Windows instance, see [Configuring a Secondary Private IP Address for Your Windows Instance in a VPC](#) in the *Amazon EC2 User Guide for Windows Instances*.

Associating an Elastic IP Address with the Secondary Private IP Address

To associate an Elastic IP address with a secondary private IP address in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Actions, Associate Address**.
4. In the **Associate Address** dialog box, select the network interface from the **Network Interface** list, and then select the secondary IP address from the **Private IP address** list.
5. Choose **Associate**.

To associate an Elastic IP address with a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Viewing Your Secondary Private IP Addresses

To view the private IP addresses assigned to a network interface in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface whose private IP addresses you want to view.
4. On the **Details** tab in the details pane, check the **Primary private IP** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the network interface.

To view the private IP addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance whose private IP addresses you want to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the instance through its network interface.

Unassigning a Secondary Private IP Address

If you no longer require a secondary private IP address, you can unassign it from the instance or the network interface. When a secondary private IP address is unassigned from an elastic network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IP address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select an instance, choose **Actions, Networking, Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, choose **Unassign**.
5. Choose **Yes, Update**, and then close the dialog box.

To unassign a secondary private IP address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions, Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, choose **Unassign**.
5. Choose **Yes, Update, Close**.

To unassign a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [unassign-private-ip-addresses](#) (AWS CLI)
- [Unregister-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Elastic IP Addresses

An *Elastic IP address* is a static IP address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IP address, which is reachable from the Internet. If your instance does not have a public IP address, you can associate an Elastic IP address with your instance to enable communication with the Internet; for example, to connect to your instance from your local computer.

Topics

- [Elastic IP Address Basics \(p. 581\)](#)
- [Elastic IP Address Differences for EC2-Classical and EC2-VPC \(p. 582\)](#)
- [Working with Elastic IP Addresses \(p. 584\)](#)
- [Using Reverse DNS for Email Applications \(p. 587\)](#)
- [Elastic IP Address Limit \(p. 587\)](#)

Elastic IP Address Basics

The following are the basic characteristics of an Elastic IP address:

- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IP address (if it had one) is released back into Amazon's pool of public IP addresses. You cannot reuse a public IP address. For more information, see [Public IP Addresses and External DNS Hostnames \(p. 572\)](#).
- You can disassociate an Elastic IP address from a resource, and reassociate it with a different resource.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Elastic IP Address Differences
for EC2-Classic and EC2-VPC**

- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
- To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance. For more information, see [Amazon EC2 Pricing](#).
- An Elastic IP address is for use in a specific region only.
- When you associate an Elastic IP address with an instance that previously had a public IP address, the public DNS hostname of the instance changes to match the Elastic IP address.
- We resolve a public DNS hostname to the public IP address or the Elastic IP address of the instance outside the network of the instance, and to the private IP address of the instance from within the network of the instance.

If your account supports EC2-Classic, the use and behavior of Elastic IP addresses for EC2-Classic and EC2-VPC may differ. For more information, see [Elastic IP Address Differences for EC2-Classic and EC2-VPC](#) (p. 582).

Elastic IP Address Differences for EC2-Classic and EC2-VPC

If your account supports EC2-Classic, there's one pool of Elastic IP addresses for use with the EC2-Classic platform and another for use with the EC2-VPC platform. You can't associate an Elastic IP address that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. However, you can migrate an Elastic IP address you've allocated for use in the EC2-Classic platform to the EC2-VPC platform. You cannot migrate an Elastic IP address to another region. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms](#) (p. 549).

When you associate an Elastic IP address with an instance in EC2-Classic, a default VPC, or an instance in a nondefault VPC in which you assigned a public IP to the eth0 network interface during launch, the instance's current public IP address is released back into the public IP address pool. If you disassociate an Elastic IP address from the instance, the instance is automatically assigned a new public IP address within a few minutes. However, if you have attached a second network interface to an instance in a VPC, the instance is not automatically assigned a new public IP address. For more information about public IP addresses, see [Public IP Addresses and External DNS Hostnames](#) (p. 572).

For information about using an Elastic IP address with an instance in a VPC, see [Elastic IP Addresses](#) in the *Amazon VPC User Guide*.

The following table lists the differences between Elastic IP addresses on EC2-Classic and EC2-VPC. For more information about the differences between private and public IP addresses, see [IP Address Differences Between EC2-Classic and EC2-VPC](#) (p. 573).

Characteristic	EC2-Classic	EC2-VPC
Allocating an Elastic IP address	When you allocate an Elastic IP address, it's for use in EC2-Classic; however, you can migrate an Elastic IP address to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 583).	When you allocate an Elastic IP address, it's for use only in a VPC.
Associating an Elastic IP address	You associate an Elastic IP address with an instance.	An Elastic IP address is a property of an elastic network interface (ENI). You

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Elastic IP Address Differences
for EC2-Classic and EC2-VPC**

Characteristic	EC2-Classic	EC2-VPC
		can associate an Elastic IP address with an instance by updating the ENI attached to the instance. For more information, see Elastic Network Interfaces (ENI) (p. 588).
Reassociating an Elastic IP address	If you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance.	If your account supports EC2-VPC only, and you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance. If you're using a VPC in an EC2-Classic account, and you try to associate an Elastic IP address that's already associated with another instance, it succeeds only if you allowed reassociation.
Stopping an instance	If you stop an instance, its Elastic IP address is disassociated, and you must reassociate the Elastic IP address when you restart the instance.	If you stop an instance, its Elastic IP address remains associated.
Assigning multiple IP addresses	Instances support only a single private IP address and a corresponding Elastic IP address.	Instances support multiple IP addresses, and each one can have a corresponding Elastic IP address. For more information, see Multiple Private IP Addresses (p. 576).

Migrating an Elastic IP Address from EC2-Classic to EC2-VPC

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform, within the same region. This can assist you to migrate your resources from EC2-Classic to a VPC; for example, you can launch new web servers in your VPC, and then use the same Elastic IP addresses that you used for your web servers in EC2-Classic for your new VPC web servers.

After you've migrated an Elastic IP address to EC2-VPC, you cannot use it in the EC2-Classic platform; however, if required, you can restore it to EC2-Classic. After you've restored an Elastic IP address to EC2-Classic, you cannot use it in EC2-VPC until you migrate it again. You can only migrate an Elastic IP address from EC2-Classic to EC2-VPC. You cannot migrate an Elastic IP address that was originally allocated for use in EC2-VPC to EC2-Classic.

To migrate an Elastic IP address, it must not be associated with an instance. For more information about disassociating an Elastic IP address from an instance, see [Disassociating an Elastic IP Address and Reassociating it with a Different Instance](#) (p. 585).

You can migrate as many EC2-Classic Elastic IP addresses as you can have in your account. However, when you migrate an Elastic IP address to EC2-VPC, it counts against your Elastic IP address limit for EC2-VPC. You cannot migrate an Elastic IP address if it will result in you exceeding your limit. Similarly, when you restore an Elastic IP address to EC2-Classic, it counts against your Elastic IP address limit for EC2-Classic. For more information, see [Elastic IP Address Limit](#) (p. 587).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

For more information, see [Moving an Elastic IP Address](#) (p. 586).

Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

Topics

- [Allocating an Elastic IP Address](#) (p. 584)
- [Describing Your Elastic IP Addresses](#) (p. 584)
- [Associating an Elastic IP Address with a Running Instance](#) (p. 585)
- [Disassociating an Elastic IP Address and Reassociating it with a Different Instance](#) (p. 585)
- [Moving an Elastic IP Address](#) (p. 586)
- [Releasing an Elastic IP Address](#) (p. 587)

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the Amazon EC2 console or the command line. If your account supports EC2-Classic, you can allocate an address for use in EC2-Classic or in EC2-VPC.

To allocate an Elastic IP address for use in EC2-VPC using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. (EC2-Classic accounts) In the **Allocate New Address** dialog box, select **VPC** from **EIP used in**, and then choose **Yes, Allocate**. Close the confirmation dialog box.
5. (VPC-only accounts) Choose **Yes, Allocate**, and close the confirmation dialog box.

To allocate an Elastic IP address for use in EC2-Classic using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. Select **EC2**, and then choose **Yes, Allocate**. Close the confirmation dialog box.

To allocate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- `allocate-address` (AWS CLI)
- `New-EC2Address` (AWS Tools for Windows PowerShell)

Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the Amazon EC2 or the command line.

To describe your Elastic IP addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.

3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

To describe your Elastic IP addresses using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-addresses](#) (AWS CLI)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Running Instance

You can associate an Elastic IP address to an instance using the Amazon EC2 console or the command line.

(VPC only) If you're associating an Elastic IP address with your instance to enable communication with the Internet, you must also ensure that your instance is in a public subnet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

To associate an Elastic IP address with an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address, choose **Actions**, and then select **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from **Instance** and then choose **Associate**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address and Reassociating it with a Different Instance

You can disassociate an Elastic IP address and then reassociate it using the Amazon EC2 console or the command line.

To disassociate and reassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Disassociate Address**.
4. Choose **Yes, Disassociate** when prompted for confirmation.
5. Select the address that you disassociated in the previous step. For **Actions**, choose **Associate Address**.
6. In the **Associate Address** dialog box, select the new instance from **Instance**, and then choose **Associate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Moving an Elastic IP Address

Currently, you can migrate an Elastic IP address to EC2-VPC or restore it to EC2-Classical using the Amazon EC2 Query API, an AWS SDK, or the AWS CLI only.

After you've performed the command to move or restore your Elastic IP address, the process of migrating the Elastic IP address can take a few minutes. Use the [describe-moving-addresses](#) command to check whether your Elastic IP address is still moving, or has completed moving.

If the Elastic IP address is in a moving state for longer than 5 minutes, contact <http://aws.amazon.com/premiumsupport/>.

To move an Elastic IP address using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [move-address-to-vpc](#) (AWS CLI)
- [MoveAddressToVpc](#) (Amazon EC2 Query API)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classical using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [restore-address-to-classic](#) (AWS CLI)
- [RestoreAddressToClassic](#) (Amazon EC2 Query API)
- [Restore-EC2AddressToClassic](#) (AWS Tools for Windows PowerShell)

To describe the status of your moving addresses using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-moving-addresses](#) (AWS CLI)
- [DescribeMovingAddresses](#) (Amazon EC2 Query API)

- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

To retrieve the allocation ID for your migrated Elastic IP address in EC2-VPC

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-addresses](#) (AWS CLI)
- [DescribeAddresses](#) (Amazon EC2 Query API)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Releasing an Elastic IP Address

If you no longer need an Elastic IP address, we recommend that you release it (the address must not be associated with an instance). You incur charges for any Elastic IP address that's allocated for use with EC2-Classic but not associated with an instance.

You can release an Elastic IP address using the Amazon EC2 console or the command line.

To release an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Release Addresses**. Choose **Yes, Release** when prompted.

To release an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [release-address](#) (AWS CLI)
- [Remove-EC2Address](#) (AWS Tools for Windows PowerShell)

Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us. AWS works with ISPs and Internet anti-spam organizations to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. Note that a corresponding forward DNS record (record type A) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

If a reverse DNS record is associated with an Elastic IP address, the Elastic IP address is locked to your account and cannot be released from your account until the record is removed.

To remove email sending limits, or to provide us with your Elastic IP addresses and reverse DNS records, go to the [Request to Remove Email Sending Limitations](#) page.

Elastic IP Address Limit

By default, all AWS accounts are limited to 5 Elastic IP addresses per region, because public (IPv4) Internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP

address primarily for the ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional Elastic IP addresses, please complete the [Amazon EC2 Elastic IP Address Request Form](#). We will ask you to describe your use case so that we can understand your need for additional addresses.

Elastic Network Interfaces (ENI)

An elastic network interface (ENI) is a virtual network interface that you can attach to an instance in a VPC. ENIs are available only for instances running in a VPC.

An ENI can include the following attributes:

- A primary private IP address.
- One or more secondary private IP addresses.
- One Elastic IP address per private IP address.
- One public IP address, which can be auto-assigned to the elastic network interface for eth0 when you launch an instance. For more information, see [Public IP Addresses for Network Interfaces](#) (p. 591).
- One or more security groups.
- A MAC address.
- A source/destination check flag.
- A description.

You can create an elastic network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of an elastic network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move an elastic network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in a VPC has a default elastic network interface (the primary network interface, eth0) that is assigned a private IP address from the IP address range of your VPC. You cannot detach a primary network interface from an instance. You can create and attach additional elastic network interfaces. The maximum number of elastic network interfaces that you can use varies by instance type. For more information, see [Private IP Addresses Per Network Interface Per Instance Type](#) (p. 589).

Attaching multiple elastic network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your VPC.
- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

Contents

- [Private IP Addresses Per Network Interface Per Instance Type](#) (p. 589)
- [Public IP Addresses for Network Interfaces](#) (p. 591)
- [Creating a Management Network](#) (p. 591)
- [Use Network and Security Appliances in Your VPC](#) (p. 592)
- [Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets](#) (p. 592)
- [Create a Low Budget High Availability Solution](#) (p. 592)
- [Monitoring IP Traffic on Your Network Interface](#) (p. 593)

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Private IP Addresses Per Network
Interface Per Instance Type**

- [Best Practices for Configuring Elastic Network Interfaces](#) (p. 593)
- [Configuring Your Network Interface Using ec2-net-utils](#) (p. 593)
- [Creating an Elastic Network Interface](#) (p. 594)
- [Deleting an Elastic Network Interface](#) (p. 595)
- [Viewing Details about an Elastic Network Interface](#) (p. 595)
- [Attaching an Elastic Network Interface When Launching an Instance](#) (p. 596)
- [Attaching an Elastic Network Interface to a Stopped or Running Instance](#) (p. 597)
- [Detaching an Elastic Network Interface from an Instance](#) (p. 598)
- [Changing the Security Group of an Elastic Network Interface](#) (p. 598)
- [Changing the Source/Destination Checking of an Elastic Network Interface](#) (p. 599)
- [Associating an Elastic IP Address with an Elastic Network Interface](#) (p. 599)
- [Disassociating an Elastic IP Address from an Elastic Network Interface](#) (p. 600)
- [Changing Termination Behavior for an Elastic Network Interface](#) (p. 600)
- [Adding or Editing a Description for an Elastic Network Interface](#) (p. 601)
- [Adding or Editing Tags for an Elastic Network Interface](#) (p. 601)

Private IP Addresses Per Network Interface Per Instance Type

The following table lists the maximum number of elastic network interfaces (ENI) per instance type, and the maximum number of private IP addresses per ENI. ENIs and multiple private IP addresses are only available for instances running in a VPC. For more information, see [Multiple Private IP Addresses](#) (p. 576).

Instance Type	Maximum Elastic Network Interfaces	IP Addresses per Interface
c1.medium	2	6
c1.xlarge	4	15
c3.large	3	10
c3.xlarge	4	15
c3.2xlarge	4	15
c3.4xlarge	8	30
c3.8xlarge	8	30
c4.large	3	10
c4.xlarge	4	15
c4.2xlarge	4	15
c4.4xlarge	8	30
c4.8xlarge	8	30
cc2.8xlarge	8	30
cg1.4xlarge	8	30

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Private IP Addresses Per Network
Interface Per Instance Type**

Instance Type	Maximum Elastic Network Interfaces	IP Addresses per Interface
cr1.8xlarge	8	30
d2.xlarge	4	15
d2.2xlarge	4	15
d2.4xlarge	8	30
d2.8xlarge	8	30
g2.2xlarge	4	15
g2.8xlarge	8	30
hi1.4xlarge	8	30
hs1.8xlarge	8	30
i2.xlarge	4	15
i2.2xlarge	4	15
i2.4xlarge	8	30
i2.8xlarge	8	30
m1.small	2	4
m1.medium	2	6
m1.large	3	10
m1.xlarge	4	15
m2.xlarge	4	15
m2.2xlarge	4	30
m2.4xlarge	8	30
m3.medium	2	6
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30
m4.large	2	10
m4.xlarge	4	15
m4.2xlarge	4	15
m4.4xlarge	8	30
m4.10xlarge	8	30
r3.large	3	10
r3.xlarge	4	15

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Public IP Addresses for Network Interfaces**

Instance Type	Maximum Elastic Network Interfaces	IP Addresses per Interface
r3.2xlarge	4	15
r3.4xlarge	8	30
r3.8xlarge	8	30
t1.micro	2	2
t2.nano	2	2
t2.micro	2	2
t2.small	2	4
t2.medium	3	6
t2.large	3	12
x1.32xlarge	8	30

Public IP Addresses for Network Interfaces

In a VPC, all subnets have a modifiable attribute that determines whether instances launched into that subnet are assigned a public IP address. For more information, see [IP Addressing Behavior for Your Subnet](#) in the *Amazon VPC User Guide*. The public IP address is assigned from Amazon's pool of public IP addresses, and is assigned to the network interface with the device index of eth0 (the primary network interface).

When you create a network interface, it inherits the public IP addressing attribute from the subnet. If you subsequently modify the public IP addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network interface for eth0, the public IP addressing attribute is determined by the network interface.

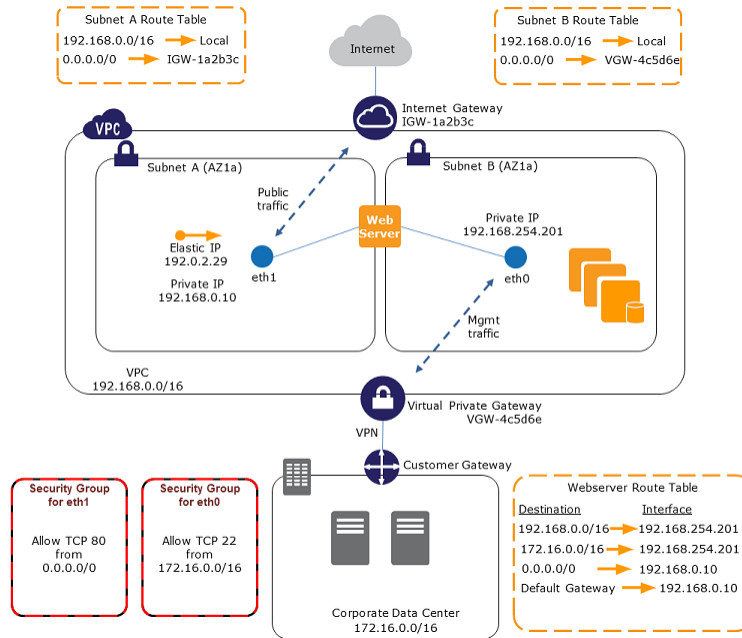
For more information, see [Public IP Addresses and External DNS Hostnames \(p. 572\)](#).

Creating a Management Network

You can create a management network using elastic network interfaces. In this scenario, the secondary elastic network interface on the instance handles public-facing traffic and the primary elastic network interface handles back-end management traffic and is connected to a separate subnet in your VPC that has more restrictive access controls. The public facing interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the Internet (for example, allow TCP port 80 and 443 from 0.0.0.0/0, or from the load balancer) while the private facing interface has an associated security group allowing SSH access only from an allowed range of IP addresses either within the VPC or from the Internet, a private subnet within the VPC or a virtual private gateway.

To ensure failover capabilities, consider using a secondary private IP for incoming traffic on an elastic network interface. In the event of an instance failure, you can move the interface and/or secondary private IP address to a standby instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Use Network and Security Appliances in Your VPC**



Use Network and Security Appliances in Your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple elastic network interfaces. You can create and attach secondary elastic network interfaces to instances in a VPC that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets

You can place an elastic network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a back-end network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the back end, and then sends requests to the servers on the back-end network.

Create a Low Budget High Availability Solution

If one of your instances serving a particular function fails, its elastic network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use an ENI as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the ENI to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic will begin flowing to the standby instance as soon as you attach the ENI to the replacement instance. Users will experience a brief loss of connectivity between the time the instance fails and the time that the ENI is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

Monitoring IP Traffic on Your Network Interface

You can enable a VPC flow log on your elastic network interface to capture information about the IP traffic going to and from the interface. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

Best Practices for Configuring Elastic Network Interfaces

- You can attach an elastic network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary (ethN) elastic network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach an elastic network interface in one subnet to an instance in another subnet in the same VPC; however, both the elastic network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the elastic network interfaces to attach to the instance for both the primary (eth0) and additional elastic network interfaces.
- Launching an Amazon Linux or Windows Server instance with multiple network interfaces automatically configures interfaces, private IP addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional elastic network interface may require you to manually bring up the second interface, configure the private IP address, and modify the route table accordingly. Instances running Amazon Linux or Windows Server automatically recognize the warm or hot attach and configure themselves.
- Attaching another elastic network interface to an instance is not a method to increase or double the network bandwidth to or from the dual-homed instance.
- If you attach two or more network interfaces from the same subnet to an instance, you may encounter networking issues such as asymmetric routing. If possible, use a secondary private IP address on the primary network interface instead. For more information, see [Assigning a Secondary Private IP Address](#) (p. 577).

Configuring Your Network Interface Using ec2-net-utils

Amazon Linux AMIs may contain additional scripts installed by AWS, known as ec2-net-utils. These scripts optionally automate the configuration of your elastic network interfaces (ENIs). These scripts are available for Amazon Linux only.

Use the following command to install the package on Amazon Linux if it's not already installed, or update it if it's installed and additional updates are available:

```
$ yum install ec2-net-utils
```

The following components are part of ec2-net-utils:

udev rules (/etc/udev/rules.d)

Identifies network interfaces when they are attached, detached, or reattached to a running instance, and ensures that the hotplug script runs (53-ec2-network-interfaces.rules).

Maps the MAC address to a device name (`75-persistent-net-generator.rules`, which generates `70-persistent-net.rules`).

hotplug script

Generates an interface configuration file suitable for use with DHCP (`/etc/sysconfig/network-scripts/ifcfg-ethN`). Also generates a route configuration file (`/etc/sysconfig/network-scripts/route-ethN`).

DHCP script

Whenever the elastic network interface receives a new DHCP lease, this script queries the instance metadata for Elastic IP addresses. For each Elastic IP address, it adds a rule to the routing policy database to ensure that outbound traffic from that address uses the correct network interface. It also adds each private IP address to the elastic network interface as a secondary address.

ec2ifup ethN

Extends the functionality of the standard **ifup**. After this script rewrites the configuration files `ifcfg-ethN` and `route-ethN`, it runs **ifup**.

ec2ifdown ethN

Extends the functionality of the standard **ifdown**. After this script removes any rules for the elastic network interface from the routing policy database, it runs **ifdown**.

ec2ifscan

Checks for network interfaces that have not been configured and configures them.

Note that this script isn't available in the initial release of `ec2-net-utils`.

To list any configuration files that were generated by `ec2-net-utils`, use the following command:

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

To disable the automation on a per-instance basis, you can add `EC2SYNC=no` to the corresponding `ifcfg-ethN` file. For example, use the following command to disable the automation for the `eth1` interface:

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

If you want to disable the automation completely, you can remove the package using the following command:

```
$ yum remove ec2-net-utils
```

Creating an Elastic Network Interface

You can create an elastic network interface using the Amazon EC2 console or the command line.

To create an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. In the **Create Network Interface** dialog box, provide the following information for the elastic network interface and choose **Yes, Create**.

- a. For **Description**, enter a descriptive name.
- b. For **Subnet**, select the subnet. Note that you can't move the elastic network interface to another subnet after it's created, and you can only attach the interface to instances in the same Availability Zone.
- c. For **Private IP**, enter the primary private IP address. If you don't specify an IP address, we'll select an available private IP address from within the selected subnet.
- d. For **Security groups**, select one or more security groups.

To create an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-network-interface](#) (AWS CLI)
- [New-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Deleting an Elastic Network Interface

You must first detach an elastic network interface from an instance before you can delete it. Deleting an elastic network interface releases all attributes associated with the interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You can delete an elastic network interface using the Amazon EC2 console or the command line.

To delete an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select an elastic network interface and choose **Delete**.
4. In the **Delete Network Interface** dialog box, choose **Yes, Delete**.

To delete an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Viewing Details about an Elastic Network Interface

You can describe an elastic network interface using the Amazon EC2 console or the command line.

To describe an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface.

4. View the details on the **Details** tab.

To describe an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

To describe an elastic network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-network-interface-attribute](#) (AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Attaching an Elastic Network Interface When Launching an Instance

You can specify an existing network interface or attach an additional network interface when you launch an instance. You can do this using the Amazon EC2 console or the command line.

Note

If an error occurs when attaching an elastic network interface to your instance, this causes the instance launch to fail.

To attach an elastic network interface when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI and instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC for **Network**, and a subnet for **Subnet**.
5. In the **Network Interfaces** section, the console enables you specify up to 2 elastic network interfaces (new, existing, or a combination) when you launch an instance. You can also enter a primary IP address and one or more secondary IP addresses for any new interface. When you've finished, choose **Next: Add Storage**.

You can add additional network interfaces to the instance after you launch it. The total number of network interfaces that you can attach varies by instance type. For more information, see [Private IP Addresses Per Network Interface Per Instance Type \(p. 589\)](#).

Note

You cannot auto-assign a public IP address to your instance if you specify more than one network interface.

6. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, you can select a security group or create a new one. Choose **Review and Launch**.

Note

If you specified an existing network interface in step 5, the instance is associated with the security group for that network interface, regardless of any option you select in this step.

9. On the **Review Instance Launch** page, details about the primary and additional network interface are displayed. Review the settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To attach an elastic network interface when launching an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Attaching an Elastic Network Interface to a Stopped or Running Instance

You can attach an elastic network interface to any of your stopped or running instances in your VPC using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

Note

If the public IP address on your instance is released, it will not receive a new one if there is more than one elastic network interface attached to the instance. For more information about the behavior of public IP addresses, see [Public IP Addresses and External DNS Hostnames \(p. 572\)](#).

To attach an elastic network interface to an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose **Actions, Networking, Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the elastic network interface and choose **Attach**.

To attach an elastic network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Attach**.
4. In the **Attach Network Interface** dialog box, select the instance and choose **Attach**.

To attach an elastic network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [attach-network-interface](#) (AWS CLI)
- [Add-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Detaching an Elastic Network Interface from an Instance

You can detach a secondary elastic network interface at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

To detach an elastic network interface from an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose **Actions, Networking, Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the elastic network interface and choose **Detach**.

To detach an elastic network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Detach**.
4. In the **Detach Network Interface** dialog box, choose **Yes, Detach**. If the elastic network interface fails to detach from the instance, choose **Force detachment**, and then try again.

To detach an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `detach-network-interface` (AWS CLI)
- `Dismount-EC2NetworkInterface` (AWS Tools for Windows PowerShell)

Changing the Security Group of an Elastic Network Interface

You can change the security groups that are associated with an elastic network interface. When you create the security group, be sure to specify the same VPC as the subnet for the interface.

You can change the security group for your elastic network interfaces using the Amazon EC2 console or the command line.

Note

To change security group membership for interfaces owned by other services, such as Elastic Load Balancing, use the console or command line interface for that service.

To change the security group of an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Change Security Groups**.
4. In the **Change Security Groups** dialog box, select the security groups to use, and choose **Save**.

To change the security group of an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Changing the Source/Destination Checking of an Elastic Network Interface

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to `disabled`. The default value is `enabled`.

You can change source/destination checking using the Amazon EC2 console or the command line.

To change source/destination checking for an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Change Source/Dest Check**.
4. In the dialog box, choose **Enabled** (if enabling) or **Disabled** (if disabling), and **Save**.

To change source/destination checking for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with an Elastic Network Interface

If you have an Elastic IP address, you can associate it with one of the private IP addresses for the elastic network interface. You can associate one Elastic IP address with each private IP address.

You can associate an Elastic IP address using the Amazon EC2 console or the command line.

To associate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Associate Address**.
4. In the **Associate Elastic IP Address** dialog box, select the Elastic IP address from the **Address** list.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Disassociating an Elastic IP Address
from an Elastic Network Interface**

5. For **Associate to private IP address**, select the private IP address to associate with the Elastic IP address.
6. Choose **Allow reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then choose **Associate Address**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address from an Elastic Network Interface

If the elastic network interface has an Elastic IP address associated with it, you can disassociate the address, and then either associate it with another elastic network interface or release it back to the address pool. Note that this is the only way to associate an Elastic IP address with an instance in a different subnet or VPC using an elastic network interface, as elastic network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the Amazon EC2 console or the command line.

To disassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, choose **Yes, Disassociate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

Changing Termination Behavior for an Elastic Network Interface

You can set the termination behavior for an elastic network interface attached to an instance so that it is automatically deleted when you delete the instance to which it's attached.

Note

By default, elastic network interfaces that are automatically created and attached to instances using the console are set to terminate when the instance terminates. However, network interfaces created using the command line interface aren't set to terminate when the instance terminates.

You can change the terminating behavior for an elastic network interface using the Amazon EC2 console or the command line.

To change the termination behavior for an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Change Termination Behavior**.
4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the elastic network interface to be deleted when you terminate an instance.

To change the termination behavior for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Adding or Editing a Description for an Elastic Network Interface

You can change the description for an elastic network interface using the Amazon EC2 console or the command line.

To change the description for an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface and choose **Actions, Change Description**.
4. In the **Change Description** dialog box, enter a description for the elastic network interface, and then choose **Save**.

To change the description for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Adding or Editing Tags for an Elastic Network Interface

Tags are metadata that you can add to an elastic network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see [Tagging Your Amazon EC2 Resources \(p. 751\)](#).

You can tag a resource using the Amazon EC2 console or the command line.

To add or edit tags for an elastic network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface.
4. In the details pane, choose **Tags**, **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag** for each tag to create, and enter a key and optional value. When you're done, choose **Save**.

To add or edit tags for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

Placement Groups

A *placement group* is a logical grouping of instances within a single Availability Zone. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see [Enhanced Networking \(p. 608\)](#).

First, you create a placement group and then you launch multiple instances into the placement group. We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group, stop and restart the instances in the placement group, and then try the launch again.

Contents

- [Placement Group Limitations \(p. 602\)](#)
- [Launching Instances into a Placement Group \(p. 603\)](#)
- [Deleting a Placement Group \(p. 604\)](#)

Placement Group Limitations

Placement groups have the following limitations:

- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group must be unique within your AWS account.
- The following are the only instance types that you can use when you launch an instance into a placement group:

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Launching Instances into a Placement Group**

- General purpose: m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge
- Compute optimized: c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
- Memory optimized: cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | x1.32xlarge
- Storage optimized: d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | hi1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge
- GPU: cg1.4xlarge | g2.2xlarge | g2.8xlarge
- The maximum network throughput speed of traffic between two instances in a placement group is limited by the slower of the two instances. For applications with high-throughput requirements, choose an instance type with 10 Gbps or 20 Gbps network connectivity. For more information about instance type network performance, see the [Amazon EC2 Instance Types Matrix](#).
- Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.
- You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about VPC peering connections, see [VPC Peering](#) in the *Amazon VPC User Guide*.
- You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.
- Reserved Instances provide a capacity reservation for EC2 instances in an Availability Zone. The capacity reservation can be used by instances in a placement group that are assigned to the same Availability Zone. However, it is not possible to explicitly reserve capacity for a placement group.
- To ensure that network traffic remains within the placement group, members of the placement group must address each other via their private IP addresses. If members address each other using their public IP addresses, throughput drops to 5 Gbps or less.
- Network traffic to and from resources outside the placement group is limited to 5 Gbps.

Launching Instances into a Placement Group

We suggest that you create an AMI specifically for the instances that you'll launch into a placement group.

To launch instances into a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Create an AMI for your instances.
 - a. From the Amazon EC2 dashboard, choose **Launch Instance**. After you complete the wizard, choose **Launch**.
 - b. Connect to your instance. (For more information, see [Connect to Your Linux Instance](#) (p. 289).)
 - c. Install software and applications on the instance, copy data, or attach additional Amazon EBS volumes.
 - d. (Optional) If your instance type supports enhanced networking, ensure that this feature is enabled by following the procedures in [Enhanced Networking on Linux](#) (p. 608).
 - e. In the navigation pane, choose **Instances**, select your instance, choose **Actions, Image, Create Image**. Provide the information requested by the **Create Image** dialog box, and then choose **Create Image**.
 - f. (Optional) You can terminate this instance if you have no further use for it.

3. Create a placement group.
 - a. In the navigation pane, choose **Placement Groups**.
 - b. Choose **Create Placement Group**.
 - c. In the **Create Placement Group** dialog box, provide a name for the placement group that is unique in the AWS account you're using, and then choose **Create**.

When the status of the placement group is `available`, you can launch instances into the placement group.
4. Launch instances into your placement group.
 - a. In the navigation pane, choose **Instances**.
 - b. Choose **Launch Instance**. Complete the wizard as directed, taking care to do the following:
 - On the **Choose an Amazon Machine Image (AMI)** page, select the **My AMIs** tab, and then select the AMI that you created.
 - On the **Choose an Instance Type** page, select an instance type that can be launched into a placement group.
 - On the **Configure Instance Details** page, enter the total number of instances that you'll need in this placement group, as you might not be able to add instances to the placement group later on.
 - On the **Configure Instance Details** page, select the placement group that you created from **Placement group**. If you do not see the **Placement group** list on this page, verify that you have selected an instance type that can be launched into a placement group, as this option is not available otherwise.

To launch instances into a placement group using the command line

1. Create an AMI for your instances using one of the following commands:
 - `create-image` (AWS CLI)
 - `New-EC2Image` (AWS Tools for Windows PowerShell)
2. Create a placement group using one of the following commands:
 - `create-placement-group` (AWS CLI)
 - `New-EC2PlacementGroup` (AWS Tools for Windows PowerShell)
3. Launch instances into your placement group using one of the following options:
 - `--placement` with `run-instances` (AWS CLI)
 - `-PlacementGroup` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Deleting a Placement Group

You can delete a placement group if you need to replace it or no longer need a placement group. Before you can delete your placement group, you must terminate all instances that you launched into the placement group.

To delete a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select and terminate all instances in the placement group. (You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.)

4. In the navigation pane, choose **Placement Groups**.
5. Select the placement group, and then choose **Delete Placement Group**.
6. When prompted for confirmation, choose **Yes, Delete**.

To delete a placement group using the command line

You can use one of the following sets of commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [terminate-instances](#) and [delete-placement-group](#) (AWS CLI)
- [Stop-EC2Instance](#) and [Remove-EC2PlacementGroup](#)(AWS Tools for Windows PowerShell)

Network Maximum Transmission Unit (MTU) for Your EC2 Instance

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. Ethernet packets consist of the frame, or the actual data you are sending, and the network overhead information that surrounds it.

Ethernet frames can come in different formats, and the most common format is the standard Ethernet v2 frame format. It supports 1500 MTU, which is the largest Ethernet packet size supported over most of the Internet. The maximum supported MTU for an instance depends on its instance type. All Amazon EC2 instance types support 1500 MTU, and many current instance sizes support 9001 MTU, or jumbo frames.

Contents

- [Jumbo Frames \(9001 MTU\) \(p. 605\)](#)
- [Path MTU Discovery \(p. 606\)](#)
- [Check the Path MTU Between Two Hosts \(p. 606\)](#)
- [Check and Set the MTU on your Amazon EC2 Instance \(p. 607\)](#)
- [Troubleshooting \(p. 608\)](#)

Jumbo Frames (9001 MTU)

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, and thus increasing the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, outside of a given AWS region (EC2-Classic), a single VPC, or a VPC peering connection, you will experience a maximum path of 1500 MTU. VPN connections and traffic sent over an Internet gateway are limited to 1500 MTU. If packets are over 1500 bytes, they are fragmented, or they are dropped if the `Don't Fragment` flag is set in the IP header.

Jumbo frames should be used with caution for Internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see [Placement Groups \(p. 602\)](#).

The following instances support jumbo frames:

- Compute optimized: C3, C4, CC2
- General purpose: M3, M4, T2
- GPU: CG1, G2
- Memory optimized: CR1, R3
- Storage optimized: D2, H11, HS1, I2

Path MTU Discovery

Path MTU Discovery is used to determine the path MTU between two devices. The path MTU is the maximum packet size that's supported on the path between the originating host and the receiving host. If a host sends a packet that's larger than the MTU of the receiving host or that's larger than the MTU of a device along the path, the receiving host or device returns the following ICMP message: *Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)*. This instructs the original host to adjust the MTU until the packet can be transmitted.

By default, security groups do not allow any inbound ICMP traffic. To ensure that your instance can receive this message and the packet does not get dropped, you must add a **Custom ICMP Rule** with the **Destination Unreachable** protocol to the inbound security group rules for your instance. For more information, see the [Adding Rules to a Security Group \(p. 495\)](#) and [API and Command Overview \(p. 497\)](#) sections in the Amazon EC2 Security Groups topic.

Important

Modifying your instance's security group to allow path MTU discovery does not guarantee that jumbo frames will not be dropped by some routers. An Internet gateway in your VPC will forward packets up to 1500 bytes only. 1500 MTU packets are recommended for Internet traffic.

Check the Path MTU Between Two Hosts

You can check the path MTU between two hosts using the **tracpath** command, which is part of the **iputils** package that is available by default on many Linux distributions, including Amazon Linux.

To check path MTU with tracpath

- Use the following command to check the path MTU between your Amazon EC2 instance and another host. You can use a DNS name or an IP address as the destination; this example checks the path MTU between an EC2 instance and `amazon.com`.

```
[ec2-user ~]$ tracpath amazon.com
1?: [LOCALHOST] pmtu 9001
1: ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1) 0.187ms
pmtu 1500
1: no reply
2: no reply
3: no reply
4: 100.64.16.241 (100.64.16.241) 0.574ms
5: 72.21.222.221 (72.21.222.221) 84.447ms asymm
21
6: 205.251.229.97 (205.251.229.97) 79.970ms asymm
19
7: 72.21.222.194 (72.21.222.194) 96.546ms asymm
16
8: 72.21.222.239 (72.21.222.239) 79.244ms asymm
15
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Check and Set the MTU on your Amazon EC2 Instance**

```
9: 205.251.225.73 (205.251.225.73)          91.867ms asymm
16
...
31: no reply
    Too many hops: pmtu 1500
    Resume: pmtu 1500
```

In this example, the path MTU is 1500.

Check and Set the MTU on your Amazon EC2 Instance

Some AMIs are configured to use jumbo frames on instance that support them, and others are configured to use standard frame sizes. You may want to use jumbo frames for network traffic within your VPC or you may want to use standard frames for Internet traffic. Whatever your use case, we recommend verifying that your instance will behave the way you expect it to. You can use the procedures in this section to check your network interface's MTU setting and modify it if needed.

To check the MTU setting on a Linux instance

- If your instance uses a Linux operating system, you can review the MTU value with the `ip` command. Run the following command to determine the current MTU value:

```
[ec2-user ~]$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state
UP mode DEFAULT group default qlen 1000
    link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

In the above example, the `mtu 9001` in the output indicates that this instance uses jumbo frames.

To set the MTU value on a Linux instance

1. If your instance uses a Linux operating system, you can set the MTU value with the `ip` command. Run the following command to set the desired MTU value. This procedure sets the MTU to 1500, but it is the same for 9001.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
```

2. (Optional) To persist your network MTU setting after a reboot, modify the following configuration files, based on your operating system type. This procedure covers Amazon Linux and Ubuntu; for other distributions, consult their specific documentation.
 - For Amazon Linux, add the following lines to your `/etc/dhcp/dhclient-eth0.conf` file.

```
interface "eth0" {
supersede interface-mtu 1500;
}
```

- For Ubuntu, add the following line to `/etc/network/interfaces.d/eth0.cfg`.

```
post-up /sbin/ifconfig eth0 mtu 1500
```

3. (Optional) Reboot your instance and verify that the MTU setting is correct.

Troubleshooting

If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see [Queries Appear to Hang](#) in the *Amazon Redshift Cluster Management Guide*

Enhanced Networking on Linux

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on [supported instance types](#) (p. 608). SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

Contents

- [Enhanced Networking Types](#) (p. 608)
- [Enabling Enhanced Networking on Your Instance](#) (p. 608)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances in a VPC](#) (p. 609)
- [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Linux Instances in a VPC](#) (p. 617)
- [Troubleshooting the Elastic Network Adapter \(ENA\)](#) (p. 626)

Enhanced Networking Types

Depending on your instance type, enhanced networking can be enabled using one of the following mechanisms:

Intel 82599 Virtual Function (VF) interface

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types. For more information, see the [Instance Type Matrix](#).

C3, C4, R3, I2, M4 and D2 instances use the Intel 82599 VF interface for enhanced networking.

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 20 Gbps.

X1 instances use the Elastic Network Adapter for enhanced networking.

Enabling Enhanced Networking on Your Instance

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in [Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances in a VPC](#) (p. 609).

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Linux Instances in a VPC](#) (p. 617).

Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances in a VPC

Amazon EC2 provides enhanced networking capabilities to C3, C4, R3, I2, M4 and D2 instances with the Intel 82599 VF interface, which uses the Intel `ixgbevf` driver.

To prepare for enhanced networking with the Intel 82599 VF interface, set up your instance as follows:

- Launch the instance from an HVM AMI using Linux kernel version of 2.6.32 or later. The latest Amazon Linux HVM AMIs have the modules required for enhanced networking installed and have the required attributes set. Therefore, if you launch an Amazon EBS-backed, enhanced networking-supported instance using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance.
- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- Install and configure the [AWS CLI](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `sriovNetSupport` attribute, may render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Contents

- [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled \(p. 609\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Amazon Linux \(p. 611\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Ubuntu \(p. 613\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Other Linux Distributions \(p. 615\)](#)
- [Troubleshooting Connectivity Issues \(p. 617\)](#)

Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled

To test whether enhanced networking with the Intel 82599 VF interface is already enabled, verify that the `ixgbevf` module is installed on your instance and that the `sriovNetSupport` attribute is set. If your instance satisfies these two conditions, then the `ethtool -i ethz` command should show that the module is in use on the network interface.

Kernel Module (`ixgbevf`)

To verify that the `ixgbevf` module is installed and that the version is compatible with enhanced networking, use the `modinfo` command as follows:

```
[ec2-user ~]$ modinfo ixgbevf
filename:          /lib/modules/3.10.48-55.140.amzn1.x86_64/kernel/drivers/
amazon/ixgbevf/ixgbevf.ko
version:           2.14.2
license:           GPL
description:       Intel(R) 82599 Virtual Function Driver
author:            Intel Corporation, <linux.nics@intel.com>
```


**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Enabling Enhanced Networking: Intel 82599 VF**

```
srcversion:      50CBF6F36B99FE70E56C95A
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intree:         Y
vermagic:       3.10.48-55.140.amzn1.x86_64 SMP mod_unload modversions
parm:          InterruptThrottleRate:Maximum interrupts per second, per
              vector, (956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

In the above Amazon Linux case, the `ixgbevf` module is already installed and it is at the minimum recommended version (2.14.2).

```
ubuntu:~$ modinfo ixgbevf
filename:       /lib/modules/3.13.0-29-generic/kernel/drivers/net/ethernet/
intel/ixgbevf/ixgbevf.ko
version:        2.11.3-k
license:        GPL
description:    Intel(R) 82599 Virtual Function Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:     0816EA811025C8062A9C269
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intree:         Y
vermagic:       3.13.0-29-generic SMP mod_unload modversions
signer:         Magrathea: Glacier signing key
sig_key:        66:02:CB:36:F1:31:3B:EA:01:C4:BD:A9:65:67:CF:A7:23:C9:70:D8
sig_hashalgo:  sha512
parm:          debug:Debug level (0=none,...,16=all) (int)
```

In the above Ubuntu instance, the module is installed, but the version is 2.11.3-k, which does not have all of the latest bug fixes that the recommended version 2.14.2 does. In this case, the `ixgbevf` module would work, but a newer version can still be installed and loaded on the instance for the best experience.

Instance Attribute (`sriovNetSupport`)

To check whether an instance has the enhanced networking `sriovNetSupport` attribute set, use the following command:

- [describe-instance-attribute](#) (AWS CLI)

```
$ aws ec2 describe-instance-attribute --instance-id instance_id --attribute
sriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {
  "Value": "simple"
},
```

Image Attribute (`sriovNetSupport`)

To check whether an AMI already has the enhanced networking `sriovNetSupport` attribute set, use the following command:

- `describe-image-attribute` (AWS CLI)

```
$ aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

Note

This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

Network Interface Driver

Use the following command to verify that the module is being used on a particular interface, substituting the interface name that you wish to check. If you are using a single interface (default), it will be `eth0`.

```
[ec2-user ~]$ ethtool -i eth0  
driver: vif  
version:  
firmware-version:  
bus-info: vif-0  
supports-statistics: yes  
supports-test: no  
supports-EEPROM-access: no  
supports-register-dump: no  
supports-priv-flags: no
```

In the above case, the `ixgbevf` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0  
driver: ixgbevf  
version: 2.14.2  
firmware-version: N/A  
bus-info: 0000:00:03.0  
supports-statistics: yes  
supports-test: yes  
supports-EEPROM-access: no  
supports-register-dump: yes  
supports-priv-flags: no
```

In this case, the `ixgbevf` module is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

Enabling Enhanced Networking with the Intel 82599 VF Interface on Amazon Linux

The latest Amazon Linux HVM AMIs have the `ixgbevf` module required for enhanced networking installed and have the required `sriovNetSupport` attribute set. Therefore, if you launch a C3, C4, R3, I2, M4, or D2 instance using a current Amazon Linux HVM AMI, enhanced networking is already

enabled for your instance. For more information, see [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled](#) (p. 609).

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking (EBS-backed instances)

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ixgbevf`:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or the following command: [reboot-instances](#) (AWS CLI).
4. Connect to your instance again and verify that the `ixgbevf` module is installed and at the minimum recommended version using the `modinfo ixgbevf` command from [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled](#) (p. 609).
5. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking \(instance store-backed instances\)](#) (p. 612).

6. From your local computer, enable the enhanced networking attribute using the following command.

Warning

There is no way to disable the enhanced networking attribute after you've enabled it.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

7. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
8. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the `ixgbevf` module is installed and loaded on your network interface using the `ethtool -i ethz` command from [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled](#) (p. 609).

To enable enhanced networking (instance store-backed instances)

If your instance is an instance store-backed instance, follow [Step 1](#) (p. 612) through [Step 4](#) (p. 612) in the previous procedure, and then create a new AMI as described in [Creating an](#)

[Instance Store-Backed Linux AMI \(p. 92\)](#). Be sure to enable the enhanced networking attribute when you register the AMI.

- `register-image` (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support simple ...
```

Enabling Enhanced Networking with the Intel 82599 VF Interface on Ubuntu

The following procedure provides the general steps that you'll take when enabling enhanced networking with the Intel 82599 VF interface on an Ubuntu instance.

To enable enhanced networking on Ubuntu (EBS-backed instances)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

Important

If during the update process, you are prompted to install `grub`, use `/dev/xvda` to install `grub` onto, and then choose to keep the current version of `/boot/grub/menu.lst`.

3. Install the `dkms` package so that your `ixgbevf` module is rebuilt every time your kernel is updated.

```
ubuntu:~$ sudo apt-get install -y dkms
```

4. Download the source for version 2.16.4 of the `ixgbevf` module on your instance from Sourceforge at <http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>.

Note

Earlier versions of `ixgbevf`, including the minimum recommended version, 2.14.2, do not build properly on some versions of Ubuntu. The 2.16.4 version of `ixgbevf` should be used for Ubuntu instances.

```
ubuntu:~$ wget "sourceforge.net/projects/e1000/files/ixgbevf  
stable/2.16.4/ixgbevf-2.16.4.tar.gz"
```

5. Decompress and unarchive the `ixgbevf` package.

```
ubuntu:~$ tar -xzf ixgbevf-2.16.4.tar.gz
```

6. Move the `ixgbevf` package to the `/usr/src/` directory so `dkms` can find it and build it for each kernel update.

```
ubuntu:~$ sudo mv ixgbevf-2.16.4 /usr/src/
```

7. Create the `dkms` configuration file with the following values, substituting your version of `ixgbevf`.
 - a. Create the file.

```
ubuntu:~$ sudo touch /usr/src/ixgbevf-2.16.4/dkms.conf
```

- b. Edit the file and add the following values.

```
ubuntu:~$ sudo vim /usr/src/ixgbevf-2.16.4/dkms.conf
PACKAGE_NAME="ixgbevf"
PACKAGE_VERSION="2.16.4"
CLEAN="cd src/; make clean"
MAKE="cd src/; make BUILD_KERNEL=${kernelver}"
BUILT_MODULE_LOCATION[0]="src/"
BUILT_MODULE_NAME[0]="ixgbevf"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ixgbevf"
AUTOINSTALL="yes"
```

8. Add, build, and install the `ixgbevf` module on your instance with `dkms`.

- a. Add the module to `dkms`.

```
ubuntu:~$ sudo dkms add -m ixgbevf -v 2.16.4
```

- b. Build the module with `dkms`.

```
ubuntu:~$ sudo dkms build -m ixgbevf -v 2.16.4
```

- c. Install the module with `dkms`.

```
ubuntu:~$ sudo dkms install -m ixgbevf -v 2.16.4
```

9. Rebuild the `initramfs` so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

10. Verify that the `ixgbevf` module is installed and at the minimum recommended version using the `modinfo ixgbevf` command from [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled](#) (p. 609).

```
ubuntu:~$ modinfo ixgbevf
filename:      /lib/modules/3.13.0-74-generic/updates/dkms/ixgbevf.ko
version:      2.16.4
license:      GPL
description:  Intel(R) 10 Gigabit Virtual Function Network Driver
author:       Intel Corporation, <linux.nics@intel.com>
srcversion:   759A432E3151C8F9F6EA882
alias:        pci:v00008086d00001515sv*sd*bc*sc*i*
alias:        pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
vermagic:    3.13.0-74-generic SMP mod_unload modversions
parm:        InterruptThrottleRate:Maximum interrupts per second, per
            vector, (956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

11. From your local computer, stop the instance using the Amazon EC2 console or the following command: `stop-instances` (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking on Ubuntu \(instance store-backed instances\)](#) (p. 615).

- From your local computer, enable the enhanced networking `sriovNetSupport` attribute using the following command. Note that there is no way to disable this attribute after you've enabled it.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

- (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#). The AMI inherits the enhanced networking `sriovNetSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
- From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
- (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking on Ubuntu (instance store-backed instances)

If your instance is an instance store-backed instance, follow [Step 1 \(p. 613\)](#) through [Step 10 \(p. 614\)](#) in the previous procedure, and then create a new AMI as described in [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#). Be sure to enable the enhanced networking attribute when you register the AMI.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support simple ...
```

Enabling Enhanced Networking with the Intel 82599 VF Interface on Other Linux Distributions

The following procedure provides the general steps that you'll take when enabling enhanced networking with the Intel 82599 VF interface on a Linux distribution other than Amazon Linux or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the specific documentation for your Linux distribution.

To enable enhanced networking on Linux (EBS-backed instances)

- Connect to your instance.
- Download the source for version 2.14.2 of the `ixgbevf` module on your instance from Sourceforge at <http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>. This is the minimum version recommended for enhanced networking.

Note

Earlier versions of `ixgbevf`, including the minimum recommended version, 2.14.2, do not build properly on some Linux distributions, including certain versions of Ubuntu. If you

receive build errors, you may try a newer version, such as 2.16.4 (which fixes the build issue on affected Ubuntu versions).

3. Compile and install the `ixgbevf` module on your instance.

If your distribution supports **dkms**, then you should consider configuring **dkms** to recompile the `ixgbevf` module whenever your system's kernel is updated. If your distribution does not support **dkms** natively, you can find it in the EPEL repository (<https://fedoraproject.org/wiki/EPEL>) for Red Hat Enterprise Linux variants, or you can download the software at <http://linux.dell.com/dkms/>. Use [Step 6 \(p. 613\)](#) through [Step 8 \(p. 614\)](#) in [To enable enhanced networking on Ubuntu \(EBS-backed instances\) \(p. 613\)](#) for help configuring **dkms**.

Warning

If you compile the `ixgbevf` module for your current kernel and then upgrade your kernel without rebuilding the driver for the new kernel, your system may revert to the distribution-specific `ixgbevf` module at the next reboot, which could make your system unreachable if the distribution-specific version is incompatible with enhanced networking.

4. Run the **sudo depmod** command to update module dependencies.
5. Update the `initramfs` on your instance to ensure that the new module loads at boot time.
6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names/](#) on the freedesktop.org website.
 - a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command:

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+\+'|^udev-[0-9]\+\+'
systemd-208-11.e17_0.2.x86_64
```

In the above Red Hat 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

- b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

```
[ec2-user ~]$ sudo sed -i '/^GRUB_CMDLINE_LINUX/s/\ "$"/ net.ifnames
=0"/' /etc/default/grub
```

- c. Rebuild the grub configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking \(instance store-backed instances\) \(p. 617\)](#)

8. From your local computer, enable the enhanced networking attribute using the following command. Note that there is no way to disable the networking attribute after you've enabled it.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

9. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` may fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
11. (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking (instance store-backed instances)

If your instance is an instance store-backed instance, follow [Step 1 \(p. 615\)](#) through [Step 5 \(p. 616\)](#) in the previous procedure, and then create a new AMI as described in [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#). Be sure to enable the enhanced networking attribute when you register the AMI.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support simple ...
```

Troubleshooting Connectivity Issues

If you lose connectivity while enabling enhanced networking, the `ixgbevf` module might be incompatible with the kernel. Try installing the version of the `ixgbevf` module included with the distribution of Linux for your instance.

If you enable enhanced networking for a PV instance or AMI, this can make your instance unreachable.

Enabling Enhanced Networking with the Elastic Network Adapter (ENA) on Linux Instances in a VPC

To prepare for enhanced networking with the ENA network adapter, set up your instance as follows:

- Launch the instance from an HVM AMI using Linux kernel version of 3.2 or later. The latest Amazon Linux HVM AMIs have the modules required for enhanced networking installed and have the required attributes set. Therefore, if you launch an Amazon EBS-backed, enhanced networking-supported instance using a current Amazon Linux HVM AMI, ENA enhanced networking is already enabled for your instance.
- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- Install and configure the [AWS CLI](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `enaSupport` attribute, may render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Contents

- [Testing Whether Enhanced Networking with ENA Is Enabled \(p. 618\)](#)
- [Enabling Enhanced Networking with ENA on Amazon Linux \(p. 620\)](#)
- [Enabling Enhanced Networking with ENA on Ubuntu \(p. 621\)](#)
- [Enabling Enhanced Networking with ENA on Other Linux Distributions \(p. 624\)](#)
- [Troubleshooting \(p. 626\)](#)

Testing Whether Enhanced Networking with ENA Is Enabled

To test whether enhanced networking with ENA is already enabled, verify that the `ena` module is installed on your instance and that the `enaSupport` attribute is set. If your instance satisfies these two conditions, then the `ethtool -i ethz` command should show that the module is in use on the network interface.

Kernel Module (`ena`)

To verify that the `ena` module is installed, use the `modinfo` command as follows:

```
[ec2-user ~]$ modinfo ena
filename:          /lib/modules/4.4.11-23.53.amzn1.x86_64/kernel/drivers/amazon/
net/ena/ena.ko
version:          0.6.6
license:          GPL
description:      Elastic Network Adapter (ENA)
author:           Amazon.com, Inc. or its affiliates
srcversion:       3141E47566402C79D6B8284
alias:            pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
intree:          Y
vermagic:        4.4.11-23.53.amzn1.x86_64 SMP mod_unload modversions
parm:            debug:Debug level (0=none,...,16=all) (int)
parm:            push_mode:Descriptor / header push mode
                 (0=automatic,1=disable,3=enable)
                 0 - Automatically choose according to device capability (default)
                 1 - Don't push anything to device memory
                 3 - Push descriptors and header buffer to device memory (int)
```

Amazon Elastic Compute Cloud User Guide for Linux Instances Enabling Enhanced Networking: ENA

```
parm:          enable_wd:Enable keepalive watchdog
(0=disable,1=enable,default=1) (int)
parm:          enable_missing_tx_detection:Enable missing Tx completions.
(default=1) (int)
parm:          numa_node_override_array:Numa node override map
(array of int)
parm:          numa_node_override:Enable/Disable numa node override
(0=disable)
(int)
```

In the above Amazon Linux case, the `ena` module is installed.

```
ubuntu:~$ modinfo ena
ERROR: modinfo: could not find module ena
```

In the above Ubuntu instance, the module is not installed, so you must first install it. For more information, see [Enabling Enhanced Networking with ENA on Ubuntu \(p. 621\)](#).

Instance Attribute (`enaSupport`)

To check whether an instance already has the enhanced networking `enaSupport` attribute set, use the following command:

- [describe-instances](#) (AWS CLI)

```
$ aws ec2 describe-instances --instance-id instance_id --query
Reservations[.Instances[.EnaSupport
```

If the `enaSupport` attribute isn't set, the returned JSON is empty; otherwise, it is set as follows:

```
[
  true
]
```

Image Attribute (`enaSupport`)

To check whether an AMI already has the enhanced networking `enaSupport` attribute set, use the following command:

- [describe-image-attribute](#) (AWS CLI)

```
$ aws ec2 describe-image-attribute --image-id ami_id --attribute enaSupport
```

Note

This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the attribute isn't set, `EnaSupport` is empty; otherwise, it is set as follows:

```
{
  "EnaSupport": {
    "Value": true
  },
  "ImageId": "ami_id"
}
```

```
}
```

Network Interface Driver

Use the following command to verify that the `ena` module is being used on a particular interface, substituting the interface name that you wish to check. If you are using a single interface (default), it will be `eth0`.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In the above case, the `ena` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 0.6.6
firmware-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In this case, the `ena` module is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

Enabling Enhanced Networking with ENA on Amazon Linux

The latest Amazon Linux HVM AMIs have the module required for enhanced networking with ENA installed and have the required `enaSupport` attribute set. Therefore, if you launch an instance with the latest Amazon Linux HVM AMI on a supported instance type, enhanced networking with ENA is already enabled for your instance. For more information, see [Testing Whether Enhanced Networking with ENA Is Enabled \(p. 618\)](#).

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking with ENA (EBS-backed instances)

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ena`:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or the following command: [reboot-instances](#) (AWS CLI).
4. Connect to your instance again and verify that the `ena` module is installed and at the minimum recommended version using the `modinfo ena` command from [Testing Whether Enhanced Networking with ENA Is Enabled](#) (p. 618).
5. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking with ENA \(instance store-backed instances\)](#) (p. 621).

6. From your local computer, enable the enhanced networking attribute using the following command.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

7. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI](#) (p. 89). The AMI inherits the enhanced networking `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking with ENA enabled by default.
8. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the `ena` module is installed and loaded on your network interface using the `ethtool -i ethz` command from [Testing Whether Enhanced Networking with ENA Is Enabled](#) (p. 618).

Note

If you are unable to connect to your instance after enabling enhanced networking with ENA, see [Troubleshooting the Elastic Network Adapter \(ENA\)](#) (p. 626).

To enable enhanced networking with ENA (instance store-backed instances)

If your instance is an instance store-backed instance, follow [Step 1](#) (p. 620) through [Step 4](#) (p. 621) in the previous procedure, and then create a new AMI as described in [Creating an Instance Store-Backed Linux AMI](#). Be sure to enable the enhanced networking `enaSupport` attribute when you register the AMI.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --ena-support ...
```

Enabling Enhanced Networking with ENA on Ubuntu

The following procedure provides the general steps that you'll take when enabling enhanced networking with ENA on an Ubuntu instance.

To enable enhanced networking with ENA on Ubuntu (EBS-backed instances)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

Important

If during the update process you are prompted to install `grub`, use `/dev/xvda` to install `grub` onto, and then choose to keep the current version of `/boot/grub/menu.lst`.

3. Install the `build-essential` packages to compile the kernel module and the `dkms` package so that your `ena` module is rebuilt every time your kernel is updated.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

4. Clone the source code for the `ena` module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

5. Move the `amzn-drivers` package to the `/usr/src/` directory so `dkms` can find it and build it for each kernel update. Append the version number (you can find the current version number in the release notes) of the source code to the directory name. For example, version `1.0.0` is shown in the example below.

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

6. Create the `dkms` configuration file with the following values, substituting your version of `ena`.
 - a. Create the file.

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

- b. Edit the file and add the following values.

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

7. Add, build, and install the `ena` module on your instance with `dkms`.
 - a. Add the module to `dkms`.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

- b. Build the module with `dkms`.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Enabling Enhanced Networking: ENA**

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

- c. Install the module with **dkms**.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

8. Rebuild the `initramfs` so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

9. Verify that the `ena` module is installed using the `modinfo ena` command from [Testing Whether Enhanced Networking with ENA Is Enabled \(p. 618\)](#).

```
ubuntu:~$ modinfo ena
filename:          /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
version:          1.0.0
license:          GPL
description:      Elastic Network Adapter (ENA)
author:           Amazon.com, Inc. or its affiliates
srcversion:       9693C876C54CA64AE48F0CA
alias:            pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:            pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:         3.13.0-74-generic SMP mod_unload modversions
parm:             debug:Debug level (0=none,...,16=all) (int)
parm:             push_mode:Descriptor / header push mode
                  (0=automatic,1=disable,3=enable)
                  0 - Automatically choose according to device capability (default)
                  1 - Don't push anything to device memory
                  3 - Push descriptors and header buffer to device memory (int)
parm:             enable_wd:Enable keepalive watchdog
                  (0=disable,1=enable,default=1) (int)
parm:             enable_missing_tx_detection:Enable missing Tx completions.
                  (default=1) (int)
parm:             numa_node_override_array:Numa node override map
                  (array of int)
parm:             numa_node_override:Enable/Disable numa node override
                  (0=disable)
                  (int)
```

10. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking with ENA on Ubuntu \(instance store-backed instances\) \(p. 624\)](#).

11. From your local computer, enable the enhanced networking attribute using the following command.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

12. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#) . The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
13. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
14. (Optional) Connect to your instance and verify that the module is installed.

Note

If you are unable to connect to your instance after enabling enhanced networking with ENA, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 626\)](#).

To enable enhanced networking with ENA on Ubuntu (instance store-backed instances)

If your instance is an instance store-backed instance, follow [Step 1 \(p. 622\)](#) through [Step 9 \(p. 623\)](#) in the previous procedure, and then create a new AMI as described in [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#). Be sure to enable the enhanced networking `enaSupport` attribute when you register the AMI.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --ena-support ...
```

Enabling Enhanced Networking with ENA on Other Linux Distributions

The following procedure provides the general steps that you'll take when enabling enhanced networking with ENA on a Linux distribution other than Amazon Linux or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the specific documentation for your Linux distribution.

To enable enhanced networking with ENA on Linux (EBS-backed instances)

1. Connect to your instance.
2. Clone the source code for the `ena` module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

3. Compile and install the `ena` module on your instance.

If your distribution supports **dkms**, then you should consider configuring **dkms** to recompile the `ena` module whenever your system's kernel is updated. If your distribution does not support

dkms natively, you can find it in the EPEL repository (<https://fedoraproject.org/wiki/EPEL>) for Red Hat Enterprise Linux variants, or you can download the software at <http://linux.dell.com/dkms/>. Use [Step 5 \(p. 622\)](#) through [Step 7 \(p. 622\)](#) in [To enable enhanced networking with ENA on Ubuntu \(EBS-backed instances\) \(p. 622\)](#) for help configuring **dkms**.

4. Run the **sudo depmod** command to update module dependencies.
5. Update the `initramfs` on your instance to ensure that the new module loads at boot time.
6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names](#) on the freedesktop.org website.
 - a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command:

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+|^udev-[0-9]\+'
systemd-208-11.e17_0.2.x86_64
```

In the above Red Hat 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

- b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

```
[ec2-user ~]$ sudo sed -i '/^GRUB_CMDLINE_LINUX/s/\ "$"/ net.ifnames
=0"/' /etc/default/grub
```

- c. Rebuild the grub configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To enable enhanced networking with ENA \(instance store-backed instances\) \(p. 626\)](#)

8. From your local computer, enable the enhanced networking `enaSupport` attribute using the following command.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --ena-
support
```

9. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 89\)](#). The AMI inherits the enhanced networking `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` may fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
11. (Optional) Connect to your instance and verify that the module is installed.

Note

If you are unable to connect to your instance after enabling enhanced networking with ENA, see [Troubleshooting the Elastic Network Adapter \(ENA\)](#) (p. 626).

To enabled enhanced networking with ENA (instance store-backed instances)

If your instance is an instance store-backed instance, follow the [Step 1](#) (p. 624) through the [Step 5](#) (p. 625) in the previous procedure, and then create a new AMI as described in [Creating an Instance Store-Backed Linux AMI](#) (p. 92). Be sure to enable the enhanced networking `enaSupport` attribute when you register the AMI.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --ena-support ...
```

Troubleshooting

For additional information about troubleshooting your ENA adapter, see [Troubleshooting the Elastic Network Adapter \(ENA\)](#) (p. 626).

Troubleshooting the Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) is designed to improve operating system health and reduce the chances of long-term disruption because of unexpected hardware behavior and or failures. The ENA architecture keeps device or driver failures as transparent to the system as possible. This topic provides troubleshooting information for ENA.

If you are unable to connect to your instance, start with the [Troubleshooting Connectivity Issues](#) (p. 627) section.

If you are able to connect to your instance, you can gather diagnostic information by using the failure detection and recovery mechanisms that are covered in the later sections of this topic.

Contents

- [Troubleshooting Connectivity Issues](#) (p. 627)

- [Keep-Alive Mechanism](#) (p. 628)
- [Register Read Timeout](#) (p. 629)
- [Statistics](#) (p. 629)
- [Driver Error Logs in syslog](#) (p. 631)

Troubleshooting Connectivity Issues

If you lose connectivity while enabling enhanced networking, the `ena` module might be incompatible with your instance's current running kernel. This can happen if you install the module for a specific kernel version (without `dkms`, or with an improperly configured `dkms.conf` file) and then your instance kernel is updated. If the instance kernel that is loaded at boot time does not have the `ena` module properly installed, your instance will not recognize the network adapter and your instance becomes unreachable.

If you enable enhanced networking for a PV instance or AMI, this can also make your instance unreachable.

If your instance becomes unreachable after enabling enhanced networking with ENA, you can disable the `enaSupport` attribute for your instance and it will fall back to the stock network adapter.

To disable enhanced networking with ENA (EBS-backed instances)

1. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To disable enhanced networking with ENA \(instance store-backed instances\)](#) (p. 627).

2. From your local computer, disable the enhanced networking attribute using the following command.

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

3. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
4. (Optional) Connect to your instance and try reinstalling the `ena` module with your current kernel version by following the steps in [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Linux Instances in a VPC](#) (p. 617).

To disable enhanced networking with ENA (instance store-backed instances)

If your instance is an instance store-backed instance, create a new AMI as described in [Creating an Instance Store-Backed Linux AMI](#) (p. 92). Be sure to disable the enhanced networking `enaSupport` attribute when you register the AMI.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --no-ena-support ...
```

Keep-Alive Mechanism

The ENA device posts keep-alive events at a fixed rate (usually once every second). The ENA driver implements a watchdog mechanism, which checks every for the presence of these keep-alive messages. If a message or messages are present, the watchdog is rearmed, otherwise the driver concludes that the device experienced a failure and then does the following:

- Dumps its current statistics to syslog
- Resets the ENA device
- Resets the ENA driver state

The above reset procedure may result in some traffic loss for a short period of time (TCP connections should be able to recover), but should not otherwise affect the user.

The ENA device may also indirectly request a device reset procedure, by not sending a keep-alive notification, for example, if the ENA device reaches an unknown state after loading an irrecoverable configuration.

Below is an example of the reset procedure:

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The
watchdog process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the
current statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.....
.....
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx
0x7 // At the end of the down process, the driver discards incomplete
packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version:
0.10 //The driver begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version:
0.0.1 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8]
Build date [Wed Apr 6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't
supported
```

```
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported  
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The  
reset process is complete
```

Register Read Timeout

The ENA architecture suggests a limited usage of memory mapped I/O (MMIO) read operations. MMIO registers are accessed by the ENA device driver only during its initialization procedure.

If the driver logs (available in **dmesg** output) indicate failures of read operations, this may be caused by an incompatible or incorrectly compiled driver, a busy hardware device, or hardware failure.

Intermittent log entries that indicate failures on read operations should not be considered an issue; the driver will retry them in this case. However, a sequence of log entries containing read failures indicate a driver or hardware problem.

Below is an example of driver log entry indicating a read operation failure due to a timeout:

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for  
timeout. expected: req id[1] offset[88] actual: req id[57006] offset[0]  
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for  
timeout. expected: req id[2] offset[8] actual: req id[57007] offset[0]  
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

Statistics

If you experience insufficient network performance or latency issues, you should retrieve the device statistics and examine them. These statistics can be obtained using **ethtool**, as shown below:

```
[ec2-user ~]$ ethtool -S ethN  
NIC statistics:  
tx_timeout: 0  
io_suspend: 0  
io_resume: 0  
wd_expired: 0  
interface_up: 1  
interface_down: 0  
admin_q_pause: 0  
queue_0_tx_cnt: 4329  
queue_0_tx_bytes: 1075749  
queue_0_tx_queue_stop: 0  
...
```

The following command output parameters are described below:

tx_timeout: *N*

The number of times that the Netdev watchdog was activated.

io_suspend: *N*

Unsupported. This value should always be zero.

io_resume: *N*

Unsupported. This value should always be zero.

wd_expired: *N*

The number of times that the driver did not receive the keep-alive event in the preceding 3 seconds.

interface_up: *N*

The number of times that the ENA interface was brought up.

`interface_down`: *N*
The number of times that the ENA interface was brought down.

`admin_q_pause`: *N*
The admin queue is in an unstable state. This value should always be zero.

`queue_N_tx_cnt`: *N*
The number of transmitted packets for queue *N*.

`queue_N_tx_bytes`: *N*
The number of transmitted bytes for queue *N*.

`queue_N_tx_queue_stop`: *N*
The number of times that queue *N* was full and stopped.

`queue_N_tx_queue_wakeup`: *N*
The number of times that queue *N* resumed after being stopped.

`queue_N_tx_dma_mapping_err`: *N*
Direct memory access error count. If this value is not 0, it indicates low system resources.

`queue_N_tx_napi_comp`: *N*
The number of times the napi handler called `napi_complete` for queue *N*.

`queue_N_tx_poll`: *N*
The number of times the napi handler was scheduled for queue *N*.

`queue_N_tx_doorbells`: *N*
The number of transmission doorbells for queue *N*.

`queue_N_tx_linearize`: *N*
The number of times SKB linearization was attempted for queue *N*.

`queue_N_tx_linearize_failed`: *N*
The number of times SKB linearization failed for queue *N*.

`queue_N_tx_prepare_ctx_err`: *N*
The number of times `ena_com_prepare_tx` failed for queue *N*. This value should always be zero; if not, see the driver logs.

`queue_N_tx_missing_tx_comp`: *N*
The number of packets that were left uncompleted for queue *N*. This value should always be zero.

`queue_N_tx_bad_req_id`: *N*
Invalid `req_id` for queue *N*. The valid `req_id` is zero, minus the `queue_size`, minus 1.

`queue_N_rx_cnt`: *N*
The number of received packets for queue *N*.

`queue_N_rx_bytes`: *N*
The number of received bytes for queue *N*.

`queue_N_rx_refil_partial`: *N*
The number of times the driver did not succeed in refilling the empty portion of the `rx` queue with the buffers for queue *N*. If this value is not zero, it indicates low memory resources.

`queue_N_rx_bad_csum`: *N*
The number of times the `rx` queue had a bad checksum for queue *N* (only if `rx` checksum offload is supported).

`queue_N_rx_page_alloc_fail`: *N*
The number of time that page allocation failed for queue *N*. If this value is not zero, it indicates low memory resources.

`queue_N_rx_skb_alloc_fail`: *N*
The number of time that SKB allocation failed for queue *N*. If this value is not zero, it indicates low system resources.

`queue_N_rx_dma_mapping_err`: *N*
Direct memory access error count. If this value is not 0, it indicates low system resources.

`queue_N_rx_bad_desc_num`: *N*
Too many buffers per packet. If this value is not 0, it indicates usage of very small buffers.

`queue_N_rx_small_copy_len_pkt`: *N*

Optimization: For packets smaller than this threshold, which is set by `sysfs`, the packet is copied directly to the stack to avoid allocation of a new page.

`ena_admin_q_aborted_cmd`: *N*

The number of admin commands that were aborted. This usually happens during the auto-recovery procedure.

`ena_admin_q_submitted_cmd`: *N*

The number of admin queue doorbells.

`ena_admin_q_completed_cmd`: *N*

The number of admin queue completions.

`ena_admin_q_out_of_space`: *N*

The number of times that the driver tried to submit new admin command, but the queue was full.

`ena_admin_q_no_completion`: *N*

The number of times that the driver did not get an admin completion for a command.

Driver Error Logs in `syslog`

The ENA driver writes log messages to **syslog** during system boot. You can examine these logs to look for errors if you are experiencing issues. Below is an example of information logged by the ENA driver in **syslog** during system boot, along with some annotations for select messages.

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 478.416939] [ENA_COM:
ena_com_validate_version] ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 478.420915] [ENA_COM:
ena_com_validate_version] ena controller version: 0.0.1 implementation
version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.256831] ena 0000:00:03.0:
Device watchdog is Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.672947] ena 0000:00:03.0:
creating 8 io queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.680885] [ENA_COM:
ena_com_init_interrupt_moderation] Feature 20 isn't supported // Interrupt
moderation is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.691609] [ENA_COM:
ena_com_get_feature_ex] Feature 10 isn't supported // RSS HASH function
configuration is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.694583] [ENA_COM:
ena_com_get_feature_ex] Feature 18 isn't supported //RSS HASH input source
configuration is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.697433] [ENA_COM:
ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.701064] ena 0000:00:03.0
(unnamed net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 479.704917] ena 0000:00:03.0:
Elastic Network Adapter (ENA) found at mem f3000000, mac addr
02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 480.805037] EXT4-fs (xvda1): re-
mounted. Opts: (null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [ 481.025842] NET: Registered
protocol family 10
```

Which errors can I ignore?

The following warnings that may appear in your system's error logs can be ignored for the Elastic Network Adapter:

Set host attribute isn't supported

Host attributes are not supported for this device.

failed to alloc buffer for rx queue

This is a recoverable error, and it indicates that there may have been a memory pressure issue when the error was thrown.

Feature `X` isn't supported

The referenced feature is not supported by the Elastic Network Adapter. Possible values for `X` include:

- 10: RSS Hash function configuration is not supported for this device.
- 12: RSS Indirection table configuration is not supported for this device.
- 18: RSS Hash Input configuration is not supported for this device.
- 20: Interrupt moderation is not supported for this device.

Failed to config AENQ

The Elastic Network Adapter does not support AENQ configuration.

Trying to set unsupported AENQ events

This error indicates an attempt to set an AENQ events group that is not supported by the Elastic Network Adapter.

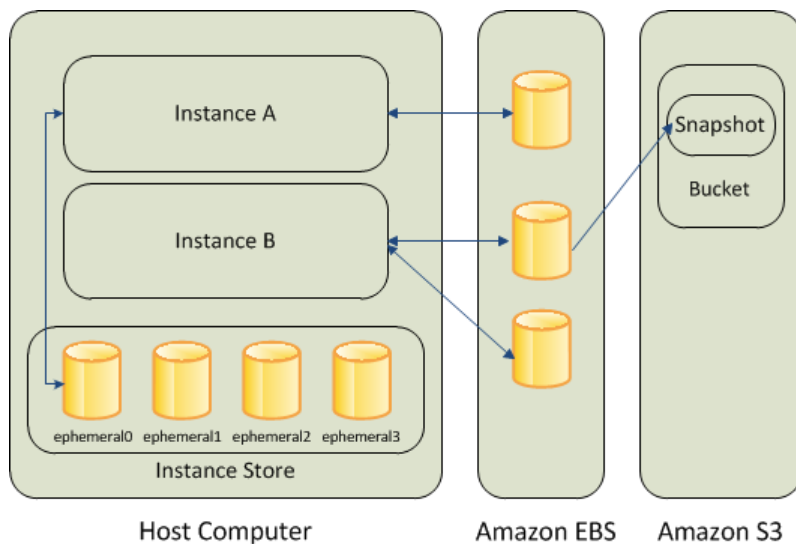
Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon EC2 to meet your specific requirements. These storage options include the following:

- [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 634)
- [Amazon EC2 Instance Store](#) (p. 715)
- [Amazon Simple Storage Service \(Amazon S3\)](#) (p. 726)

The following figure shows the relationship between these types of storage.



Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see [Amazon EBS Encryption](#) (p. 695).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance. For more information, see [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 634).

Amazon EC2 Instance Store

Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see [Amazon EC2 Instance Store](#) (p. 715).

Amazon S3

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. For more information, see [Amazon Simple Storage Service \(Amazon S3\)](#) (p. 726).

Adding Storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see [Block Device Mapping](#) (p. 731).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS Volume to an Instance](#) (p. 654).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the [Projecting Costs](#) section of the [Amazon Elastic Block Store page](#).

Amazon EBS is recommended when data must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

For simplified data encryption, you can launch your EBS volumes as encrypted volumes. Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, manage, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and

snapshots created from the volume are all encrypted. The encryption occurs on the servers that hosts EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see [Amazon EBS Encryption \(p. 695\)](#).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see [Request to Increase the Amazon EBS Volume Limit](#).

Contents

- [Features of Amazon EBS \(p. 635\)](#)
- [Amazon EBS Volumes \(p. 636\)](#)
- [Amazon EBS Snapshots \(p. 685\)](#)
- [Amazon EBS–Optimized Instances \(p. 692\)](#)
- [Amazon EBS Encryption \(p. 695\)](#)
- [Amazon EBS Volume Performance on Linux Instances \(p. 699\)](#)

Features of Amazon EBS

- You can create EBS General Purpose SSD ([gp2](#)), Provisioned IOPS SSD ([io1](#)), Throughput Optimized HDD ([st1](#)), and Cold HDD ([sc1](#)) volumes up to 16 TiB in size. You can mount these volumes as devices on your Amazon EC2 instances. You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. For more information, see [Creating an Amazon EBS Volume \(p. 650\)](#).
- With General Purpose SSD ([gp2](#)) volumes, you can expect base performance of 100 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. [gp2](#) volumes are ideal for a broad range of use cases such as boot volumes, small and medium-size databases, and development and test environments. [gp2](#) volumes support up to 10,000 IOPS and 160 MB/s of throughput. For more information, see [General Purpose SSD \(\[gp2\]\(#\)\) Volumes \(p. 640\)](#).
- With Provisioned IOPS SSD ([io1](#)) volumes, you can provision a specific level of I/O performance. [io1](#) volumes support up to 20,000 IOPS and 320 MB/s of throughput. This allows you to predictably scale to tens of thousands of IOPS per EC2 instance. For more information, see [Provisioned IOPS SSD \(\[io1\]\(#\)\) Volumes \(p. 643\)](#).
- Throughput Optimized HDD ([st1](#)) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 500 MiB/s, this volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. For more information, see [Throughput Optimized HDD \(\[st1\]\(#\)\) Volumes \(p. 644\)](#).
- Cold HDD ([sc1](#)) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 250 MiB/s, [sc1](#) is a good fit ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, [sc1](#) provides inexpensive block storage. For more information, see [Cold HDD \(\[sc1\]\(#\)\) Volumes \(p. 646\)](#).
- EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive). For

more information on creating file systems and mounting volumes, see [Making an Amazon EBS Volume Available for Use](#) (p. 655).

- You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. For more information, see [Amazon EBS Encryption](#) (p. 695).
- You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions. For more information, see [Amazon EBS Snapshots](#) (p. 685).
- EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that region. You can copy snapshots to other regions and then restore them to new volumes there, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery. For more information, see [Creating an Amazon EBS Snapshot](#) (p. 686), [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 652), and [Copying an Amazon EBS Snapshot](#) (p. 688).
- A large repository of public data set snapshots can be restored to EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see [Using Public Data Sets](#) (p. 740).
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need. For more information, see [Amazon EBS Volume Performance on Linux Instances](#) (p. 699).

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application, or for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance. After a volume is attached to an instance, you can use it like any other physical hard drive. Amazon EBS provides the following volume types: General Purpose SSD (`gp2`), Provisioned IOPS SSD (`io1`), Throughput Optimized HDD (`st1`), Cold HDD (`sc1`), and Magnetic (`standard`). They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see [Amazon EBS Volume Types](#) (p. 638).

Contents

- [Benefits of Using EBS Volumes](#) (p. 637)
- [Amazon EBS Volume Types](#) (p. 638)
- [Creating an Amazon EBS Volume](#) (p. 650)
- [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 652)
- [Attaching an Amazon EBS Volume to an Instance](#) (p. 654)
- [Making an Amazon EBS Volume Available for Use](#) (p. 655)
- [Viewing Volume Information](#) (p. 657)
- [Monitoring the Status of Your Volumes](#) (p. 658)
- [Detaching an Amazon EBS Volume from an Instance](#) (p. 669)
- [Deleting an Amazon EBS Volume](#) (p. 671)
- [Expanding the Storage Space of an EBS Volume on Linux](#) (p. 671)

- [Expanding a Linux Partition \(p. 676\)](#)

Benefits of Using EBS Volumes

EBS volumes provide several benefits that are not supported by instance store volumes.

- **Data availability**

When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive; the instance can format the EBS volume with a file system, such as ext3, and then install applications.

An EBS volume can be attached to only one instance at a time within the same Availability Zone. However, multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your EBS volumes at no additional charge (this includes data for the root device volumes for EBS-backed instances). For more information, see [Monitoring Volumes with CloudWatch \(p. 658\)](#).

- **Data persistence**

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

By default, EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume protected by Amazon EBS encryption. For more information, see [Amazon EBS Encryption \(p. 695\)](#).

By default, EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

- **Data encryption**

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. All EBS volume types support encryption. You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to Amazon EBS storage. For more information, see [Amazon EBS Encryption \(p. 695\)](#).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you

automatically. This key is used for Amazon EBS encryption unless you select a customer master key (CMK) that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

- **Snapshots**

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new EBS volumes, expand the size of a volume, or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are restored from encrypted snapshots are automatically encrypted. By optionally specifying a different volume size or a different Availability Zone, you can use this functionality to increase the size of an existing volume or to create duplicate volumes in new Availability Zones. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see [Tagging Your Amazon EC2 Resources \(p. 751\)](#).

Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volume types fall into two categories:

- SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
- HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

The following table describes the use cases and performance characteristics for each volume type:

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-	Lowest cost HDD volume designed for less frequently accessed workloads

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
	of transactional workloads		intensive workloads	
Use Cases	<ul style="list-style-type: none"> Recommended for most workloads System boot volumes Virtual desktops Low-latency interactive apps Development and test environments 	<ul style="list-style-type: none"> Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume Large database workloads, such as: <ul style="list-style-type: none"> MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle 	<ul style="list-style-type: none"> Streaming workloads requiring consistent, fast throughput at a low price Big data Data warehouses Log processing Cannot be a boot volume 	<ul style="list-style-type: none"> Throughput-oriented storage for large volumes of data that is infrequently accessed Scenarios where the lowest storage cost is important Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	10,000	20,000	500	250
Max. Throughput/Volume†	160 MiB/s	320 MiB/s	500 MiB/s	250 MiB/s
Max. IOPS/Instance	48,000	48,000	48,000	48,000
Max. Throughput/Instance	800 MiB/s	800 MiB/s	800 MiB/s	800 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

*Default volume type

**gp2/io1 based on 16KiB I/O size, st1/sc1 based on 1 MiB I/O size

† To achieve this throughput, you must have an instance that supports it, such as r3.8xlarge or x1.32xlarge.

The following table describes previous-generation EBS volume types. If you need higher performance or performance consistency than previous-generation volumes can provide, we recommend that you

consider using General Purpose SSD ([gp2](#)) or other current volume types. For more information, see [Previous Generation Volumes](#).

Previous Generation Volumes	
Volume Type	EBS Magnetic
Description	Previous generation HDD
Use Cases	Workloads where data is infrequently accessed
API Name	standard
Volume Size	1 GiB-1 TiB
Max. IOPS/Volume	40-200
Max. Throughput/Volume	40-90 MiB/s
Max. IOPS/Instance	48,000
Max. Throughput/Instance	800 MiB/s
Dominant Performance Attribute	IOPS

Note

Linux AMIs require GPT partition tables and GRUB 2 for boot volumes 2 TiB (2048 GiB) or larger. Many Linux AMIs today use the MBR partitioning scheme, which only supports up to 2047 GiB boot volumes. If your instance does not boot with a boot volume that is 2 TiB or larger, the AMI you are using may be limited to a 2047 GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances.

There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your EBS volumes, see [Amazon EBS Volume Performance on Linux Instances \(p. 699\)](#).

For more information about pricing for these volume types, see [Amazon EBS Pricing](#).

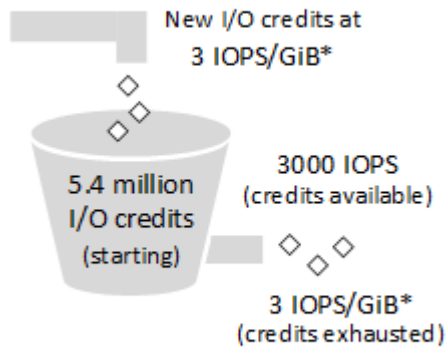
General Purpose SSD ([gp2](#)) Volumes

General Purpose SSD ([gp2](#)) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. A [gp2](#) volume can range in size from 1 GiB to 16 TiB.

I/O Credits and Burst Performance

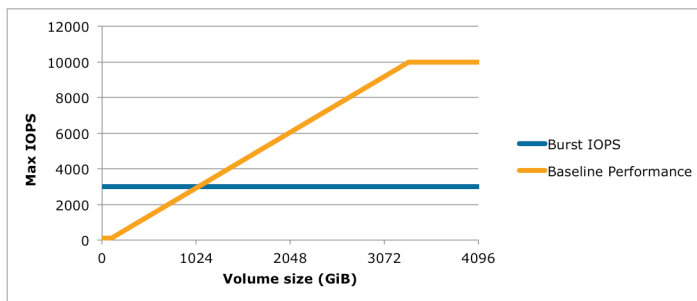
The performance of [gp2](#) volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your [gp2](#) volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for [gp2](#).

GP2 burst bucket



* Scaling linearly between minimum 100 IOPS and maximum 10,000 IOPS.

Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB `gp2` volume has a baseline performance of 300 IOPS.



When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. Volumes larger than 1,000 GiB have a baseline performance that is equal or greater than the maximum burst performance, and their I/O credit balance never depletes. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5.4 million I/O credits).

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1862	54,000
100	300	2,000	18,000
214 (Min. size for max. throughput)	642	2,290	8,412

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
3,334 (Min. size for max. IOPS)	10,000	N/A*	N/A*
16,384 (16 TiB, max. volume size)	10,000	N/A*	N/A*

* Bursting and I/O credits are only relevant to volumes under 1,000 GiB, where burst performance exceeds baseline performance.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the following equation:

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

What happens if I empty my I/O credit balance?

If your gp2 volume uses all of its I/O credit balance, the maximum IOPS performance of the volume will remain at the baseline IOPS performance level (the rate at which your volume earns credits) and the throughput limit is reduced to the baseline IOPS multiplied by the maximum throughput and divided by 3,000, until I/O demand drops below the baseline level and unused credits are added to the I/O credit balance.

$$\text{Throughput limit with empty I/O credit balance} = \frac{(\text{Baseline IOPS}) \times (\text{Max throughput})}{3,000}$$

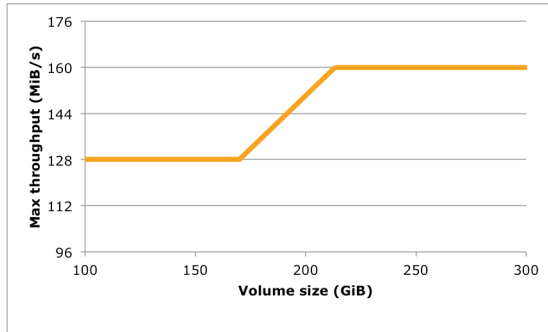
For example, a 100 GiB gp2 volume with an empty credit balance has a baseline IOPS performance limit of 300 IOPS and a throughput limit of 12.8 MiB/s ((128 MiB/s)*300/3000). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see [I/O Characteristics \(p. 704\)](#).

If you notice that your volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), you should consider using a larger gp2 volume (with a higher baseline performance level) or switching to an io1 volume for workloads that require sustained IOPS performance greater than 10,000 IOPS.

Throughput Performance

The throughput limit for gp2 volumes is 128 MiB/s for volumes less than or equal to 170 GiB; for volumes over 170 GiB, this limit increases at the rate of 768 KiB/s per GiB to a maximum of 160 MiB/s (at 214 GiB and larger).

Amazon Elastic Compute Cloud User Guide for Linux Instances EBS Volumes

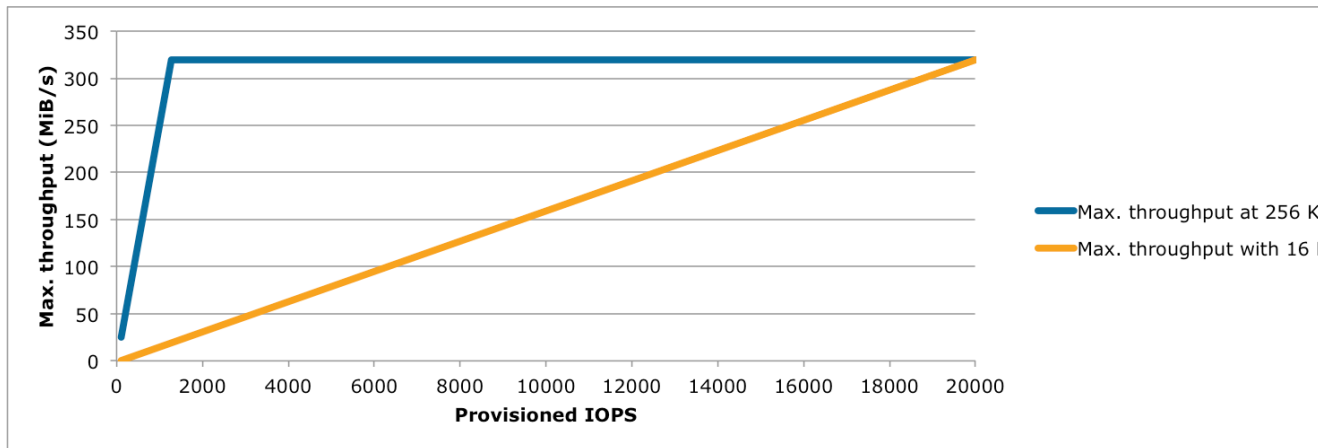


Provisioned IOPS SSD (i_o1) Volumes

Provisioned IOPS SSD (i_o1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Instead of using a bucket and credit model to calculate performance, an i_o1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An i_o1 volume can range in size from 4 GiB to 16 TiB and you can provision up to 20,000 IOPS per volume. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1. For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. Any volume 400 GiB in size or greater allows provisioning up to the 20,000 IOPS maximum.

The throughput limit of i_o1 volumes is 256 KiB for each IOPS provisioned, up to a maximum of 320 MiB/s (at 1,280 IOPS).



Your per-I/O latency experience depends on the IOPS provisioned and your workload pattern. For the best per-I/O latency experience, we recommend that you provision an IOPS-to-GiB ratio greater than 2:1. For example, a 2,000 IOPS volume should be smaller than 1,000 GiB.

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (i_o1) volumes. If you are unable to create an i_o1 volume (or launch an instance with an i_o1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports i_o1 volumes by creating a 4 GiB i_o1 volume in that zone.

Throughput Optimized HDD (st1) Volumes

Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. Bootable st1 volumes are not supported.

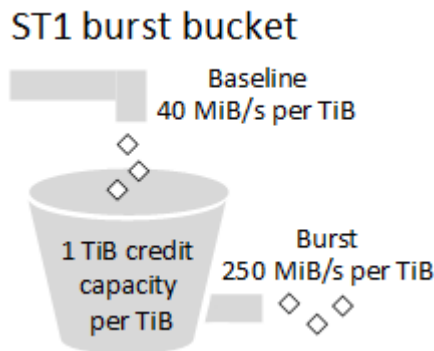
Note

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [Inefficiency of Small Read/Writes on HDD](#) (p. 649).

Throughput Credits and Burst Performance

Like gp2, st1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it will be able to drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for st1.



Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Baseline Throughput}$$

For a 1 TiB st1 volume, burst throughput is limited to 250 MiB/s, the bucket fills with credits at 40 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 500 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 40 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 20 to a cap of 500 MiB/s, which is reached at 12.5 TiB because

$$12.5 \text{ TiB} \times \frac{40 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

Burst throughput varies from 125 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB because

$$250 \text{ MiB/s}$$

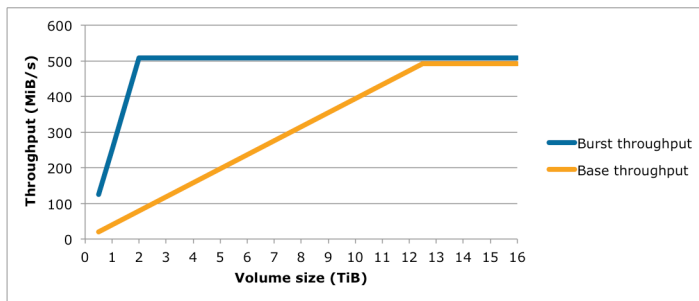
**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

$$2 \text{ TiB} \times \frac{\text{-----}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

The following table states the full range of base and burst throughput values for `st1`:

Volume Size (TiB)	ST1 Base Throughput (MiB/s)	ST1 Burst Throughput (MiB/s)
0.5	20	125
1	40	250
2	80	500
3	120	500
4	160	500
5	200	500
6	240	500
7	280	500
8	320	500
9	360	500
10	400	500
11	440	500
12	480	500
12.5	500	500
13	500	500
14	500	500
15	500	500
16	500	500

The following diagram plots the table values:



Note

Throughput for an `st1` volume is also capped at the baseline while a snapshot is being created.

Cold HDD (sc1) Volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. Bootable sc1 volumes are not supported.

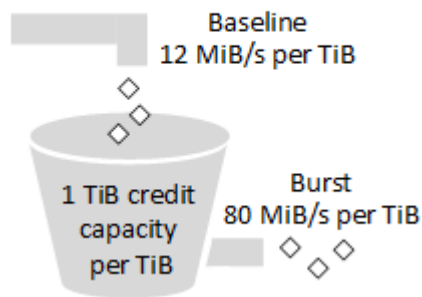
Note

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [Inefficiency of Small Read/Writes on HDD](#) (p. 649).

Throughput Credits and Burst Performance

Like gp2, sc1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it will be able to drive I/O at the burst level.

SC1 burst bucket



Subject to throughput and throughput-credit caps, the available throughput of an sc1 volume is expressed by the following formula:

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Baseline Throughput}$$

For a 1 TiB sc1 volume, burst throughput is limited to 80 MiB/s, the bucket fills with credits at 12 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 250 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 12 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 6 MiB/s to a maximum of 192 MiB/s, which is reached at 16 TiB because

$$16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB because

$$3.125 \text{ TiB} \times \frac{250 \text{ MiB/s}}{3.125 \text{ TiB}} = 250 \text{ MiB/s}$$

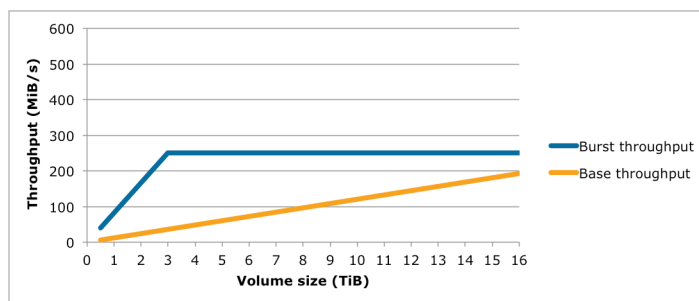
**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

1 TiB

The following table states the full range of base and burst throughput values for `sc1`:

Volume Size (TiB)	SC1 Base Throughput (MiB/s)	SC1 Burst Throughput (MiB/s)
0.5	6	40
1	12	80
2	24	160
3	36	240
3.125	37.5	250
4	48	250
5	60	250
6	72	250
7	84	250
8	96	250
9	108	250
10	120	250
11	132	250
12	144	250
13	156	250
14	168	250
15	180	250
16	192	250

The following diagram plots the table values:



Magnetic (standard)

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes

deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB.

Note

Magnetic is a Previous Generation Volume. For new applications, we recommend using one of the newer volume types. For more information, see [Previous Generation Volumes](#).

Performance Considerations When Using HDD Volumes

For optimal throughput results using HDD volumes, plan your workloads with the following considerations in mind.

Throughput Optimized HDD vs. Cold HDD

The `st1` and `sc1` bucket sizes vary according to volume size, and a full bucket contains enough tokens for a full volume scan. However, larger `st1` and `sc1` volumes take longer for the volume scan to complete due to per-instance and per-volume throughput limits. Volumes attached to smaller instances are limited to the per-instance throughput rather than the `st1` or `sc1` throughput limits.

Both `st1` and `sc1` are designed for performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour.

The following table shows ideal scan times for volumes of various size, assuming full buckets and sufficient instance throughput.

In general, scan times are expressed by this formula:

$$\frac{\text{Volume size}}{\text{Throughput}} = \text{Scan time}$$

For example, taking the performance consistency guarantees and other optimizations into account, an `st1` customer with a 5 TiB volume can expect to complete a full volume scan in 2.91 to 3.27 hours.

$$\frac{5 \text{ TiB}}{500 \text{ MiB/s}} = \frac{5 \text{ TiB}}{0.00047684 \text{ TiB/s}} = 10,486 \text{ s} = 2.91 \text{ hours (optimal)}$$

$$2.91 \text{ hours} + \frac{2.91 \text{ hours}}{(0.90)(0.99)} = 3.27 \text{ hours (minimum expected)}$$

of the time <-- From expected performance of 90% of burst 99%

Similarly, an `sc1` customer with a 5 TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

$$\frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} = 20972 \text{ s} = 5.83 \text{ hours (optimal)}$$

$$5.83 \text{ hours} + \frac{5.83 \text{ hours}}{(0.90)(0.99)} = 6.54 \text{ hours (minimum expected)}$$

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

Volume Size (TiB)	ST1 Scan Time with Burst (Hours)*	SC1 Scan Time with Burst (Hours)*
1	1.17	3.64
2	1.17	3.64
3	1.75	3.64
4	2.33	4.66
5	2.91	5.83
6	3.50	6.99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6.99	13.98
13	7.57	15.15
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

* These scan times assume an average queue depth (rounded to the nearest whole number) of four or more when performing 1 MiB of sequential I/O.

Therefore if you have a throughput-oriented workload that needs to complete scans quickly (up to 500 MiB/s), or requires several full volume scans a day, use `st1`. If you are optimizing for cost, your data is relatively infrequently accessed, and you don't need more than 250 MiB/s of scanning performance, then use `sc1`.

Inefficiency of Small Read/Writes on HDD

The performance model for `st1` and `sc1` volumes is optimized for sequential I/Os, favoring high-throughput workloads, offering acceptable performance on workloads with mixed IOPS and throughput, and discouraging workloads with small, random I/O.

For example, an I/O request of 1 MiB or less counts as a 1 MiB I/O credit. However, if the I/Os are sequential, they are merged into 1 MiB I/O blocks and count only as a 1 MiB I/O credit.

Limitations on per-Instance Throughput

Throughput for `st1` and `sc1` volumes will always be determined by the smaller of the following:

- Throughput limits of the volume
- Throughput limits of the instance

As for all Amazon EBS volumes, we recommend that you select an appropriate EBS-optimized EC2 instance in order to avoid network bottlenecks. For more information, see [Amazon EBS-Optimized Instances](#).

Creating an Amazon EBS Volume

You can create an Amazon EBS volume that you can then attach to any EC2 instance within the same Availability Zone. You can choose to create an encrypted EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types](#) (p. 697).

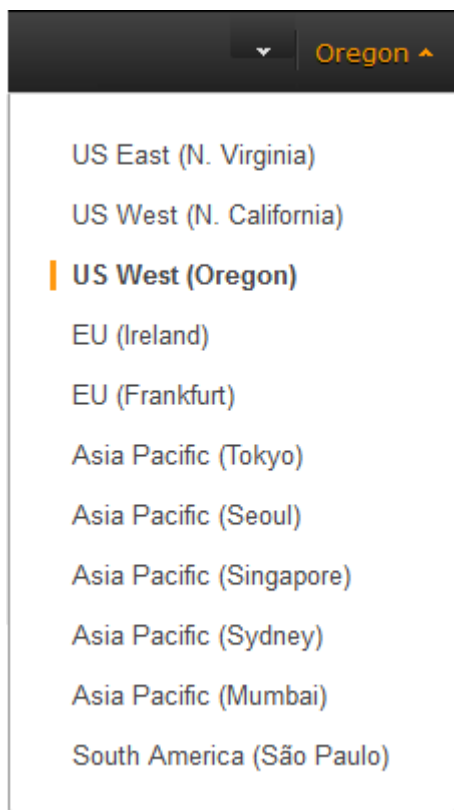
You can also create and attach EBS volumes when you launch instances by specifying a block device mapping. For more information, see [Launching an Instance](#) (p. 279) and [Block Device Mapping](#) (p. 731). You can restore volumes from previously created snapshots. For more information, see [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 652).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS SSD (`io1`) volume and attach it to an instance with enough bandwidth to support your application, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. The same advice holds for Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) volumes. For more information, see [Amazon EC2 Instance Configuration](#) (p. 701).

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once. For more information, see [Initializing Amazon EBS Volumes](#) (p. 706).

To create an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 743).



3. In the navigation pane, under **ELASTIC BLOCK STORE**, choose **Volumes**.
4. Above the upper pane, choose **Create Volume**.
5. In the **Create Volume** dialog box, for **Volume Type**, select **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types \(p. 638\)](#).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (i_o1) volumes. If you are unable to create an i_o1 volume (or launch an instance with an i_o1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports i_o1 volumes by creating a 4 GiB i_o1 volume in that zone.

6. For **Size**, enter the size of the volume, in GiB.
7. For i_o1 volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume should support.
8. For **Availability Zone**, select the Availability Zone in which to create the volume.
9. (Optional) To create an encrypted volume, select the **Encrypted** box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account, or you can choose any Customer Master Key (CMK) that you have previously created using the AWS Key Management Service. Available keys are visible in the **Master Key** drop down menu, or you can paste the full ARN of any key that you have access to. For more information, see the [AWS Key Management Service Developer Guide](#).

Note

Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 697\)](#).

10. Choose **Yes, Create**.

Important

If you receive one of the following errors, the current volume creation would exceed the default storage limit for your account:

```
Maximum number of active volumes bytes, 20, exceeded.  
Maximum number of active gp2 volumes bytes, 20, exceeded.  
Maximum number of active io1 volumes bytes, 20, exceeded.
```

To view the default service limits for Amazon EBS, see [Amazon Elastic Block Store \(Amazon EBS\) Limits](#) in the *Amazon Web Services General Reference*. To request an increase in your storage limits, see [Request to Increase the Amazon EBS Volume Limit](#).

To create an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (AWS Tools for Windows PowerShell)

Restoring an Amazon EBS Volume from a Snapshot

You can restore an Amazon EBS volume with data from a snapshot stored in Amazon S3. You need to know the ID of the snapshot you wish to restore your volume from and you need to have access permissions for the snapshot. For more information on snapshots, see [Amazon EBS Snapshots \(p. 685\)](#).

New volumes created from existing EBS snapshots load lazily in the background. This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before your attached instance can start accessing the volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and continues loading the rest of the data in the background.

EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 697\)](#).

Because of security constraints, you cannot directly restore an EBS volume from a shared encrypted snapshot that you do not own. You must first create a copy of the snapshot, which you will own. You can then restore a volume from that copy. For more information, see [Amazon EBS Encryption](#).

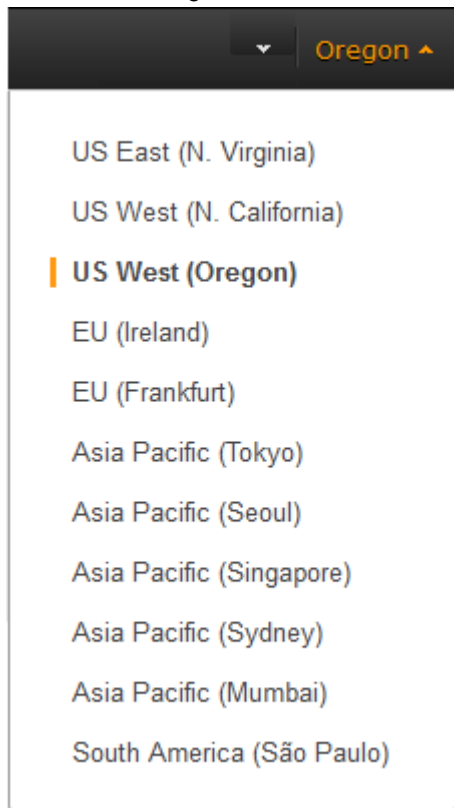
New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. Performance is restored after the data is accessed once.

For most applications, amortizing the initialization cost over the lifetime of the volume is acceptable. If you need to ensure that your restored volume always functions at peak capacity in production, you can force the immediate initialization of the entire volume using **dd** or **fiio**. For more information, see [Initializing Amazon EBS Volumes \(p. 706\)](#).

To restore an EBS volume from a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. From the navigation bar, select the region that your snapshot is located in. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 743). If you need to restore the snapshot to a volume in a different region, you can copy your snapshot to the new region and then restore it to a volume in that region. For more information, see [Copying an Amazon EBS Snapshot](#) (p. 688).



3. In the navigation pane, choose **Volumes, Create Volume**.
4. In the **Create Volume** dialog box, for **Volume Type**, choose **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types](#) (p. 638).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (io1) volumes. If you are unable to create an io1 volume (or launch an instance with an io1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports io1 volumes by creating a 4 GiB io1 volume in that zone.

5. For **Snapshot**, start typing the ID or description of the snapshot from which you are restoring the volume, and select it from the list of suggested options.

Note

Volumes that are restored from encrypted snapshots can only be attached to instances that support Amazon EBS encryption. For more information, see [Supported Instance Types](#) (p. 697).

6. For **Size**, enter the size of the volume in GiB, or verify that the default size of the snapshot is adequate.

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

7. For `io1` volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
8. In the **Availability Zone** list, select the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances within the same Availability Zone.
9. Choose **Yes, Create**.

Important

If you restored a snapshot to a larger volume than the default for that snapshot, you need to extend the file system on the volume to take advantage of the extra space. For more information, see [Expanding the Storage Space of an EBS Volume on Linux \(p. 671\)](#).

After you've restored a volume from a snapshot, you can attach it to an instance to begin using it. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).

To restore an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-volume` (AWS CLI)
- `New-EC2Volume` (AWS Tools for Windows PowerShell)

Attaching an Amazon EBS Volume to an Instance

You can attach an EBS volumes to one of your instances that is in the same Availability Zone as the volume.

Prerequisites

- Determine the device names that you'll use. For more information, see [Device Naming on Linux Instances \(p. 729\)](#).
- Determine how many volumes you can attach to your instance. For more information, see [Instance Volume Limits \(p. 728\)](#).
- If a volume is encrypted, it can only be attached to an instance that supports Amazon EBS encryption. For more information, see [Supported Instance Types \(p. 697\)](#).
- If a volume has an AWS Marketplace product code:
 - The volume can only be attached to a stopped instance.
 - You must be subscribed to the AWS Marketplace code that is on the volume.
 - The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
 - AWS Marketplace product codes are copied from the volume to the instance.

To attach an EBS volume to an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select a volume and choose **Actions, Attach Volume**.
4. In the **Attach Volume** dialog box, start typing the name or ID of the instance to attach the volume to for **Instance**, and select it from the list of suggestion options (only instances that are in the same Availability Zone as the volume are displayed).
5. You can keep the suggested device name, or enter a different supported device name.

Important

The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

6. Choose **Attach**.
7. Connect to your instance and make the volume available. For more information, see [Making an Amazon EBS Volume Available for Use](#) (p. 655).

To attach an EBS volume to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [attach-volume](#) (AWS CLI)
- [Add-EC2Volume](#) (AWS Tools for Windows PowerShell)

Making an Amazon EBS Volume Available for Use

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the EBS volume and is transparent to applications using the device.

Note that you can take snapshots of your EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see [Amazon EBS Snapshots](#) (p. 685).

Making the Volume Available on Linux

Use the following procedure to make the volume available. Note that you can get directions for volumes on a Windows instance from [Making the Volume Available on Windows](#) in the *Amazon EC2 User Guide for Windows Instances*.

To make an EBS volume available for use on Linux

1. Connect to your instance using SSH. For more information, see [Step 2: Connect to Your Instance](#) (p. 28).
2. Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of `/dev/sdh`, your device might be renamed `/dev/xvdh` or `/dev/hdh` by the kernel; in most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), even the trailing letter might also change (where `/dev/sda` could become `/dev/xvde`). In these cases, each device name trailing letter is incremented the same number of times. For example, `/dev/sdb` would become `/dev/xvdf` and `/dev/sdc` would become `/dev/xvdg`. Amazon Linux AMIs create a symbolic link with the name you specify at launch that points to the renamed device path, but other AMIs might behave differently.

Use the **lsblk** command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use.

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf 202:80  0 100G 0 disk
xvda1 202:1  0   8G  0 disk /
```

The output of `lsblk` removes the `/dev/` prefix from full device paths. In this example, `/dev/xvda1` is mounted as the root device (note the `MOUNTPOINT` is listed as `/`, the root of the Linux file system hierarchy), and `/dev/xvdf` is attached, but it has not been mounted yet.

3. Determine whether you need to create a file system on the volume. New volumes are raw block devices, and you need to create a file system on them before you can mount and use them. Volumes that have been restored from snapshots likely have a file system on them already; if you create a new file system on top of an existing file system, the operation overwrites your data. Use the `sudo file -s device` command to list special information, such as file system type.

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

If the output of the previous command shows simply `data` for the device, then there is no file system on the device and you need to create one. You can go on to [Step 4 \(p. 656\)](#). If you run this command on a device that contains a file system, then your output will be different.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-
a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge
files)
```

In the previous example, the device contains `Linux rev 1.0 ext4 filesystem data`, so this volume does not need a file system created (you can skip [Step 4 \(p. 656\)](#) if your output shows file system data).

4. (Conditional) Use the following command to create an ext4 file system on the volume. Substitute the device name (such as `/dev/xvdf`) for `device_name`. Depending on the requirements of your application or the limitations of your operating system, you can choose a different file system type, such as ext3 or XFS.

Caution

This step assumes that you're mounting an empty volume. If you're mounting a volume that already has data on it (for example, a volume that was restored from a snapshot), don't use `mkfs` before mounting the volume (skip to the next step instead). Otherwise, you'll format the volume and delete the existing data.

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

5. Use the following command to create a mount point directory for the volume. The mount point is where the volume is located in the file system tree and where you read and write files to after you mount the volume. Substitute a location for `mount_point`, such as `/data`.

```
[ec2-user ~]$ sudo mkdir mount_point
```

6. Use the following command to mount the volume at the location you just created.

```
[ec2-user ~]$ sudo mount device_name mount_point
```

7. (Optional) To mount this EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file.
 - a. Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while you are editing it.

```
[ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
```

- b. Open the `/etc/fstab` file using any text editor, such as **nano** or **vim**.

Note

You need to open the file as `root` or by using the **sudo** command.

- c. Add a new line to the end of the file for your volume using the following format.

```
device_name mount_point file_system_type fs_mntops fs_freq  
fs_passno
```

The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the following example for them (`defaults,nofail 0 2`). For more information on `/etc/fstab` entries, see the **fstab** manual page (by entering **man fstab** on the command line). For example, to mount the ext4 file system on the device `/dev/xvdf` at the mount point `/data`, add the following entry to `/etc/fstab`.

Note

If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the `nofail` mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add the `nobootwait` mount option.

```
/dev/xvdf /data ext4 defaults,nofail 0 2
```

- d. After you've added the new entry to `/etc/fstab`, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in `/etc/fstab`.

```
[ec2-user ~]$ sudo mount -a
```

If the previous command does not produce an error, then your `/etc/fstab` file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your `/etc/fstab`.

Warning

Errors in the `/etc/fstab` file can render a system unbootable. Do not shut down a system that has errors in the `/etc/fstab` file.

- e. (Optional) If you are unsure how to correct `/etc/fstab` errors, you can always restore your backup `/etc/fstab` file with the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

8. Review the file permissions of your new volume mount to make sure that your users and applications can write to the volume. For more information about file permissions, see [File security](#) at *The Linux Documentation Project*.

Viewing Volume Information

You can view descriptive information for your Amazon EBS volumes in a selected region at a time in the AWS Management Console. You can also view detailed information about a single volume, including the size, volume type, whether or not the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

To view information about an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. To view more information about a volume, select it.

To view information about an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volumes](#) (AWS CLI)
- [Get-EC2Volume](#) (AWS Tools for Windows PowerShell)

Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

Contents

- [Monitoring Volumes with CloudWatch \(p. 658\)](#)
- [Monitoring Volumes with Status Checks \(p. 661\)](#)
- [Monitoring Volume Events \(p. 663\)](#)
- [Working with an Impaired Volume \(p. 664\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 667\)](#)

Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for EBS-backed instances.
Detailed	Provisioned IOPS SSD (i _o 1) volumes automatically send one-minute metrics to CloudWatch.

When you get data from CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the CloudWatch API or the Amazon EC2 console. The console takes the raw data from the CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Amazon EBS Metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. Amazon EBS General Purpose SSD (gp2), Throughput Optimized HDD (st1), Cold HDD (sc1), and

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch. Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. For more information about how to monitor Amazon EBS, see [Monitoring the Status of Your Volumes](#) in the *Amazon EC2 User Guide for Linux Instances*.

The `AWS/EBS` namespace includes the following metrics.

Metric	Description
VolumeReadBytes VolumeWriteBytes	Provides information on the I/O operations in a specified period of time. The <code>Sum</code> statistic reports the total number of bytes transferred during the period. The <code>Average</code> statistic reports the average size of each I/O operation during the period. The <code>SampleCount</code> statistic reports the total number of I/O operations during the period. The <code>Minimum</code> and <code>Maximum</code> statistics are not relevant for this metric. Data is only reported to Amazon CloudWatch when the volume is active. If the volume is idle, no data is reported to Amazon CloudWatch. Units: Bytes
VolumeReadOps VolumeWriteOps	The total number of I/O operations in a specified period of time. Note To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count
VolumeTotalReadTime VolumeTotalWriteTime	The total number of seconds spent by all operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. Units: Seconds
VolumeIdleTime	The total number of seconds in a specified period of time when no read or write operations were submitted. Units: Seconds
VolumeQueueLength	The number of read and write operation requests waiting to be completed in a specified period of time. Units: Count
VolumeThroughputPercentage	Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. Note During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time).

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

Metric	Description
	Units: Percent
VolumeConsumedReadWriteIOPS	<p>Used with Provisioned IOPS SSD volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time.</p> <p>I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS.</p> <p>Units: Count</p>
BurstBalance	<p>Used with Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes only. Provides information on the balance available in the burst bucket. Data is only reported to CloudWatch when the volume is active. If the volume is not attached, no data is reported.</p> <p>Units: Percent</p>

Dimensions for Amazon EBS Metrics

The only dimension that Amazon EBS sends to CloudWatch is the volume ID. This means that all available statistics are filtered by volume ID.

Graphs in the Amazon EC2 Console

After you create a volume, you can view the volume's monitoring graphs in the Amazon EC2 console. Select a volume on the **Volumes** page in the console and choose **Monitoring**. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph	Description using raw metrics
Read Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeReadBytes}) / \text{Period} / 1024$
Write Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (Ops/s)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$
Write Throughput (Ops/s)	$\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$
Avg Queue Length (ops)	$\text{Avg}(\text{VolumeQueueLength})$
% Time Spent Idle	$\text{Sum}(\text{VolumeIdleTime}) / \text{Period} * 100$
Avg Read Size (KiB/op)	$\text{Avg}(\text{VolumeReadBytes}) / 1024$
Avg Write Size (KiB/op)	$\text{Avg}(\text{VolumeWriteBytes}) / 1024$
Avg Read Latency (ms/op)	$\text{Avg}(\text{VolumeTotalReadTime}) * 1000$
Avg Write Latency (ms/op)	$\text{Avg}(\text{VolumeTotalWriteTime}) * 1000$

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

Monitoring Volumes with Status Checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is `impaired`. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as `fsck` (Linux) or `chkdsk` (Windows), before doing so.

Note

Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

If the consistency of a particular volume is not a concern for you, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for `io1` volumes that are attached to an instance and is not valid for General Purpose SSD (`gp2`), Throughput Optimized HDD (`st1`), Cold HDD (`sc1`), or Magnetic (`standard`) volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach a `io1` volume to an instance for this check to report the I/O performance status.

Important

While initializing `io1` volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a `warning` state in the **I/O Performance** status check. This is expected, and you can ignore the `warning` state on `io1` volumes while you are initializing them. For more information, see [Initializing Amazon EBS Volumes \(p. 706\)](#).

The following table lists statuses for Amazon EBS volumes.

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
<code>ok</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
<code>warning</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations)

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
		Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

To view and work with status checks, you can use the Amazon EC2 console, the API, or the command line interface.

To view status checks in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. On the **EBS Volumes** page, use the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and choose **Status Checks**.

Volumes: | **vol-d882c69b**

IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies

Description | **Status Checks** | Monitoring | Tags

Volume Status impaired

IO Status Disabled

Since December 23, 2013 7:06:41 PM UTC+2

Description Awaiting Action: Enable IO

Auto-Enabled IO Disabled [Edit](#)

[Find out more](#) about working with volume status checks and events.
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our [help page](#).

5. If you have a volume with a failed status check (status is `impaired`), see [Working with an Impaired Volume](#) (p. 664).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see [Monitoring Volume Events \(p. 663\)](#).

To view volume status information with the command line

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volume-status](#) (AWS CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

Monitoring Volume Events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the `AutoEnableIO` volume attribute. For more information about changing this attribute, see [Working with an Impaired Volume \(p. 664\)](#).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO

Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

IO Enabled

I/O operations were explicitly enabled for this volume.

IO Auto-Enabled

I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

Normal

For `io1` volumes only. Volume performance is as expected.

Degraded

For `io1` volumes only. Volume performance is below expectations.

Severely Degraded

For `io1` volumes only. Volume performance is well below expectations.

Stalled

For `io1` volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

To view events for your volumes in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. All instances and volumes that have events are listed. You can filter by volume to view only volume status. You can also filter on specific status types.

- Select a volume to view its specific event.

The screenshot shows the AWS Management Console interface for EBS volumes. At the top, there is an 'Actions' dropdown menu. Below it, a filter bar shows 'Volume resources', 'All event types', and 'Ongoing and scheduled'. A search bar is also present. A table lists volume resources with columns for Resource Name, Resource Type, Resource Id, Availability Zone, Event Type, and Event Status. The second row is selected, showing a volume with Resource Id 'vol-3682c675' in 'us-east-1d' availability zone, with an event type of 'potential-data-inconsistency' and a status of 'Awaiting Action: Enable IO'. Below the table, the event details for 'vol-3682c675' are displayed in a yellow warning box. The details include: Availability Zone: us-east-1d, Event Type: potential-data-inconsistency, Event Status: Awaiting Action: Enable IO, IO status: IO Disabled, Attached to: i-93aae4ea, Start Time: December 23, 2013 7:09:20 PM UTC+2, and End time. A link is provided to 'Find out more about monitoring volume events'.

If you have a volume where I/O is disabled, see [Working with an Impaired Volume \(p. 664\)](#). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

To view events for your volumes with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volume-status](#) (AWS CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

Options

- [Option 1: Perform a Consistency Check on the Volume Attached to its Instance](#) (p. 665)
- [Option 2: Perform a Consistency Check on the Volume Using Another Instance](#) (p. 666)
- [Option 3: Delete the Volume If You No Longer Need It](#) (p. 667)

Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

To perform a consistency check on an attached volume

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume on which to enable I/O operations.
 - d. In the details pane, choose **Enable Volume IO**.

Volumes: | vol-d882c69b

The screenshot shows the Amazon EC2 console interface for a specific EBS volume. At the top, there is a yellow warning banner with an exclamation mark icon and the text: "IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistency". Below the banner are four tabs: "Description", "Status Checks", "Monitoring", and "Tags". The "Description" tab is active, displaying the following details:

Volume ID	vol-d882c69b
Capacity	100 GiB
Created	November 21, 2013 3:42:01 PM UTC+2
State	available
Volume type	io1
Product codes	-

- e. In **Enable Volume IO**, choose **Yes, Enable**.
3. Check the data on the volume.
 - a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes you can contact support. Choose **Troubleshoot**, and then on the **Troubleshoot Status Checks** dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [enable-volume-io](#) (AWS CLI)
- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 2: Perform a Consistency Check on the Volume Using Another Instance

Use the following procedure to check the volume outside your production environment.

Important

This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume to detach.
 - d. Choose **Actions, Force Detach Volume**. You'll be prompted for confirmation.
3. Enable I/O on the volume.
 - a. In the navigation pane, choose **Volumes**.
 - b. Select the volume that you detached in the previous step.
 - c. In the details pane, choose **Enable Volume IO**.

Volumes: | vol-d882c69b

IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistency

Description	Status Checks	Monitoring	Tags
Volume ID	vol-d882c69b		
Capacity	100 GiB		
Created	November 21, 2013 3:42:01 PM UTC+2		
State	available		
Volume type	io1		
Product codes	-		

- d. In the **Enable Volume IO** dialog box, choose **Yes, Enable**.
4. Attach the volume to another instance. For information, see [Launch Your Instance \(p. 278\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).
 5. Check the data on the volume.
 - a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes, you can contact support. Choose **Troubleshoot**, and then in the troubleshooting dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [enable-volume-io](#) (AWS CLI)
- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 3: Delete the Volume If You No Longer Need It

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see [Deleting an Amazon EBS Volume \(p. 671\)](#).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 652\)](#).

Working with the AutoEnableIO Volume Attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see [Monitoring Volume Events \(p. 663\)](#).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

To view the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. In the lower pane, choose **Status Checks**.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, either `Enabled` or `Disabled`.

Volumes: | vol-d882c69b



IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies

Description

Status Checks

Monitoring

Tags

Volume Status impaired

IO Status Disabled

Since December 23, 2013 7:06:41 PM UTC+2

Description Awaiting Action: Enable IO

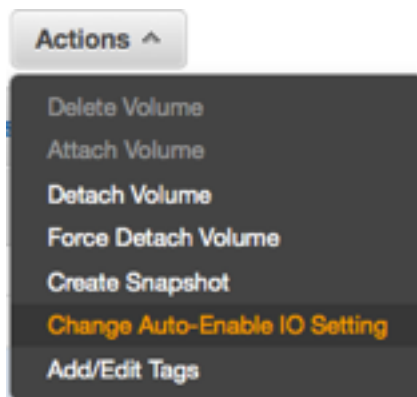
Auto-Enabled IO Disabled [Edit](#)

[Find out more](#) about working with volume status checks and events.

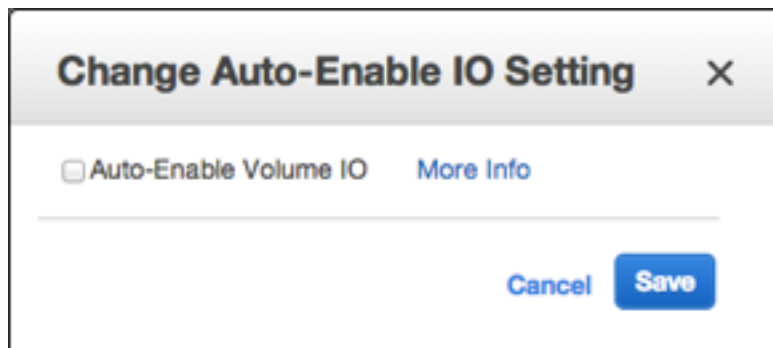
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our

To modify the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, choose **Actions**.
5. Choose **Change Auto-Enable IO Setting**.



6. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.



7. Choose **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, choose **Status Checks, Edit**.

To view or modify the `AutoEnableIO` attribute of a volume with the command line

You can use one of the following commands to view the `AutoEnableIO` attribute of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [describe-volume-attribute](#) (AWS CLI)
- [Get-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `AutoEnableIO` attribute of a volume, you can use one of the commands below.

- [modify-volume-attribute](#) (AWS CLI)
- [Edit-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

Detaching an Amazon EBS Volume from an Instance

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, if the instance is running, you must first unmount the volume from the instance.;

If an EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

Important

After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the AWS Free Tier. You must delete a volume to avoid incurring further charges. For more information, see [Deleting an Amazon EBS Volume](#) (p. 671).

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see [Viewing Volume Information](#) (p. 657).

Note that you can reattach a volume that you detached (without unmounting it), but it might not get the same mount point and the data on the volume might be out of sync if there were writes to the volume in progress when it was detached.

To detach an EBS volume using the console

1. Use the following command to unmount the `/dev/sdh` device.

```
[ec2-user ~]$ umount -d /dev/sdh
```

2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Volumes**.
4. Select a volume and choose **Actions, Detach Volume**.
5. In the confirmation dialog box, choose **Yes, Detach**.

To detach an EBS volume from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [detach-volume](#) (AWS CLI)
- [Dismount-EC2Volume](#) (AWS Tools for Windows PowerShell)

Troubleshooting

This section deals with common problems encountered when detaching volumes, and how to resolve them.

Note

To guard against the possibility of data loss, take a snapshot of your volume before attempting to unmount it. Forced detachment of a stuck volume can cause damage to the file system or the data it contains or an inability to attach a new volume using the same device name, unless you reboot the instance.

- If you encounter problems while detaching a volume through the Amazon EC2 console, it may be helpful to use the **describe-volumes** CLI command to diagnose the issue. For more information, see [describe-volumes](#).
- If your volume stays in the `detaching` state, you can force the detachment by choosing **Force Detach**. Use this option only as a last resort to detach a volume from a failed instance, or if you are detaching a volume with the intention of deleting it. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.
- If you've tried to force the volume to detach multiple times over several minutes and it stays in the `detaching` state, you can post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the volume ID and describe the steps that you've already taken.
- When you attempt to detach a volume that is still mounted, the volume can become stuck in the `busy` state while it is trying to detach. The following output from **describe-volumes** shows an example of this condition:

```
[ec2-user ~]$ aws ec2 describe-volumes --region us-west-2 --volume-ids
vol-1234abcd
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2b",
      "Attachments": [
        {
          "AttachTime": "2016-07-21T23:44:52.000Z",
          "InstanceId": "i-fedc9876",
          "VolumeId": "vol-1234abcd",
          "State": "busy",
          "DeleteOnTermination": false,
```

```
        "Device": "/dev/sdf"  
    }  
    . . .
```

When you encounter this state, detachment can be delayed indefinitely until you unmount the volume, force detachment, reboot the instance, or all three.

Deleting an Amazon EBS Volume

After you no longer need an Amazon EBS volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to re-create the volume later.

To delete a volume, it must be in the `available` state (not attached to an instance).

To delete an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select a volume and choose **Actions, Delete Volume**.
4. In the confirmation dialog box, choose **Yes, Delete**.

To delete an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-volume](#) (AWS CLI)
- [Remove-EC2Volume](#) (AWS Tools for Windows PowerShell)

Expanding the Storage Space of an EBS Volume on Linux

You can increase the storage space of an existing EBS volume without losing the data on the volume. To do this, you migrate your data to a larger volume and then extend the file system on the volume to recognize the newly-available space. After you verify that your new volume is working properly, you can delete the old volume.

Tasks

- [Migrating Your Data to a Larger Volume \(p. 672\)](#)
- [Extending a Linux File System \(p. 673\)](#)
- [Deleting the Old Volume \(p. 675\)](#)

If you need to expand the storage space of a volume on a Windows instance, see [Expanding the Storage Space of a Volume](#) in the *Amazon EC2 User Guide for Windows Instances*.

If you create a larger volume, you will be charged for the additional storage. For more information, see the *Amazon Elastic Block Store* section on the [Amazon EC2 Pricing](#) page.

Note

If your storage needs demand a larger EBS volume than AWS provides, you may want to use RAID 0 to "stripe" a single logical volume across multiple physical volumes. For more information see [RAID Configuration on Linux](#).

Migrating Your Data to a Larger Volume

You must stop your instance to expand the storage space. When you stop and start an instance, be aware of the following:

- Any data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, back it up to persistent storage.
- If your instance is running in a VPC and has a public IP address, we release the address and give it a new public IP address. The instance retains its private IP addresses and any Elastic IP addresses.
- If your instance is running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. You must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can temporarily suspend the Auto Scaling processes for the group. For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling User Guide*.

To migrate your data to a larger volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then locate the instance with the volume that you want to expand.
3. Make a note of the instance ID and Availability Zone. You will specify this information when you attach a new volume to the instance later in this procedure.
4. Verify that the instance **Shutdown Behavior** is set to **Stop** and not **Terminate**.
 - a. Choose the instance.
 - b. From the context-menu (right-click) choose **Instance Settings**, and then choose **Change Shutdown Behavior**.
 - c. If the **Shutdown behavior** is set to **Terminate**, choose **Stop**, and then choose **Apply**.

If the **Shutdown behavior** is already set to **Stop**, then choose **Cancel**.
5. Stop the instance. For more information about how to stop an instance, see [Stopping and Starting Your Instances](#) (p. 300).
Warning
When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
6. Create a snapshot of the volume to expand.
 - a. In the navigation pane, choose **Volumes**, and then locate the volume you want to expand.
 - b. From the context-menu (right-click) choose the volume that you want to expand, and then choose **Create Snapshot**.
 - c. Enter information in the **Name** and **Description** fields, and then choose **Yes, Create**.
7. Create a new volume from the snapshot.
 - a. In the navigation pane, choose **Snapshots**.
 - b. When the status of the snapshot that you just created is set to **completed**, choose the snapshot, and then from the context-menu (right-click) choose **Create Volume**.
 - c. In the **Create Volume** dialog box, choose the desired volume type and enter the new volume size. You must also set the **Availability Zone** to match the instance Availability Zone. Choose **Yes, Create**.

Important

If you do not set the **Availability Zone** to match the instance then you will not be able to attach the new volume to the instance.

8. Detach the old volume.
 - a. In the navigation pane, choose **Volumes**, and then choose the old volume from the list. Make a note of the device name in the **Attachment Information** field. You will specify this information when you attach a new volume to the instance later in this procedure. The information appears in the following format:

```
i-xxxxxxxxxxxxxxxxxxxxx (instance_name):device_name
```
 - b. From the context-menu (right-click) choose the old volume, and then choose **Detach Volume**.
 - c. In the **Detach Volume** dialog box, choose **Yes, Detach**. It may take several minutes for the volume to detach.
9. Attach the newly expanded volume
 - a. In the navigation pane, choose **Volumes**.
 - b. From the context-menu (right-click) choose the new volume, and then choose **Attach Volume**.
 - c. Start typing the name or ID of the instance in the **Instance** field, and then choose the instance.
 - d. Enter the same device name retrieved in [Step 8.a \(p. 673\)](#), and then choose **Yes, Attach**. It is important to attach the new volume to the exact location you noted above (for example `/dev/sda1`).
10. Restart the instance.
 - a. In the navigation pane, choose **Instances** and then choose the instance you want to restart.
 - b. From the context-menu (right-click) choose **Instance State**, and then choose **Start**.
 - c. In the **Start Instances** dialog box, choose **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume, for example `/dev/sda1`.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space. On Linux, use the **df -h** command to check the file system size.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.9G  943M  6.9G  12% /
tmpfs           1.9G   0    1.9G   0% /dev/shm
```

If the size does not reflect your newly-expanded volume, you must extend the file system of your device so that your instance can use the new space. For more information, see [Extending a Linux File System \(p. 673\)](#).

Extending a Linux File System

In Linux, you use a file system-specific command to resize the file system to the larger size of the new volume. This command works even if the volume you wish to extend is the root volume. For `ext2`, `ext3`, and `ext4` file systems, this command is **resize2fs**. For XFS file systems, this command is **xfs_growfs**. For other file systems, refer to the specific documentation for those file systems for instructions on extending them.

If you are unsure of which file system you are using, you can use the **file -s** command to list the file system data for a device. The following example shows a Linux `ext4` file system and an SGI XFS file system.

```
[ec2-user ~]$ sudo file -s /dev/xvd*  
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...  
/dev/xvdf: SGI XFS filesystem data ...
```

Note

If the volume you are extending has been partitioned, you need to increase the size of the partition before you can resize the file system. For more information, see [Expanding a Linux Partition \(p. 676\)](#).

To check if your volume partition needs resizing

- Use the **lsblk** command to list the block devices attached to your instance. The example below shows three volumes: `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdf`.

```
[ec2-user ~]$ lsblk  
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
xvda        202:0    0 30G  0 disk  
##xvda1    202:1    0 30G  0 part /  
xvdb        202:16   0 30G  0 disk /mnt  
xvdf        202:80   0 35G  0 disk  
##xvdf1    202:81   0  8G  0 part
```

The root volume, `/dev/xvda1`, is a partition on `/dev/xvda`. Notice that they are both 30 GiB in size. In this case, the partition occupies all of the room on the device, so it does not need resizing.

The volume `/dev/xvdb` is not partitioned at all, so it does not need resizing.

However, `/dev/xvdf1` is an 8 GiB partition on a 35 GiB device and there are no other partitions on the volume. In this case, the partition must be resized in order to use the remaining space on the volume. For more information, see [Expanding a Linux Partition \(p. 676\)](#). After you resize the partition, you can follow the next procedure to extend the file system to occupy all of the space on the partition.

To extend a Linux file system

- Log in to your Linux instance using an SSH client. For more information about connecting to a Linux instance, see [Connecting to Your Linux Instance Using SSH \(p. 289\)](#).
- Use the **df -h** command to report the existing file system disk space usage. In this example, the `/dev/xvda1` device has already been expanded to 70 GiB, but the `ext4` file system only sees the original 8 GiB size, and the `/dev/xvdf` device has been expanded to 100 GiB, but the XFS file system only sees the original 1 GiB size.

```
[ec2-user ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/xvda1      7.9G  943M  6.9G  12% /  
tmpfs           1.9G   0    1.9G   0% /dev/shm  
/dev/xvdf       1014M  33M  982M   4% /mnt
```

- Use the file system-specific command to resize the file system to the new size of the volume. For a Linux `ext2`, `ext3`, or `ext4` file system, use the following command, substituting the device name that you want to extend.

Amazon Elastic Compute Cloud User Guide for Linux Instances EBS Volumes

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
resize2fs 1.42.3 (14-May-2012)
Filesystem at /dev/xvda1 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 5
Performing an on-line resize of /dev/xvda1 to 18350080 (4k) blocks.
The filesystem on /dev/xvda1 is now 18350080 blocks long.
```

For an XFS file system, first install the XFS userspace tools:

```
[ec2-user ~]$ sudo yum install xfsprogs
```

Then use the following command, substituting the mount point of the file system (XFS file systems must be mounted to resize them).

```
[ec2-user ~]$ sudo xfs_growfs -d /mnt
meta-data=/dev/xvdf          isize=256    agcount=4, agsize=65536 blks
=                               sectsz=512   attr=2
data      =                   bsize=4096  blocks=262144, imaxpct=25
=                               sunit=0     swidth=0 blks
naming    =version 2          bsize=4096  ascii-ci=0
log       =internal          bsize=4096  blocks=2560, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime =none               extsz=4096  blocks=0, rtextents=0
data blocks changed from 262144 to 26214400
```

Note

If you receive an `xfstl failed: Cannot allocate memory` error, you may need to update the Linux kernel on your instance. For more information, refer to your specific operating system documentation.

If you receive an `The filesystem is already nnnnnnn blocks long. Nothing to do!` error, see [Expanding a Linux Partition \(p. 676\)](#).

4. Use the `df -h` command to report the existing file system disk space usage, which should now show the full 70 GiB on the `ext4` file system and 100 GiB on the XFS file system.

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      69G   951M   68G   2% /
tmpfs           1.9G     0   1.9G   0% /dev/shm
/dev/xvdf       100G    45M  100G   1% /mnt
```

Deleting the Old Volume

After the new volume has been attached and extended in the instance, you can delete the old volume if it is no longer needed.

To delete the old volume

1. In the Amazon EC2 console, choose **Volumes** in the navigation pane and then choose the volume you want to delete.
2. From the context-menu (right-click) choose **Delete Volume**.
3. In the **Delete Volume** dialog box, choose **Yes, Delete**.

Expanding a Linux Partition

Some Amazon EC2 root volumes and volumes that are restored from snapshots contain a partition that actually holds the file system and the data. If you think of a volume as a container, a partition is another container inside the volume, and the data resides on the partition. Growing the volume size does not grow the partition; to take advantage of a larger volume, the partition must be expanded to the new size.

Note

Not all volumes restored from snapshots are partitioned, and this procedure may not apply to your volume. You may just need to resize the file system on your volume to make all of the space available. If you are not sure if your volume has a partition that needs resizing, see [To check if your volume partition needs resizing \(p. 674\)](#) for more information.

If the partition you want to expand is not the root partition, then you can simply unmount it and resize the partition from the instance itself. If the partition you need to resize is the root partition for an instance, the process becomes more complicated because you cannot unmount the root partition of a running instance. You have to perform the following procedures on another instance, which is referred to as a *secondary instance*.

Important

The following procedures were written for and tested on Amazon Linux. Other distributions with different tool sets and tool versions may behave differently.

Topics

- [Preparing a Linux Root Partition for Expansion \(p. 676\)](#)
- [Expanding a Linux Partition Using parted \(p. 677\)](#)
- [Expanding a Linux Partition Using gdisk \(p. 681\)](#)
- [Returning an Expanded Partition to its Original Instance \(p. 685\)](#)

Preparing a Linux Root Partition for Expansion

There are several steps that you need to take to expand the root partition of an instance. If the partition you need to expand is not the root partition, then this procedure is not necessary.

To prepare a Linux root partition for expansion

1. If your primary instance is running, stop it. You cannot perform the rest of this procedure on a running instance. For more information, see [Stop and Start Your Instance \(p. 299\)](#).
2. Check the integrity of your volume. File-system corruption that is picked up by the snapshot may render a restored root volume unbootable.
3. Take a snapshot of your volume. It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 686\)](#).
4. Record the device name that the volume is attached to. You can find this information on the **Root device** field of the instance's details pane. The value is likely `/dev/sda1` or `/dev/xvda`.
5. Detach the volume from the primary instance. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 669\)](#).
6. Attach the volume to another (secondary) instance in the same Availability Zone. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#). If your EBS volume is encrypted, you must use a secondary instance that supports Amazon EBS encryption; otherwise, you can use a `t2.micro` instance for this procedure. For more information, see [Supported Instance Types \(p. 697\)](#). If you do not already have a secondary instance, you will need to launch one. For more information, see [Launching an Instance \(p. 279\)](#).

Important

The secondary instance must be running when you attach the volume, and you should not reboot the secondary instance while multiple root volumes are attached; booting an instance with multiple root volumes attached could cause the instance to boot to the wrong volume.

7. Log in to the secondary instance with SSH. For more information, see [Connect to Your Linux Instance \(p. 289\)](#). Continue with the next procedure.

Expanding a Linux Partition Using parted

The **parted** utility is a partition editing tool that is available on most Linux distributions. It can create and edit both MBR partition tables and GPT partition tables. Some versions of **parted** (newer than version 2.1) have limited support for GPT partition tables and they may cause boot issues if their version of **parted** is used to modify boot volumes. You can check your version of **parted** with the `parted --version` command.

If you are expanding a partition that resides on a GPT partitioned device, you should choose to use the **gdisk** utility instead. If you're not sure which disk label type your volume uses, you can check it with the `sudo fdisk -l` command. For more information, see [To expand a Linux partition using gdisk \(p. 681\)](#).

To expand a Linux partition using parted

If the partition you need to expand is the root partition, be sure to follow the steps in [To prepare a Linux root partition for expansion \(p. 676\)](#) first.

1. Identify the device that contains the partition that you want to expand. Use the **lsblk** command to list all devices and partitions attached to the instance.

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf         202:80  0  100G  0 disk
##xvdf1     202:81  0    8G  0 part /mnt
xvda1       202:1   0   30G  0 disk /
```

In this example, the `xvdf` device has 100 GiB of available storage and it contains an 8 GiB partition.

2. Unmount the partition if it is mounted. Run the **umount** command with the value of `MOUNTPOINT` from the **lsblk** command. In this example, the `MOUNTPOINT` value for the partition is `/mnt`.

```
[ec2-user ~]$ sudo umount /mnt
```

3. Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 686\)](#).
4. Run the **parted** command on the device (and not the partition on the device). Remember to add the `/dev/` prefix to the name that **lsblk** outputs.

```
[ec2-user ~]$ sudo parted /dev/xvdf
GNU Parted 2.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

5. Change the **parted** units of measure to sectors.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

```
(parted) unit s
```

6. Run the **print** command to list the partitions on the device. For certain partition table types, you might be prompted to repair the partition table for the larger volume size. Answer 'Ignore' to any questions about fixing the existing partition table; you will create a new table later.

```
(parted) print
```

- a. If you receive the following message, enter 'Ignore' to prevent the backup GPT location from changing.

```
Error: The backup GPT table is not at the end of the disk, as it
should be. This might mean that another operating
system believes the disk is smaller. Fix, by moving the backup to the
end (and removing the old backup)?
Fix/Ignore/Cancel? Ignore
```

- b. If you receive the following message, enter 'Ignore' again to keep the space on the drive the same.

```
Warning: Not all of the space available to /dev/xvdf appears to be
used, you can fix the GPT to use all of the
space (an extra 46137344 blocks) or continue with the current setting?
Fix/Ignore? Ignore
```

7. Examine the output for the total size of the disk, the partition table type, the number of the partition, the start point of the partition, and any flags, such as `boot`. For `gpt` partition tables, note the name of the partition; for `msdos` partition tables, note the `Type` field (`primary` or `extended`). These values are used in the upcoming steps.

The following is a `gpt` partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

Number	Start	End	Size	File system	Name
	Flags				
128	2048s	4095s	2048s		BIOS Boot Partition
	bios_grub				
1	4096s	16777182s	16773087s	ext4	Linux

The following is an `msdos` partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
```

Number	Start	End	Size	Type	File system	Flags
1	2048s	35649535s	35647488s	primary	ext3	

8. Delete the partition entry for the partition using the number (1) from the previous step.

```
(parted) rm 1
```

9. Create a new partition that extends to the end of the volume.

(For the `gpt` partition table example) Note the start point and name of partition 1 above. For the `gpt` example, there is a start point of 4096s, and the name `Linux`. Run the `mkpart` command with the start point of partition 1, the name, and 100% to use all of the available space.

```
(parted) mkpart Linux 4096s 100%
```

(For the `msdos` partition table example) Note the start point and the partition type of partition 1 above. For the `msdos` example, there is a start point of 2048s and a partition type of `primary`. Run the `mkpart` command with a primary partition type, the start point of partition 1, and 100% to use all of the available space.

```
(parted) mkpart primary 2048s 100%
```

10. Run the `print` command again to verify your partition.

(For the `gpt` partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End       Size      File system  Name
Flags
128     2048s  4095s    2048s                    BIOS Boot Partition
 bios_grub
1       4096s  209713151s 209709056s ext4         Linux
```

(For the `msdos` partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End       Size      Type      File system  Flags
1       2048s  104857599s 104855552s primary  ext3
```

11. Check to see that any flags that were present earlier are still present for the partition that you expanded. In some cases the `boot` flag may be lost. If a flag was dropped from the partition when it was expanded, add the flag with the following command, substituting your partition number and the flag name. For example, the following command adds the `boot` flag to partition 1.

```
(parted) set 1 boot on
```

You can run the `print` command again to verify your change.

12. Run the `quit` command to exit `parted`.

```
(parted) quit
```

Note

Because you removed a partition and added a partition, **parted** may warn that you may need to update `/etc/fstab`. This is only required if the partition number changes.

13. Check the file system to make sure there are no errors (this is required before you may extend the file system). Note the file system type from the previous **print** commands. Choose one of the commands below based on your file system type; if you are using a different file system, consult the documentation for that file system to determine the correct check command.

(For `ext3` or `ext4` file systems)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(For `xfs` file systems)

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
          - zero log...
          - scan filesystem freespace and inode maps...
          - found root inode chunk
Phase 3 - for each AG...
          - scan and clear agi unlinked lists...
          - process known inodes and perform inode discovery...
          - agno = 0
          - agno = 1
          - agno = 2
          - agno = 3
          - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
          - setting up duplicate extent list...
          - check for inodes claiming duplicate blocks...
          - agno = 0
          - agno = 1
          - agno = 2
          - agno = 3
Phase 5 - rebuild AG headers and trees...
          - reset superblock...
Phase 6 - check inode connectivity...
          - resetting contents of realtime bitmap and summary inodes
          - traversing filesystem ...
          - traversal finished ...
          - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

- The next steps differ depending on whether the expanded partition belongs on the current instance or if it is the root partition for another instance.
 - If this partition belongs on the current instance, remount the partition at the `MOUNTPOINT` identified in [Step 2 \(p. 677\)](#).

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
```

After you have mounted the partition, extend the file system to use the newly available space by following the procedures in [Extending a Linux File System \(p. 673\)](#).

- If this volume is the root partition for another instance, proceed to the procedures in [Returning an Expanded Partition to its Original Instance \(p. 685\)](#).

Expanding a Linux Partition Using `gdisk`

The **`gdisk`** utility (sometimes called GPT fdisk) is a text-based, menu-driven tool for creating and editing partition tables, and it has better support for GPT partition tables than **`parted`** in some distributions. Many common Linux distributions (such as Amazon Linux and Ubuntu) provide **`gdisk`** by default. If your distribution does not provide the **`gdisk`** command, you can find out how to get it by visiting [Obtaining GPT fdisk](#); in many cases, it is much easier to launch an Amazon Linux instance to use as a secondary instance because the **`gdisk`** command is already available.

To expand a Linux partition using `gdisk`

If the partition you need to expand is the root partition, be sure to follow the steps in [To prepare a Linux root partition for expansion \(p. 676\)](#) first.

- Identify the device that contains the partition that you want to expand. Use the **`lsblk`** command to list all devices and partitions attached to the instance.

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO MOUNTPOINT
xvdf        202:80  0  100G  0
##xvdf1     202:81  0   9.9G  0 /mnt
xvda1       202:1   0   30G  0 /
```

In this example, the `xvdf` device has 100 GiB of available storage and it contains an 9.9 GiB partition.

- Unmount the partition if it is mounted. Run the **`umount`** command with the value of `MOUNTPOINT` from the **`lsblk`** command. In this example, the `MOUNTPOINT` value for the partition is `/mnt`.

```
[ec2-user ~]$ sudo umount /mnt
```

- Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 686\)](#).
- Run the **`gdisk`** command on the device (and not the partition on the device). Remember to add the `/dev/` prefix to the name that **`lsblk`** outputs.

```
[ec2-user ~]$ sudo gdisk /dev/xvdf
gdisk /dev/xvdf
GPT fdisk (gdisk) version 0.8.10
```


**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

```
Partition table scan:
 MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
```

5. Run the **p** command to print the partition table for the device.
6. Examine the output for the disk identifier, partition number, starting sector, code for the partition, and name of the partition. If your volume has multiple partitions, take note of each one.

```
Command (? for help): p
Disk /dev/xvdf: 209715200 sectors, 100.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 947F4655-F3BF-4A1F-8203-A7B30C2A4425
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20705246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2108 sectors (1.0 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1             2048             20705152     9.9 GiB   EF00  lxroot
```

In the above example the disk identifier is 947F4655-F3BF-4A1F-8203-A7B30C2A4425, the partition number is 1, the starting sector is 2048, the code is EF00, and the name is lxroot.

7. Because the existing partition table was originally created for a smaller volume, you need to create a new partition table for the larger volume. Run the **o** command to create a new, empty partition table.

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

8. Use the **n** command to create a new partition entry for each partition on the device.
 - If your volume has only one partition, at each prompt, enter the values that you recorded earlier. For the last sector value, use the default value to expand to the entire volume size.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-209715166, default = 209715166) or
 {+-}size{KMGTP}: 209715166
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): EF00
Changed type of partition to 'EFI System'
```

- If your volume has more than one partition, there is likely a BIOS boot partition, and a main data partition. Create a new partition entry for the BIOS boot partition using the values that you recorded earlier. Create another new partition entry for the main data partition using the values that you recorded earlier, but for the last sector value, use the default value to expand to the entire volume size.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

```
Last sector (2048-209715166, default = 209715166) or
{+-}size{KMGTP}: 4095
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): EF02
Changed type of partition to 'BIOS boot partition'

Command (? for help): n
Partition number (2-128, default 2): 2
First sector (34-209715166, default = 4096) or {+-}size{KMGTP}: 4096
Last sector (4096-209715166, default = 209715166) or
{+-}size{KMGTP}: 209715166
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 0700
Changed type of partition to 'Microsoft basic data'
```

9. Use the **c** command to change the name of each partition to the name of the previous partition. If your partition did not have a name, simply type **Enter**.

```
Command (? for help): c
Using 1
Enter name: lxroot
```

10. Use the **x** command to enter the expert command menu.
11. Use the **g** command to change the disk identifier to the original value.

```
Expert command (? for help): g
Enter the disk's unique GUID ('R' to randomize): 947F4655-F3BF-4A1F-8203-
A7B30C2A4425
The new disk GUID is 947F4655-F3BF-4A1F-8203-A7B30C2A4425
```

12. Use the **w** command to write the changes to the device and exit.

```
Expert command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE
EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
```

13. Check the file system to make sure there are no errors (this is required before you may extend the file system).
- a. Find the file system type with the following command, substituting the partition you just expanded (this may be `/dev/xvdf2` if your volume had multiple partitions).

```
[ec2-user ~]$ sudo file -sL /dev/xvdf1
```

- b. Choose one of the commands below based on your file system type; if you are using a different file system, consult the documentation for that file system to determine the correct check command.

(For ext3 or ext4 file systems)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Volumes**

```
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(For xfs file systems)

Note

You may need to install the `xfsprogs` package to work with XFS file systems. Use the following command to add XFS support to your Amazon Linux instance.

```
[ec2-user ~]$ sudo yum install -y xfsprogs
```

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
          - zero log...
          - scan filesystem freespace and inode maps...
          - found root inode chunk
Phase 3 - for each AG...
          - scan and clear agi unlinked lists...
          - process known inodes and perform inode discovery...
          - agno = 0
          - agno = 1
          - agno = 2
          - agno = 3
          - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
          - setting up duplicate extent list...
          - check for inodes claiming duplicate blocks...
          - agno = 0
          - agno = 1
          - agno = 2
          - agno = 3
Phase 5 - rebuild AG headers and trees...
          - reset superblock...
Phase 6 - check inode connectivity...
          - resetting contents of realtime bitmap and summary inodes
          - traversing filesystem ...
          - traversal finished ...
          - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. The next steps differ depending on whether the expanded partition belongs on the current instance or if it is the root partition for another instance.
- If this partition belongs on the current instance, remount the partition at the `MOUNTPOINT` identified in [Step 2 \(p. 681\)](#).

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
```

After you have mounted the partition, extend the file system to use the newly available space by following the procedures in [Extending a Linux File System \(p. 673\)](#).

- If this volume is the root partition for another instance, proceed to the procedures in [Returning an Expanded Partition to its Original Instance](#) (p. 685).

Returning an Expanded Partition to its Original Instance

If you expanded a root partition from another instance, follow this procedure to return the volume to its original instance.

To return an expanded root partition to its original instance

1. Detach the expanded partition from its secondary instance. For more information, see [Detaching an Amazon EBS Volume from an Instance](#) (p. 669).
2. Reattach the volume to the primary instance using the device name that you identified in [Step 4](#) (p. 676) of the [preparation procedure](#) (p. 676). For more information, see [Attaching an Amazon EBS Volume to an Instance](#) (p. 654).
3. Start the primary instance. For more information, see [Stop and Start Your Instance](#) (p. 299).
4. (Optional) If you launched a secondary instance for the sole purpose of expanding the partition, you can terminate the instance to stop incurring charges. For more information, see [Terminate Your Instance](#) (p. 305).
5. Connect to your primary instance and extend the file system to use the newly available space by following the procedures in [Extending a Linux File System](#) (p. 673).

After you are finished with this expanding the file system, you can create an AMI from the instance that you can use to launch new instances with the desired partition size. For more information, see [Amazon Machine Images \(AMI\)](#) (p. 69).

Amazon EBS Snapshots

You can back up the data on your EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs. When you delete a snapshot, only the data unique to that snapshot is removed. Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.

Contents

- [Snapshot Overview](#) (p. 685)
- [Creating an Amazon EBS Snapshot](#) (p. 686)
- [Deleting an Amazon EBS Snapshot](#) (p. 687)
- [Copying an Amazon EBS Snapshot](#) (p. 688)
- [Viewing Amazon EBS Snapshot Information](#) (p. 690)
- [Sharing an Amazon EBS Snapshot](#) (p. 691)

Snapshot Overview

When you create an EBS volume, you can create it based on an existing snapshot. The new volume begins as an exact replica of the original volume that was used to create the snapshot. When you create a volume from an existing snapshot, it loads lazily in the background so that you can begin using them right away. If you access a piece of data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information, see [Creating an Amazon EBS Snapshot](#) (p. 686).

By modifying their access permissions, snapshots can be shared across AWS accounts. You can make copies of your own snapshots as well as snapshots that have been shared with you. For more information, see [Sharing an Amazon EBS Snapshot \(p. 691\)](#).

EBS snapshots broadly support EBS encryption:

- Snapshots of encrypted volumes are automatically encrypted.
- Volumes that are created from encrypted snapshots are automatically encrypted.
- When you copy an unencrypted snapshot that you own, you can encrypt it during the copy process.
- When you copy an encrypted snapshot that you own, you can reencrypt it with a different key during the copy process.

For more information, see [Amazon EBS Encryption](#).

Snapshots are constrained to the region in which they are created. After you have created a snapshot of an EBS volume, you can use it to create new volumes in the same region. For more information, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 652\)](#). You can also copy snapshots across regions, making it easier to leverage multiple regions for geographical expansion, data center migration, and disaster recovery. You can copy any accessible snapshots that have a `completed` status. For more information, see [Copying an Amazon EBS Snapshot \(p. 688\)](#).

Creating an Amazon EBS Snapshot

After writing data to an EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is `pending` until the snapshot is complete (when all of the modified blocks have been transferred to Amazon S3), which can take several hours for large initial snapshots or subsequent snapshots where many blocks have changed. While it is completing, an in-progress snapshot is not affected by ongoing reads and writes to the volume.

Important

Although you can take a snapshot of a volume while a previous snapshot of that volume is in the `pending` status, having multiple `pending` snapshots of a volume may result in reduced volume performance until the snapshots complete.

There is a limit of 5 `pending` snapshots for a single volume. If you receive a `ConcurrentSnapshotLimitExceeded` error while trying to create multiple concurrent snapshots of the same volume, wait for one or more of the `pending` snapshots to complete before creating another snapshot of that volume.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. The data in your encrypted volumes and any associated snapshots is protected both at rest and in motion. For more information, see [Amazon EBS Encryption](#).

By default, only you can create volumes from snapshots that you own. However, you can share your unencrypted snapshots with specific AWS accounts, or you can share them with the entire AWS community by making them public. For more information, see [Sharing an Amazon EBS Snapshot \(p. 691\)](#).

You can share an encrypted snapshot only with specific AWS accounts. For others to use your shared, encrypted snapshot, you must also share the CMK key that was used to encrypt it. Users with access to your encrypted snapshot must create their own personal copy of it and then use that copy to restore

the volume. Your copy of a shared, encrypted snapshot can also be re-encrypted with a different key. For more information, see [Sharing an Amazon EBS Snapshot \(p. 691\)](#).

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is `pending`.

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

To unmount the volume in Linux, use the following command:

```
umount -d device_name
```

Where *device_name* is the device name (for example, `/dev/sdh`).

After you've created a snapshot, you can tag it to help you manage it later. For example, you can add tags describing the original volume from which the snapshot was created, or the device name that was used to attach the original volume to an instance. For more information, see [Tagging Your Amazon EC2 Resources \(p. 751\)](#).

To create a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Choose **Create Snapshot**.
4. In the **Create Snapshot** dialog box, select the volume to create a snapshot for, and then choose **Create**.

To create a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-snapshot` (AWS CLI)
- `New-EC2Snapshot` (AWS Tools for Windows PowerShell)

Deleting an Amazon EBS Snapshot

When you delete a snapshot, only the data exclusive to that snapshot is removed. Deleting previous snapshots of a volume does not affect your ability to restore volumes from later snapshots of that volume.

If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Note that you can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see [Deregistering Your AMI \(p. 140\)](#).

To delete a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Delete** from the **Actions** list.
4. Choose **Yes, Delete**.

To delete a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-snapshot](#) (AWS CLI)
- [Remove-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Copying an Amazon EBS Snapshot

With Amazon EBS, you can create point-in-time snapshots of volumes which we store for you in Amazon Simple Storage Service (Amazon S3). After you've created a snapshot and it has finished copying to Amazon S3 (when the snapshot status is `completed`), you can copy it from one AWS region to another, or within the same region. Amazon S3 server-side encryption (256-bit AES) protects a snapshot's data-in-transit during copying. The snapshot copy receives a snapshot ID different from the original snapshot's ID.

Note

To copy an Amazon Relational Database Service (Amazon RDS) snapshot, see [Copying a DB Snapshot](#) in the Amazon Relational Database Service User Guide.

You can use a copy of a snapshot in the following ways:

- Geographic expansion: Launch your applications in a new region.
- Migration: Move an application to a new region, to enable better availability and minimize cost.
- Disaster recovery: Back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary region. This minimizes data loss and recovery time.
- Encryption: Encrypt a previously unencrypted snapshot, change the key with which the snapshot is encrypted, or, for encrypted snapshots that have been shared with you, create a copy that you own in order to restore the volume from it.
- Data retention and auditing requirements: Copy your encrypted EBS snapshots from one AWS account to another to preserve data logs or other files for auditing or data retention. Using a different account helps prevent accidental snapshot deletions, and protects you if your main AWS account is compromised.

Note

Snapshots created by the CopySnapshot action have an arbitrary volume ID that should not be used for any purpose.

User-defined tags are not copied from the source snapshot to the new snapshot. After the copy operation is complete, you can apply user-defined tags to the new snapshot. For more information, see [Tagging Your Amazon EC2 Resources \(p. 751\)](#).

You can have up to five snapshot copy requests in progress to a single destination per account. You can copy any accessible snapshots that have a `completed` status, including shared snapshots and snapshots that you've created. You can also copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination region.

When you copy a snapshot, you are only charged for the data transfer and storage used to copy the snapshot data across regions and to store the copied snapshot in the destination region. You are not charged if the snapshot copy fails. However, if you cancel a snapshot copy that is not yet complete, or delete the source snapshot while the copy is in progress, you are charged for the bandwidth of the data transferred.

The first snapshot copy to another region is always a full copy. Each subsequent snapshot copy is incremental (which makes the copy process faster), meaning that only the blocks in the snapshot that have changed after your last snapshot copy to the same destination are transferred. Support for incremental snapshots is specific to a region pair where a previous complete snapshot copy of the source volume is already available in the destination region, and it is limited to the default EBS CMK for encrypted snapshots. For example, if you copy an unencrypted snapshot from the US East (N. Virginia) region to the US West (Oregon) region, the first snapshot copy of the volume is a full copy and subsequent snapshot copies of the same volume transferred between the same regions are incremental.

Note

Snapshot copies within a single region do not copy any data at all as long as the following conditions apply:

- The encryption status of the snapshot copy does not change during the copy operation
- For encrypted snapshots, both the source snapshot and the copy are encrypted with the default EBS CMK

If you would like another account to be able to copy your snapshot, you must either modify the snapshot permissions to allow access to that account or make the snapshot public so that all AWS accounts may copy it. For more information, see [Sharing an Amazon EBS Snapshot \(p. 691\)](#).

Encrypted Snapshots

When you copy a snapshot, you can choose to encrypt the copy (if the original snapshot was not encrypted) or you can specify a CMK different from the original one, and the resulting copied snapshot will use the new CMK. However, changing the encryption status of a snapshot or using a non-default EBS CMK during a copy operation always results in a full copy (not incremental), which may incur greater data transfer and storage charges.

To copy an encrypted snapshot from another account, you must have permissions to use the snapshot and you must have permissions to use the customer master key (CMK) that was used to encrypt the original snapshot. For more information, see [Sharing an Amazon EBS Snapshot \(p. 691\)](#).

Note

When copying an encrypted snapshot that was shared with you, you should consider re-encrypting the snapshot during the copy process with a different key that you control. This protects you if the original key is compromised, or if the owner revokes the key for any reason, which could cause you to lose access to the volume you created.

To copy a snapshot using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
 - **Destination region:** Select the region where you want to write the copy of the snapshot.
 - **Description:** By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.
 - **Encryption:** If the source snapshot is not encrypted, you can choose to encrypt the copy. You cannot decrypt an encrypted snapshot.

- **Master Key:** The customer master key (CMK) that will be used to encrypt this snapshot. You can select from master keys in your account or type/paste the ARN of a key from a different account. You can create a new master encryption key in the IAM console.
5. Choose **Copy**.
 6. In the **Copy Snapshot** confirmation dialog box, choose **Snapshots** to go to the **Snapshots** page in the region specified, or choose **Close**.

To view the progress of the copy process later, switch to the destination region, and then refresh the **Snapshots** page. Copies in progress are listed at the top of the page.

To check for failure

If you attempt to copy an encrypted snapshot without having permissions to use the encryption key, the operation will fail silently. The error state will not be displayed in the console until you refresh the page. You can also check the state of the snapshot from the command line. For example:

```
$ aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

If the copy failed because of insufficient key permissions, you will see the following message:

```
"StateMessage": "Given key ID is not accessible"
```

Note

When copying an encrypted snapshot, you must have describe permissions on the default CMK. Explicitly denying these permissions will result in copy failure.

To copy a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [copy-snapshot](#) (AWS CLI)
- [Copy-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Viewing Amazon EBS Snapshot Information

You can view detailed information about your snapshots.

To view snapshot information using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. To reduce the list, choose an option from the **Filter** list. For example, to view only your snapshots, choose **Owned By Me**. You can filter your snapshots further by using the advanced search options. Choose the search bar to view the filters available.
4. To view more information about a snapshot, choose it.

To view snapshot information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshots](#) (AWS CLI)
- [Get-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Sharing an Amazon EBS Snapshot

You can share your unencrypted snapshots with your co-workers or others in the AWS community by modifying the permissions of the snapshot. Users that you have authorized can quickly use your unencrypted shared snapshots as the basis for creating their own EBS volumes. If you choose, you can also make your unencrypted snapshots available publicly to all AWS users.

You can share an encrypted snapshot with specific AWS accounts, though you cannot make it public. For others to use the snapshot, you must also share the custom CMK key used to encrypt it. Cross-account permissions may be applied to a custom key either when it is created or at a later time. Users with access can copy your snapshot and create their own EBS volumes based on your snapshot while your original snapshot remains unaffected.

Important

When you share a snapshot (whether by sharing it with another AWS account or making it public to all), you are giving others access to all the data on your snapshot. Share snapshots only with people with whom you want to share *all* your snapshot data.

Several technical and policy restrictions apply to sharing snapshots:

- Snapshots are constrained to the region in which they were created. If you would like to share a snapshot with another region, you need to copy the snapshot to that region. For more information about copying snapshots, see [Copying an Amazon EBS Snapshot \(p. 688\)](#).
- If your snapshot uses the longer resource ID format, you can only share it with another account that also supports longer IDs. For more information, see [Resource IDs](#).
- AWS prevents you from sharing snapshots that were encrypted with your default CMK. Snapshots that you intend to share must instead be encrypted with a custom CMK. For information about creating keys, see [Creating Keys](#).
- Users of your shared CMK who will be accessing encrypted snapshots must be granted `DescribeKey` and `ReEncrypt` permissions. For information about managing and sharing CMK keys, see [Controlling Access to Customer Master Keys](#).
- If you have access to a shared encrypted snapshot and you wish to restore a volume from it, you must create a personal copy of the snapshot and then use that copy to restore the volume. We recommend that you re-encrypt the snapshot during the copy process with a different key that you control. This protects your access to the volume if the original key is compromised, or if the owner revokes the key for any reason.

To modify snapshot permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Modify Permissions** from the **Actions** list.
4. Choose whether to make the snapshot public or to share it with specific AWS accounts:
 - To make the snapshot public, choose **Public**.

This is not a valid option for encrypted snapshots or snapshots with AWS Marketplace product codes.

- To expose the snapshot to only specific AWS accounts, choose **Private**, enter the ID of the AWS account (without hyphens) in the **AWS Account Number** field, and choose **Add Permission**. Repeat until you've added all the required AWS accounts.

Important

If your snapshot is encrypted, you must ensure that the following are true:

- The snapshot is encrypted with a custom CMK, not your default CMK. If you attempt to change the permissions of a snapshot encrypted with your default CMK, the console will display an error message.
- You are sharing the custom CMK with the accounts that have access to your snapshot.

5. Choose **Save**.

To view and modify snapshot permissions using the command line

To view the `createVolumePermission` attribute of a snapshot, you can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `createVolumePermission` attribute of a snapshot, you can use one of the following commands.

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

Amazon EBS–Optimized Instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS, with options between 500 Mbps and 10,000 Mbps, depending on the instance type you use. When attached to an EBS–optimized instance, General Purpose SSD (`gp2`) volumes are designed to deliver within 10% of their baseline and burst performance 99% of the time in a given year, and Provisioned IOPS SSD (`io1`) volumes are designed to deliver within 10% of their provisioned performance 99.9% of the time in a given year. Both Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) guarantee performance consistency of 90% of burst throughput 99% of the time in a given year. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see [Amazon EBS Volume Types \(p. 638\)](#).

When you enable EBS optimization for an instance that is not EBS–optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see [EBS-optimized Instances](#) on the Amazon EC2 Pricing page.

Contents

- [Instance Types that Support EBS Optimization \(p. 692\)](#)
- [Enabling EBS Optimization at Launch \(p. 694\)](#)
- [Modifying EBS Optimization for a Running Instance \(p. 695\)](#)

Instance Types that Support EBS Optimization

The following table shows which instance types support EBS optimization, the dedicated bandwidth to Amazon EBS, the maximum number of IOPS the instance can support if you are using a 16 KiB I/O size, and the typical maximum aggregate throughput that can be achieved on that connection in MiB/s with a streaming read workload and 128 KiB I/O size. Choose an EBS–optimized instance

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Optimization**

that provides more dedicated EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

Note that some instance types are EBS-optimized by default. For instances that are EBS-optimized by default, there is no need to enable EBS optimization and there is no effect if you disable EBS optimization using the CLI or API. You can enable EBS optimization for the other instance types that support EBS optimization when you launch the instances, or enable EBS optimization after the instances are running.

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000	250	16,000
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000
d2.8xlarge	Yes	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000	250	16,000
m1.large		500	62.5	4,000
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	Yes	450	56.25	3,600
m4.xlarge	Yes	750	93.75	6,000
m4.2xlarge	Yes	1,000	125	8,000

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Optimization**

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
m4.4xlarge	Yes	2,000	250	16,000
m4.10xlarge	Yes	4,000	500	32,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000
r3.4xlarge		2,000	250	16,000
x1.32xlarge	Yes	10,000	1,250	65,000

* These instance types must be launched as EBS-optimized to consistently achieve this level of performance.

** This value is a rounded approximation based on a 100% read-only workload and it is provided as a baseline configuration aid. EBS-optimized connections are full-duplex, and can drive more throughput and IOPS in a 50/50 read/write workload where both communication lanes are used. In some cases, network, file system, and Amazon EBS encryption overhead can reduce the maximum throughput and IOPS available.

Note that some instances with 10-gigabit network interfaces, such as `i2.8xlarge` and `r3.8xlarge` do not offer EBS-optimization, and therefore do not have dedicated EBS bandwidth available and are not listed here. On these instances, network traffic and Amazon EBS traffic is shared on the same 10-gigabit network interface. Some other 10-gigabit network instances, such as `c4.8xlarge` and `d2.8xlarge` offer dedicated EBS bandwidth in addition to a 10-gigabit interface which is used exclusively for network traffic.

Enabling EBS Optimization at Launch

You can enable EBS optimization for an instance by setting its `EBS-optimized` attribute.

To enable EBS optimization when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Launch Instance**. In **Step 1: Choose an Amazon Machine Image (AMI)**, select an AMI.
3. In **Step 2: Choose an Instance Type**, select an instance type that is listed as supporting EBS optimization.
4. In **Step 3: Configure Instance Details**, complete the fields that you need and select **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support EBS optimization, this option is not present. If the instance type that you selected is EBS-optimized by default, this option is selected and you can't deselect it.
5. Follow the directions to complete the wizard and launch your instance.

To enable EBS optimization when launching an instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--ebs-optimized` with [run-instances](#) (AWS CLI)
- `-EbsOptimized` with [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Modifying EBS Optimization for a Running Instance

You can enable or disable EBS optimization for a running instance by modifying its EBS-optimized instance attribute.

To enable EBS optimization for a running instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **Change Instance Type**.
6. In the **Change Instance Type** dialog box, do one of the following:
 - If the instance type of your instance is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it. You can click **Cancel**, because EBS optimization is already enabled for the instance.
 - If the instance type of your instance supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
 - If the instance type of your instance does not support EBS optimization, **EBS-optimized** is deselected and you can't select it. You can select an instance type from **Instance Type** that supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
7. Click **Actions**, select **Instance State**, and then click **Start**.

To enable EBS optimization for a running instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--ebs-optimized` with [modify-instance-attribute](#) (AWS CLI)
- `-EbsOptimized` with [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Amazon EBS Encryption

Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume

The encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and any snapshots created from them. The first time you

create an encrypted volume in a region, a default CMK is created for you automatically. This key is used for Amazon EBS encryption unless you select a CMK that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, and disable keys to define access controls, and to audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

This feature is supported with all EBS volume types (General Purpose SSD [gp2], Provisioned IOPS SSD [io1], Throughput Optimized HDD [st1], Cold HDD [sc1], and Magnetic [standard]), and you can expect the same IOPS performance on encrypted volumes as you would with unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access unencrypted volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts if you take the following steps:

1. Use a custom CMK, not your default CMK, to encrypt your volume.
2. Give the specific accounts access to the custom CMK.
3. Create the snapshot.
4. Give the specific accounts access to the snapshot.

For more information, see [Sharing an Amazon EBS Snapshot](#).

Amazon EBS encryption is only available on certain instance types. You can attach both encrypted and unencrypted volumes to a supported instance type. For more information, see [Supported Instance Types](#) (p. 697).

Contents

- [Encryption Key Management](#) (p. 696)
- [Supported Instance Types](#) (p. 697)
- [Changing the Encryption State of Your Data](#) (p. 697)

Encryption Key Management

Amazon EBS encryption handles key management for you. Each newly-created volume is encrypted with a unique 256-bit key; any snapshots of this volume and any subsequent volumes created from those snapshots also share that key. These keys are protected by our own key management infrastructure, which implements strong logical and physical security controls to prevent unauthorized access. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm.

You cannot change the CMK that is associated with an existing snapshot or encrypted volume. However, you can associate a different CMK during a snapshot copy operation (including encrypting a copy of an unencrypted snapshot) and the resulting copied snapshot will use the new CMK.

Amazon's overall key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms and is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

Each AWS account has a unique master key that is stored completely separate from your data, on a system that is surrounded with strong physical and logical security controls. Each encrypted volume (and its subsequent snapshots) is encrypted with a unique volume encryption key that is then encrypted with a region-specific secure master key. The volume encryption keys are used in memory on the server that hosts your EC2 instance; they are never stored on disk in plain text.

Supported Instance Types

Amazon EBS encryption is available on the instance types listed in the table below. These instance types leverage the Intel AES New Instructions (AES-NI) instruction set to provide faster and simpler data protection. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

Instance family	Instance types that support Amazon EBS encryption
General purpose	m3.medium m3.large m3.xlarge m3.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge t2.nano t2.micro t2.small t2.medium t2.large
Compute optimized	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
Memory optimized	cr1.8xlarge r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge x1.32xlarge
Storage optimized	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
GPU instances	g2.2xlarge g2.8xlarge

For more information about these instance types, see [Instance Type Details](#).

Changing the Encryption State of Your Data

There is no direct way to encrypt an existing unencrypted volume, or to remove encryption from an encrypted volume. However, you can migrate data between encrypted and unencrypted volumes. You can also apply a new encryption status while copying a snapshot:

- While copying an unencrypted snapshot of an unencrypted volume, you can encrypt the copy. Volumes restored from this encrypted copy will also be encrypted.
- While copying an encrypted snapshot of an encrypted volume, you can re-encrypt the copy using a different CMK. Volumes restored from the encrypted copy will only be accessible using the newly applied CMK.

Migrate Data between Encrypted and Unencrypted Volumes

When you have access to both an encrypted and unencrypted volume, you can freely transfer data between them. EC2 carries out the encryption or decryption operations transparently.

To migrate data between encrypted and unencrypted volumes

1. Create your destination volume (encrypted or unencrypted, depending on your need) by following the procedures in [Creating an Amazon EBS Volume \(p. 650\)](#).
2. Attach the destination volume to the instance that hosts the data to migrate. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).
3. Make the destination volume available by following the procedures in [Making an Amazon EBS Volume Available for Use \(p. 655\)](#). For Linux instances, you can create a mount point at `/mnt/destination` and mount the destination volume there.
4. Copy the data from your source directory to the destination volume. It may be most convenient to use a bulk-copy utility for this.

Linux

Use the **rsync** command as follows to copy the data from your source to the destination volume. In this example, the source data is located in `/mnt/source` and the destination volume is mounted at `/mnt/destination`.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Windows

At a command prompt, use the **robocopy** command to copy the data from your source to the destination volume. In this example, the source data is located in `D:\` and the destination volume is mounted at `E:\`.

```
PS C:\Users\Administrator> robocopy D:\ E:\ /e /copyall /eta
```

Apply Encryption While Copying a Snapshot

Because you can apply encryption to a snapshot while copying it, another path to encrypting your data is the following procedure.

To encrypt a volume's data by means of snapshot copying

1. Create a snapshot of your unencrypted EBS volume. This snapshot is also unencrypted.
2. Copy the snapshot while applying encryption parameters. The resulting target snapshot is encrypted.
3. Restore the encrypted snapshot to a new volume, which is also encrypted.

For more information, see [Copying an Amazon EBS Snapshot](#).

Re-Encrypt a Snapshot with a New CMK

The ability to encrypt a snapshot during copying also allows you to re-encrypt an already-encrypted snapshot that you own. In this operation, the plaintext of your snapshot will be encrypted using a new CMK that you provide. Volumes restored from the resulting copy will only be accessible using the new CMK.

In a related scenario, you may choose to re-encrypt a snapshot that has been shared with you. Before you can restore a volume from a shared encrypted snapshot, you must create your own copy of it. By default, the copy will be encrypted with the key shared by the snapshot's owner. However, we recommend that you re-encrypt the snapshot during the copy process with a different key that you control. This protects your access to the volume if the original key is compromised, or if the owner revokes the key for any reason.

The following procedure demonstrates how to re-encrypt a snapshot that you own.

To re-encrypt a snapshot using the console

1. Create a custom CMK. For more information, see [AWS Key Management Service Developer Guide](#).
2. Create an EBS volume encrypted with (for this example) your default CMK.
3. Create a snapshot of your encrypted EBS volume. This snapshot is also encrypted with your default CMK.
4. On the **Snapshots** page, choose **Actions**, then choose **Copy**.

5. In the **Copy Snapshot** window, supply the complete ARN for your custom CMK (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Click **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—will be encrypted with your custom CMK.

The following procedure demonstrates how to re-encrypt a shared encrypted snapshot as you copy it. For this to work, you need access permissions to both the shared encrypted snapshot and to the CMK that encrypted it.

To copy and re-encrypt a shared snapshot using the console

1. Choose the shared encrypted snapshot on the **Snapshots** page, choose **Actions**, then choose **Copy**.
2. In the **Copy Snapshot** window, supply the complete ARN for a CMK that you own (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Click **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—will be encrypted with the CMK that you supplied. Changes to the original shared snapshot, its encryption status, or the shared CMK will have no effect on your copy.

For more information, see [Copying an Amazon EBS Snapshot](#).

Amazon EBS Volume Performance on Linux Instances

Several factors, including I/O characteristics and the configuration of your instances and volumes, can affect the performance of Amazon EBS. Customers who follow the guidance on our Amazon EBS and Amazon EC2 product detail pages typically achieve good performance out of the box. However, there are some cases where you may need to do some tuning in order to achieve peak performance on the platform. This topic discusses general best practices as well as performance tuning that is specific to certain use cases. We recommend that you tune performance with information from your actual workload, in addition to benchmarking, to determine your optimal configuration. After you learn the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and at your options for increasing Amazon EBS performance to meet those requirements.

For more information, see [Amazon EBS Volume Performance on Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Amazon EBS Performance Tips](#) (p. 699)
- [Amazon EC2 Instance Configuration](#) (p. 701)
- [I/O Characteristics](#) (p. 704)
- [Initializing Amazon EBS Volumes](#) (p. 706)
- [RAID Configuration on Linux](#) (p. 707)
- [Benchmark EBS Volumes](#) (p. 711)

Amazon EBS Performance Tips

These tips represent best practices for getting optimal performance from your EBS volumes in a variety of user scenarios.

Use EBS-Optimized Instances

On instances without support for EBS-optimized throughput, network traffic can contend with traffic between your instance and your EBS volumes; on EBS-optimized instances, the two types of traffic are kept separate. Some EBS-optimized instance configurations incur an extra cost (such as C3, R3, and M3), while others are always EBS-optimized at no extra cost (such as M4, C4, and D2). For more information, see [Amazon EC2 Instance Configuration \(p. 701\)](#).

Understand How Performance is Calculated

When you measure the performance of your EBS volumes, it is important to understand the units of measure involved and how performance is calculated. For more information, see [I/O Characteristics \(p. 704\)](#).

Understand Your Workload

There is a relationship between the maximum performance of your EBS volumes, the size and number of I/O operations, and the time it takes for each action to complete. Each of these factors (performance, I/O, and latency) affects the others, and different applications are more sensitive to one factor or another. For more information, see [Benchmark EBS Volumes \(p. 711\)](#).

Be Aware of the Performance Penalty When Initializing Volumes from Snapshots

There is a significant increase in latency when you first access each block of data on a new EBS volume that was restored from a snapshot. You can avoid this performance hit by accessing each block prior to putting the volume into production. This process is called *initialization* (formerly known as pre-warming). For more information, see [Initializing Amazon EBS Volumes \(p. 706\)](#).

Factors That Can Degrade HDD Performance

When you create a snapshot of a Throughput Optimized HDD (`st1`) or Cold HDD (`sc1`) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress. This behavior is specific to these volume types. Other factors that can limit performance include driving more throughput than the instance can support, the performance penalty encountered while initializing volumes restored from a snapshot, and excessive amounts of small, random I/O on the volume. For more information about calculating throughput for HDD volumes, see [Amazon EBS Volume Types](#).

Your performance can also be impacted if your application isn't sending enough I/O requests. This can be monitored by looking at your volume's queue length and I/O size. The queue length is the number of pending I/O requests from your application to your volume. For maximum consistency, HDD-backed volumes must maintain a queue length (rounded to the nearest whole number) of 4 or more when performing 1 MiB sequential I/O. For more information about ensuring consistent performance of your volumes, see [I/O Characteristics \(p. 704\)](#).

Increase Read-Ahead for High-Throughput, Read-Heavy Workloads on `st1` and `sc1`

Some workloads are read-heavy and access the block device through the operating system page cache (for example, from a file system). In this case, to achieve the maximum throughput, we recommend that you configure the read-ahead setting to 1 MiB. This is a per-block-device setting that should only be applied to your HDD volumes. The following examples assume that you are on an Amazon Linux instance.

To examine the current value of read-ahead for your block devices, use the following command:

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

Block device information is returned in the following format:

Amazon Elastic Compute Cloud User Guide for Linux Instances EBS Performance

RO	RA	SSZ	BSZ	StartSec	Size	Device
rw	256	512	4096	4096	8587820544	/dev/<device>

The device shown reports a read-ahead value of 256 bytes (the default). Multiply this number by the sector size (512 bytes) to obtain the size of the read-ahead buffer, which in this case is 128 KiB. To set the buffer value to 1 MiB, use the following command:

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

Verify that the read-ahead setting now displays 2,048 by running the first command again.

Only use this setting when your workload consists of large, sequential I/Os. If it consists mostly of small, random I/Os, this setting will actually degrade your performance. In general, if your workload consists mostly of small or random I/Os, you should consider using a General Purpose SSD ([gp2](#)) volume rather than [st1](#) or [sc1](#).

Use a Modern Linux Kernel

Use a modern Linux kernel with support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation EC2 instance. If your average I/O size is at or near 44 KiB, you may be using an instance or kernel without support for indirect descriptors. For information about deriving the average I/O size from Amazon CloudWatch metrics, see [I/O Characteristics](#) (p. 704).

To achieve maximum throughput on [st1](#) or [sc1](#) volumes for any Linux kernel 4.2 and above, we recommend setting the `xen_blkfront.max` parameter to 256. This parameter can be set in your OS boot command line. For example, in an Amazon Linux AMI, you can add it to the end of the kernel line in the GRUB configuration found in `/boot/grub/menu.lst`:

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0  
xen_blkfront.max=256
```

Reboot your instance for this setting to take effect.

For more information, see [Configuring GRUB](#). Other Linux distributions, especially those that do not use the GRUB bootloader, may require a different approach to adjusting the kernel parameters.

For more information about EBS I/O characteristics, see the [Amazon EBS: Designing for Performance](#) re:Invent presentation on this topic.

Use RAID 0 to Maximize Utilization of Instance Resources

Some instance types can drive more I/O throughput than what you can provision for a single EBS volume. You can join multiple [gp2](#), [io1](#), [st1](#), or [sc1](#) volumes together in a RAID 0 configuration to use the available bandwidth for these instances. For more information, see [RAID Configuration on Linux](#) (p. 707).

Track Performance with Amazon CloudWatch

Amazon Web Services provides performance metrics for Amazon EBS that you can analyze and view with Amazon CloudWatch and status checks that you can use to monitor the health of your volumes. For more information, see [Monitoring the Status of Your Volumes](#) (p. 658).

Amazon EC2 Instance Configuration

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance

out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you stripe multiple volumes together in a RAID configuration.

Use EBS-Optimized or 10 Gigabit Network Instances

Any performance-sensitive workloads that require minimal variability and dedicated Amazon EC2 to Amazon EBS traffic, such as production databases or business applications, should use volumes that are attached to an EBS-optimized instance or an instance with 10 Gigabit network connectivity. EC2 instances that do not meet this criteria offer no guarantee of network resources. The only way to ensure sustained reliable network bandwidth between your EC2 instance and your EBS volumes is to launch the EC2 instance as EBS-optimized or choose an instance type with 10 Gigabit network connectivity. To see which instance types include 10 Gigabit network connectivity, see [Instance Type Details](#). For information about configuring EBS-optimized instances, see [Amazon EBS–Optimized Instances](#).

Choose an EC2 Instance with Enough Bandwidth

Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. The following table shows which instance types are available to be launched as EBS-optimized, the dedicated throughput to instance types are available to be launched as EBS-optimized, the dedicated bandwidth to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate I/O bandwidth available on that connection in MB/s. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the Amazon EBS to Amazon EC2 connection will become a performance bottleneck.

Note

The table below and the following examples use 16 KB as an I/O size for explanatory purposes only; your application I/O size may vary (Amazon EBS measures each I/O operation per second that is 256 KiB or smaller as one IOPS). For more information about IOPS and the relationship between I/O size and volume throughput limits, see [I/O Characteristics \(p. 704\)](#).

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000	250	16,000
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Performance**

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
d2.8xlarge	Yes	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000	250	16,000
m1.large		500	62.5	4,000
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	Yes	450	56.25	3,600
m4.xlarge	Yes	750	93.75	6,000
m4.2xlarge	Yes	1,000	125	8,000
m4.4xlarge	Yes	2,000	250	16,000
m4.10xlarge	Yes	4,000	500	32,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000
r3.4xlarge		2,000	250	16,000
x1.32xlarge	Yes	10,000	1,250	65,000

* These instance types must be launched as EBS-optimized to consistently achieve this level of performance.

** This value is a rounded approximation based on a 100% read-only workload and it is provided as a baseline configuration aid. EBS-optimized connections are full-duplex, and can drive more throughput and IOPS in a 50/50 read/write workload where both communication lanes are used. In some cases, network, file system, and Amazon EBS encryption overhead can reduce the maximum throughput and IOPS available.

Note that some instances with 10-gigabit network interfaces, such as `i2.8xlarge`, `c3.8xlarge`, and `r3.8xlarge`, do not offer EBS-optimization, and therefore do not have dedicated EBS bandwidth available and are not listed here. However, you can use all of that bandwidth for traffic to Amazon EBS if your application isn't pushing other network traffic that contends with Amazon EBS. Some other 10-gigabit network instances, such as `c4.8xlarge` and `d2.8xlarge` offer dedicated Amazon EBS bandwidth in addition to a 10-gigabit interface which is used exclusively for network traffic.

The `m1.large` instance has a maximum 16 KB IOPS value of 4,000, but unless this instance type is launched as EBS-optimized, that value is an absolute best-case scenario and is not guaranteed; to

consistently achieve 4,000 16 KB IOPS, you must launch this instance as EBS-optimized. However, if a 4,000 IOPS `io1` volume is attached to an EBS-optimized `m1.large` instance, the Amazon EC2 to Amazon EBS connection bandwidth limit prevents this volume from providing the 320 MB/s maximum aggregate throughput available to it. In this case, we must use an EBS-optimized EC2 instance that supports at least 320 MB/s of throughput, such as the `c4.8xlarge` instance type.

Volumes of type General Purpose SSD (`gp2`) have a throughput limit between 128 MB/s and 160 MB/s per volume (depending on volume size), which pairs well with a 1,000 Mbps EBS-optimized connection. Instance types that offer more than 1,000 Mbps of throughput to Amazon EBS can use more than one `gp2` volume to take advantage of the available throughput. Volumes of type Provisioned IOPS SSD (`io1`) have a throughput limit range of 256 KiB for each IOPS provisioned, up to a maximum of 320 MiB/s (at 1,280 IOPS). For more information, see [Amazon EBS Volume Types](#) (p. 638).

Instance types with 10 Gigabit network connectivity support up to 800 MB/s of throughput and 48,000 16K IOPS for unencrypted Amazon EBS volumes and up to 25,000 16K IOPS for encrypted Amazon EBS volumes. Because the maximum `io1` value for EBS volumes is 20,000 for `io1` volumes and 10,000 for `gp2` volumes, you can use several EBS volumes simultaneously to reach the level of I/O performance available to these instance types. For more information about which instance types include 10 Gigabit network connectivity, see [Instance Type Details](#).

You should use EBS-optimized instances when available to get the full performance benefits of Amazon EBS `gp2` and `io1` volumes. For more information, see [Amazon EBS-Optimized Instances](#) (p. 692).

I/O Characteristics

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes—General Purpose SSD (`gp2`) and Provisioned IOPS SSD (`io1`)—deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes—Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`)—deliver optimal performance only when I/O operations are large and sequential. To understand how SSD and HDD volumes will perform in your application, it is important to know the connection between demand on the volume, the quantity of IOPS available to it, the time it takes for an I/O operation to complete, and the volume's throughput limits.

IOPS

IOPS are a unit of measure representing input/output operations per second. The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at 256 KiB for SSD volumes and 1,024 KiB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes.

When small I/O operations are physically contiguous, Amazon EBS attempts to merge them into a single I/O up to the maximum size. For example, for SSD volumes, a single 1,024 KiB I/O operation would count as 4 operations, while 256 I/O operations at 4 KiB each would count as 64 operations. However, 256 random I/O operations at 4 KiB each would count as 256 operations.

Similarly, for HDD-backed volumes, both a single 1,024 KiB I/O operation and 8 sequential 128 KiB operations would count as one operation. However, 8 random 128 KiB I/O operations would count as 8 operations.

Consequently, when you create an SSD-backed volume supporting 3,000 IOPS (either by provisioning an `io1` volume at 3,000 IOPS or by sizing a `gp2` volume at 1000 GiB), and you attach it to an EBS-optimized instance that can provide sufficient bandwidth, you can transfer up to 3,000 I/Os of data per second, with throughput determined by I/O size.

Volume Queue Length and Latency

The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation, in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgement from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS.

Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to fully use the performance available to your EBS volume, then your volume might not deliver the IOPS or throughput that you have provisioned.

Transaction-intensive applications are sensitive to increased I/O latency and are well-suited for SSD-backed `io1` and `gp2` volumes. You can maintain high IOPS while keeping latency down by maintaining a low queue length and a high number of IOPS available to the volume. Consistently driving more IOPS to a volume than it has available can cause increased I/O latency.

Throughput-intensive applications are less sensitive to increased I/O latency, and are well-suited for HDD-backed `st1` and `sc1` volumes. You can maintain high throughput to HDD-backed volumes by maintaining a high queue length when performing large, sequential I/O.

I/O size and volume throughput limits

For SSD-backed volumes, if your I/O size is very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the throughput limit of the volume. For example, a `gp2` volume under 1000 GiB with burst credits available has an IOPS limit of 3,000 and a volume throughput limit of 160 MiB/s. If you are using a 256 KiB I/O size, your volume reaches its throughput limit at 640 IOPS ($640 \times 256 \text{ KiB} = 160 \text{ MiB}$). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 160 MiB/s. (These examples assume that your volume's I/O is not hitting the throughput limits of the instance.) For more information about the throughput limits for each EBS volume type, see [Amazon EBS Volume Types \(p. 638\)](#).

For smaller I/O operations, you may see a higher-than-provisioned IOPS value as measured from inside your instance. This happens when the instance operating system merges small I/O operations into a larger operation before passing them to Amazon EBS.

If your workload uses sequential I/Os on HDD-backed `st1` and `sc1` volumes, you may experience a higher than expected number of IOPS as measured from inside your instance. This happens when the instance operating system merges sequential I/Os and counts them in 1,024 KiB-sized units. If your workload uses small or random I/Os, you may experience a lower throughput than you expect. This is because we count each random, non-sequential I/O toward the total IOPS count, which can cause you to hit the volume's IOPS limit sooner than expected.

Whatever your EBS volume type, if you are not experiencing the IOPS or throughput you expect in your configuration, ensure that your EC2 instance bandwidth is not the limiting factor. You should always use a current-generation, EBS-optimized instance (or one that includes 10 Gb/s network connectivity) for optimal performance. For more information, see [Amazon EC2 Instance Configuration \(p. 701\)](#). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.

Monitor I/O Characteristics with CloudWatch

You can monitor these I/O characteristics with each volume's [CloudWatch metrics](#). Important metrics to consider include:

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance` displays the burst bucket balance for HDD-backed volumes as a percentage. When your burst bucket is depleted, volume throughput is throttled to the baseline. Check the `BurstBalance` value to determine whether your volume is being throttled for this reason.

HDD-backed `st1` and `sc1` volumes are designed to perform best with workloads that take advantage of the 1,024 KiB maximum I/O size. To determine your volume's average I/O size, divide `VolumeWriteBytes` by `VolumeWriteOps`. The same calculation applies to read operations. If average I/O size is below 64 KiB, increasing the size of the I/O operations sent to an `st1` or `sc1` volume should improve performance.

Note

If average I/O size is at or near 44 KiB, you may be using an instance or kernel without support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation instance.

If your I/O latency is higher than you require, check `VolumeQueueLength` to make sure your application is not trying to drive more IOPS than you have provisioned. If your application requires a greater number of IOPS than your volume can provide, you should consider using a larger `gp2` volume with a higher base performance level or an `io1` volume with more provisioned IOPS to achieve faster latencies.

For more information about Amazon EBS I/O characteristics, see the [Amazon EBS: Designing for Performance](#) re:Invent presentation on this topic.

Initializing Amazon EBS Volumes

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all of the blocks on your volume before you use it; this process is called *initialization*. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

Important

While initializing `io1` volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a `warning` state in the **I/O Performance** status check. This is expected, and you can ignore the `warning` state on `io1` volumes while you are initializing them. For more information, see [Monitoring Volumes with Status Checks](#) (p. 661).

Initializing Amazon EBS Volumes on Linux

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). For volumes that have been restored from snapshots, use the `dd` or `fiio` utilities to read from all of the blocks on a volume. All existing data on the volume will be preserved.

To initialize a volume restored from a snapshot on Linux

1. Attach the newly-restored volume to your Linux instance.
2. Use the `lsblk` command to list the block devices on your instance.

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf 202:80 0 30G 0 disk
```

```
xvda1 202:1    0    8G  0 disk /
```

Here you can see that the new volume, `/dev/xvdf`, is attached, but not mounted (because there is no path listed under the `MOUNTPOINT` column).

3. Use the `dd` or `fiio` utilities to read all of the blocks on the device. The `dd` command is installed by default on Linux systems, but `fiio` is considerably faster because it allows multi-threaded reads.

Note

This step may take several minutes up to several hours, depending on your EC2 instance bandwidth, the IOPS provisioned for the volume, and the size of the volume.

- **Using `dd`:** The `if` (input file) parameter should be set to the drive you wish to initialize. The `of` (output file) parameter should be set to the Linux null virtual device, `/dev/null`. The `bs` parameter sets the block size of the read operation; for optimal performance, this should be set to 1 MB.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

- **Using `fiio`:** If you have `fiio` installed on your system, you can copy and paste the command below to initialize your volume. The `--filename` (input file) parameter should be set to the drive you wish to initialize.

Note

To install `fiio` on Amazon Linux, use the following command: `sudo yum install -y fiio`

To install `fiio` on Ubuntu, use the following command: `sudo apt-get install -y fiio`

```
[ec2-user ~]$ sudo fiio --filename=/dev/xvdf --rw=randread --bs=128k --  
iodepth=32 --ioengine=libaio --direct=1 --name=volume-initialize
```

When the operation is finished, you will see a report of the read operation. Your volume is now ready for use. For more information, see [Making an Amazon EBS Volume Available for Use](#) (p. 655).

RAID Configuration on Linux

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. This replication makes Amazon EBS volumes ten times more reliable than typical commodity disk drives. For more information, see [Amazon EBS Availability and Durability](#) in the Amazon EBS product detail pages.

Note

You should avoid booting from a RAID volume. Grub is typically installed on only one device in a RAID array, and if one of the mirrored devices fails, you may be unable to boot the operating system.

If you need to create a RAID array on a Windows instance, see [RAID Configuration on Windows](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [RAID Configuration Options](#) (p. 708)
- [Creating a RAID Array on Linux](#) (p. 708)

RAID Configuration Options

The following table compares the common RAID 0 and RAID 1 options.

Configuratio	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important

RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Creating a RAID 0 array allows you to achieve a higher level of performance for a file system than you can provision on a single Amazon EBS volume. A RAID 1 array offers a "mirror" of your data for extra redundancy. Before you perform this procedure, you need to decide how large your RAID array should be and how many IOPS you want to provision.

The resulting size of a RAID 0 array is the sum of the sizes of the volumes within it, and the bandwidth is the sum of the available bandwidth of the volumes within it. The resulting size and bandwidth of a RAID 1 array is equal to the size and bandwidth of the volumes in the array. For example, two 500 GiB Amazon EBS volumes with 4,000 provisioned IOPS each will create a 1000 GiB RAID 0 array with an available bandwidth of 8,000 IOPS and 640 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 320 MB/s of throughput.

This documentation provides basic RAID setup examples. For more information about RAID configuration, performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

Creating a RAID Array on Linux

Use the following procedure to create the RAID array. Note that you can get directions for Windows instances from [Creating a RAID Array on Windows](#) in the *Amazon EC2 User Guide for Windows Instances*.

To create a RAID array on Linux

1. Create the Amazon EBS volumes for your array. For more information, see [Creating an Amazon EBS Volume](#) (p. 650).

Important

Create volumes with identical size and IOPS performance values for your array. Make sure you do not create an array that exceeds the available bandwidth of your EC2 instance. For more information, see [Amazon EC2 Instance Configuration \(p. 701\)](#).

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).
3. Use the **mdadm** command to create a logical RAID device from the newly attached Amazon EBS volumes. Substitute the number of volumes in your array for *number_of_volumes* and the device names for each volume in the array (such as `/dev/xvdf`) for *device_name*. You can also substitute *MY_RAID* with your own unique name for the array.

Note

You can list the devices on your instance with the **lsblk** command to find the device names.

(RAID 0 only) To create a RAID 0 array, execute the following command (note the `--level=0` option to stripe the array):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --  
name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

(RAID 1 only) To create a RAID 1 array, execute the following command (note the `--level=1` option to mirror the array):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=1 --  
name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

4. Allow time for the RAID array to initialize and synchronize. You can track the progress of these operations with the following command:

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

This yields output such as the following:

```
Personalities : [raid1]  
md0 : active raid1 xvdg[1] xvdf[0]  
      20955008 blocks super 1.2 [2/2] [UU]  
      [=====>.....] resync = 46.8% (9826112/20955008)  
      finish=2.9min speed=63016K/sec
```

In general, you can display detailed information about your RAID array with the following command:

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

This yields information such as the following:

```
/dev/md0:  
  Version : 1.2  
  Creation Time : Mon Jun 27 11:31:28 2016  
  Raid Level : raid1  
  Array Size : 20955008 (19.98 GiB 21.46 GB)  
  Used Dev Size : 20955008 (19.98 GiB 21.46 GB)  
  Raid Devices : 2
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS Performance**

```
Total Devices : 2
Persistence : Superblock is persistent

Update Time : Mon Jun 27 11:37:02 2016
State : clean
...
...
...

Number Major Minor RaidDevice State
0 202 80 0 active sync /dev/sdf
1 202 96 1 active sync /dev/sdg
```

- 5. Create a file system on your RAID array, and give that file system a label to use when you mount it later. For example, to create an `ext4` file system with the label `MY_RAID`, execute the following command:

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

Depending on the requirements of your application or the limitations of your operating system, you can use a different file system type, such as `ext3` or `XFS` (consult your file system documentation for the corresponding file system creation command).

- 6. Create a mount point for your RAID array.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

- 7. Finally, mount the RAID device on the mount point that you created:

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

Your RAID device is now ready for use.

- 8. (Optional) To mount this Amazon EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file.
 - a. Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while you are editing it.

```
[ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
```

- b. Open the `/etc/fstab` file using your favorite text editor, such as `nano` or `vim`.
 - c. Add a new line to the end of the file for your volume using the following format.

```
device_label mount_point file_system_type fs_mntops fs_freq
fs_passno
```

The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the example below for them (`defaults,nofail 0 2`). For more information about `/etc/fstab` entries, see the **fstab** manual page (by entering `man fstab` on the command line). For example, to mount the `ext4` file system on the device with the label `MY_RAID` at the mount point `/mnt/raid`, add the following entry to `/etc/fstab`.

Note

If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the `nofail` mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add the `nobootwait` mount option.

```
LABEL=MY_RAID      /mnt/raid  ext4  defaults,nofail  0
2
```

- d. After you've added the new entry to `/etc/fstab`, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in `/etc/fstab`.

```
[ec2-user ~]$ sudo mount -a
```

If the previous command does not produce an error, then your `/etc/fstab` file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your `/etc/fstab`.

Warning

Errors in the `/etc/fstab` file can render a system unbootable. Do not shut down a system that has errors in the `/etc/fstab` file.

- e. (Optional) If you are unsure how to correct `/etc/fstab` errors, you can always restore your backup `/etc/fstab` file with the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Benchmark EBS Volumes

This section demonstrates how you can test the performance of Amazon EBS volumes by simulating I/O workloads. The process is as follows:

1. Launch an EBS-optimized instance.
2. Create new EBS volumes.
3. Attach the volumes to your EBS-optimized instance.
4. Configure and mount the block device.
5. Install a tool to benchmark I/O performance.
6. Benchmark the I/O performance of your volumes.
7. Delete your volumes and terminate your instance so that you don't continue to incur charges.

Important

Some of the procedures described in this topic will result in the destruction of existing data on the EBS volumes you benchmark. The benchmarking procedures are intended for use on volumes specially created for testing purposes, not production volumes.

Set Up Your Instance

To get optimal performance from EBS volumes, we recommend that you use an EBS-optimized instance. EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with instance. EBS-optimized instances deliver dedicated bandwidth between Amazon EC2 and Amazon EBS, with options between 500 and 10,000 Mbps, depending on the instance type.

To create an EBS-optimized instance, choose **Launch as an EBS-Optimized instance** when launching the instance using the Amazon EC2 console, or specify `--ebs-optimized` when using the command line. Be sure that you launch a current-generation instance that supports this option. For the example tests in this topic, we recommend that you launch a `c3.4xlarge` instance. For more information, see [Amazon EBS-Optimized Instances \(p. 692\)](#).

Setting up Provisioned IOPS SSD (`io1`) volumes

To create an `io1` volume, choose **Provisioned IOPS SSD** when creating the volume using the Amazon EC2 console, or, at the command line, specify `--type io1 --iops n` where `n` is an integer between 100 and 20000. For information about creating EBS volumes, see [Creating an Amazon EBS Volume \(p. 650\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).

For the example tests, we recommend that you create a RAID array with 6 volumes, which offers a high level of performance. Because you are charged by gigabytes provisioned (and the number of provisioned IOPS for `io1` volumes), not the number of volumes, there is no additional cost for creating multiple, smaller volumes and using them to create a stripe set. If you're using Oracle Orion to benchmark your volumes, it can simulate striping the same way that Oracle ASM does, so we recommend that you let Orion do the striping. If you are using a different benchmarking tool, you need to stripe the volumes yourself.

To create a six-volume stripe set on Amazon Linux, use a command such as the following:

```
[ec2-user ~]$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

For this example, the file system is XFS. Use the file system that meets your requirements. Use the following command to install XFS file system support:

```
[ec2-user ~]$ sudo yum install -y xfsprogs
```

Then, use these commands to create, mount, and assign ownership to the XFS file system:

```
[ec2-user ~]$ sudo mkdir -p /mnt/p_iops_vol0 && sudo mkfs.xfs /dev/md0  
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_vol0  
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_vol0/
```

Setting up Throughput Optimized HDD (`st1`) or Cold HDD (`sc1`) volumes

To create an `st1` volume, choose **Throughput Optimized HDD** when creating the volume using the Amazon EC2 console, or specify `--type st1` when using the command line. To create an `sc1` volume, choose **Cold HDD** when creating the volume using the Amazon EC2 console, or specify `--type sc1` when using the command line. For information about creating EBS volumes, see [Creating an Amazon EBS Volume \(p. 650\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).

AWS provides a JSON template for use with AWS CloudFormation that simplifies this setup procedure. Access the [template](#) and save it as a JSON file. AWS CloudFormation allows you to configure your own SSH keys and offers an easy way to set up a performance test environment to evaluate `st1` volumes. The template creates a current-generation instance and a 2 TiB `st1` volume, and attaches the volume to the instance at `/dev/xvdf`.

To create an HDD volume with the template

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation/>.

2. Choose **Create Stack**.
3. Choose **Upload a Template to Amazon S3** and select the JSON template you previously obtained.
4. Give your stack a name like “ebs-perf-testing”, and select an instance type (the default is r3.8xlarge) and SSH key.
5. Choose **Next** twice, and then choose **Create Stack**.
6. After the status for your new stack moves from **CREATE_IN_PROGRESS** to **COMPLETE**, choose **Outputs** to get the public DNS entry for your new instance, which will have a 2 TiB `st1` volume attached to it.
7. Connect using SSH to your new stack as user `ec2-user`, with the hostname obtained from the DNS entry in the previous step.
8. Proceed to [Install Benchmark Tools](#) (p. 713).

Install Benchmark Tools

The following table lists some of the possible tools you can use to benchmark the performance of EBS volumes.

Tool	Description
fio	<p>For benchmarking I/O performance. (Note that fio has a dependency on <code>libaio-devel</code>.)</p> <p>To install fio on Amazon Linux, run the following command:</p> <pre>[ec2-user ~]\$ sudo yum install -y fio</pre> <p>To install fio on Ubuntu, run the following command:</p> <pre>\$ sudo apt-get install -y fio</pre>
Oracle Orion Calibration Tool	For calibrating the I/O performance of storage systems to be used with Oracle databases.

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands provided below are intended as examples to help you get started.

Choosing the Volume Queue Length

Choosing the best volume queue length based on your workload and volume type.

Queue Length on SSD-backed Volumes

To determine the optimal queue length for your workload on SSD-backed volumes, we recommend that you target a queue length of 1 for every 500 IOPS available (baseline for `gp2` volumes and the provisioned amount for `io1` volumes). Then you can monitor your application performance and tune that value based on your application requirements.

Increasing the queue length is beneficial until you achieve the provisioned IOPS, throughput or optimal system queue length value, which is currently set to 32. For example, a volume with 1,000 provisioned IOPS should target a queue length of 2. You should experiment with tuning these values up or down to see what performs best for your application.

Queue Length on HDD-backed Volumes

To determine the optimal queue length for your workload on HDD-backed volumes, we recommend that you target a queue length of at least 4 while performing 1MiB sequential I/Os. Then you can monitor your application performance and tune that value based on your application requirements. For example, a 2 TiB *st1* volume with burst throughput of 500 MiB/s and IOPS of 500 should target a queue length of 4, 8, or 16 while performing 1,024 KiB, 512 KiB, or 256 KiB sequential I/Os respectively. You should experiment with tuning these values value up or down to see what performs best for your application.

Perform Benchmarking

The following procedures describe benchmarking commands for various EBS volume types.

Run the following commands on an EBS-optimized instance with attached EBS volumes. If the EBS volumes were restored from snapshots, be sure to initialize them before benchmarking. For more information, see [Initializing Amazon EBS Volumes \(p. 706\)](#).

When you are finished testing your volumes, see the following topics for help cleaning up: [Deleting an Amazon EBS Volume \(p. 671\)](#) and [Terminate Your Instance \(p. 305\)](#).

Benchmarking *io1* Volumes

Run **fiio** on the stripe set that you created.

The following command performs 16 KB random write operations.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_vol0 \  
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G \  
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

The following command performs 16 KB random read operations.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_vol0 \  
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G \  
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

For more information about interpreting the results, see this tutorial: [Inspecting disk IO performance with fio](#).

Benchmarking *st1* and *sc1* Volumes

Run **fiio** on your *st1* or *sc1* volume.

Note

Prior to running these tests, set buffered I/O on your instance as described in [Increase Read-Ahead for High-Throughput, Read-Heavy Workloads on *st1* and *sc1* \(p. 700\)](#).

The following command performs 1 MiB sequential read operations against an attached *st1* block device (e.g., `/dev/xvdf`):

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=read \  
--randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 -- \  
runtime=180 \  
--name=fio_direct_read_test
```

The following command performs 1 MiB sequential write operations against an attached *st1* block device:

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=write
```

```
--randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --  
runtime=180  
--name=fio_direct_write_test
```

Some workloads perform a mix of sequential reads and sequential writes to different parts of the block device. To benchmark such a workload, we recommend that you use separate, simultaneous **fio** jobs for reads and writes, and use the **fio** `offset_increment` option to target different block device locations for each job.

Running this workload is a bit more complicated than a sequential-write or sequential-read workload. Use a text editor to create a **fio** job file, called `fio_rw_mix.cfg` in this example, that contains the following:

```
[global]  
clocksource=clock_gettime  
randrepeat=0  
runtime=180  
offset_increment=100g  
  
[sequential-write]  
bs=1M  
ioengine=libaio  
direct=1  
iodepth=8  
filename=/dev/<device>  
do_verify=0  
rw=write  
rwmixread=0  
rwmixwrite=100  
  
[sequential-read]  
bs=1M  
ioengine=libaio  
direct=1  
iodepth=8  
filename=/dev/<device>  
do_verify=0  
rw=read  
rwmixread=100  
rwmixwrite=0
```

Then run the following command:

```
[ec2-user ~]$ sudo fio fio_rw_mix.cfg
```

For more information about interpreting the results, see the [Inspecting disk I/O performance with fio](#) tutorial.

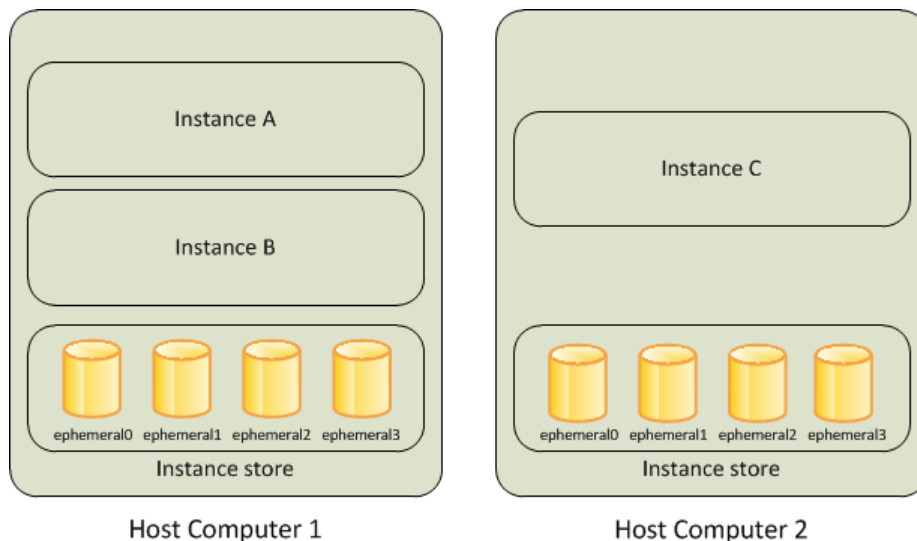
Multiple **fio** jobs for direct I/O, even though using sequential read or write operations, can result in lower than expected throughput for `st1` and `sc1` volumes. We recommend that you use one direct I/O job and use the `iodepth` parameter to control the number of concurrent I/O operations.

Amazon EC2 Instance Store

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage

of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store varies by instance type. The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer.



Contents

- [Instance Store Lifetime \(p. 716\)](#)
- [Instance Store Volumes \(p. 717\)](#)
- [Add Instance Store Volumes to Your EC2 Instance \(p. 719\)](#)
- [SSD Instance Store Volumes \(p. 722\)](#)
- [Instance Store Swap Volumes \(p. 723\)](#)
- [Optimizing Disk Performance for Instance Store Volumes \(p. 726\)](#)

Instance Store Lifetime

You can specify instance store volumes for an instance only when you launch it. The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under the following circumstances:

- The underlying disk drive fails
- The instance stops
- The instance terminates

Therefore, do not rely on instance store for valuable, long-term data. Instead, you can build a degree of redundancy (for example, RAID 1/5/6), or use a file system (for example, HDFS and MapR-FS) that supports redundancy and fault tolerance. You can also back up data periodically to more durable data storage solutions such as Amazon S3 or Amazon EBS.

You can't detach an instance store volume from one instance and attach it to a different instance. If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes of the instances that you launch from the AMI.

Instance Store Volumes

The instance type determines the size of the instance store available and the type of hardware used for the instance store volumes. Instance store volumes are included as part of the instance's hourly cost. You must specify the instance store volumes that you'd like to use when you launch the instance, and then format and mount them before using them. You can't make an instance store volume available after you launch the instance. For more information, see [Add Instance Store Volumes to Your EC2 Instance](#) (p. 719).

Some instance types use solid state drives (SSD) to deliver very high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures. For more information, see [SSD Instance Store Volumes](#) (p. 722).

The following table provides the quantity, size, type, and performance optimizations of instance store volumes available on each instance type.

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
c1.medium	1 x 350 GB†	HDD	✓	
c1.xlarge	4 x 420 GB (1,680 GB)	HDD	✓	
c3.large	2 x 16 GB (32 GB)	SSD	✓	
c3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
c3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
c3.4xlarge	2 x 160 GB (320 GB)	SSD	✓	
c3.8xlarge	2 x 320 GB (640 GB)	SSD	✓	
cc2.8xlarge	4 x 840 GB (3,360 GB)	HDD	✓	
cg1.4xlarge	2 x 840 GB (1,680 GB)	HDD	✓	
cr1.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
d2.xlarge	3 x 2,000 GB (6 TB)	HDD		
d2.2xlarge	6 x 2,000 GB (12 TB)	HDD		
d2.4xlarge	12 x 2,000 GB (24 TB)	HDD		

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Store Volumes**

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
d2.8xlarge	24 x 2,000 GB (48 TB)	HDD		
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
hi1.4xlarge	2 x 1,024 GB (2,048 GB)	SSD		
hs1.8xlarge	24 x 2,000 GB (48 TB)	HDD	✓	
i2.xlarge	1 x 800 GB	SSD		✓
i2.2xlarge	2 x 800 GB (1,600 GB)	SSD		✓
i2.4xlarge	4 x 800 GB (3,200 GB)	SSD		✓
i2.8xlarge	8 x 800 GB (6,400 GB)	SSD		✓
m1.small	1 x 160 GB†	HDD	✓	
m1.medium	1 x 410 GB	HDD	✓	
m1.large	2 x 420 GB (840 GB)	HDD	✓	
m1.xlarge	4 x 420 GB (1,680 GB)	HDD	✓	
m2.xlarge	1 x 420 GB	HDD	✓	
m2.2xlarge	1 x 850 GB	HDD	✓	
m2.4xlarge	2 x 840 GB (1,680 GB)	HDD	✓	
m3.medium	1 x 4 GB	SSD	✓	
m3.large	1 x 32 GB	SSD	✓	
m3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
m3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
r3.large	1 x 32 GB	SSD		✓
r3.xlarge	1 x 80 GB	SSD		✓
r3.2xlarge	1 x 160 GB	SSD		✓
r3.4xlarge	1 x 320 GB	SSD		✓

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Add Instance Store Volumes**

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
r3.8xlarge	2 x 320 GB (640 GB)	SSD		✓
x1.32xlarge	2 x 1,920 GB (3,840 GB)	SSD		

* Volumes attached to certain instances will suffer a first-write penalty unless initialized. For more information about initializing instance store volumes, see [Optimizing Disk Performance for Instance Store Volumes](#).

** SSD-based instance store volumes that support TRIM instructions are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see [Instance Store Volume TRIM Support \(p. 722\)](#).

† The `c1.medium` and `m1.small` instance types also include a 900 MB instance store swap volume, which may not be automatically enabled at boot time. For more information, see [Instance Store Swap Volumes \(p. 723\)](#).

Add Instance Store Volumes to Your EC2 Instance

You specify the EBS volumes and instance store volumes for your instance using a block device mapping. Each entry in a block device mapping includes a device name and the volume that it maps to. The default block device mapping is specified by the AMI you use. Alternatively, you can specify a block device mapping for the instance when you launch it. For more information, see [Block Device Mapping \(p. 731\)](#).

A block device mapping always specifies the root volume for the instance. The root volume is either an Amazon EBS volume or an instance store volume. For more information, see [Storage for the Root Device \(p. 71\)](#). The root volume is mounted automatically. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GB.

You can use a block device mapping to specify additional EBS volumes when you launch your instance, or you can attach additional EBS volumes after your instance is running. For more information, see [Amazon EBS Volumes \(p. 636\)](#).

You can specify the instance store volumes for your instance only when you launch an instance. You can't attach instance store volumes to an instance after you've launched it.

The number and size of available instance store volumes for your instance varies by instance type. Some instance types do not support instance store volumes. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 717\)](#). If the instance type you choose for your instance supports instance store volumes, you must add them to the block device mapping for the instance when you launch it. After you launch the instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them. Note that the root volume of an instance store-backed instance is mounted automatically.

Contents

- [Adding Instance Store Volumes to an AMI \(p. 720\)](#)
- [Adding Instance Store Volumes to an Instance \(p. 720\)](#)
- [Making Instance Store Volumes Available on Your Instance \(p. 721\)](#)

Adding Instance Store Volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. After you add instance store volumes to an AMI, any instance that you launch from the AMI includes these instance store volumes. Note that when you launch an instance, you can omit volumes specified in the AMI block device mapping and add new volumes.

Important

For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volumes that are specified only in the block device mapping of the AMI.

To add instance store volumes to an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions**, select **Image**, and then select **Create Image**.
4. In the **Create Image** dialog, add a meaningful name and description for your image.
5. For each instance store volume to add, choose **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. (For more information, see [Device Naming on Linux Instances \(p. 729\)](#).) The number of available instance store volumes depends on the instance type.

The screenshot shows the 'Create Image' dialog in the Amazon EC2 console. It displays a table for block device mapping with columns: Type, Device, Snapshot, and Size (GiB). The first row is for the 'Root' device, mapped to '/dev/xvda', using snapshot 'snap-bfb086e1', with a size of 8 GiB. Below this, there are two rows for instance store volumes. The first row is for 'Instance Store 0', mapped to '/dev/sdb', with a size of 'N/A'. The second row is for 'Instance Store 1', mapped to '/dev/sdc', with a size of 8 GiB. A dropdown menu is open for the 'Type' column of the second row, showing options: EBS, Instance Store 1, and Add New Volume. A search box with the text 'Search (case-insensitive)' is visible next to the size input for the second row.

Type	Device	Snapshot	Size (GiB)
Root	/dev/xvda	snap-bfb086e1	8
Instance Store 0	/dev/sdb	N/A	N/A
EBS	/dev/sdc	Search (case-insensitive)	8

6. Click **Create Image**.

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-image` or `register-image` (AWS CLI)
- `New-EC2Image` and `Register-EC2Image` (AWS Tools for Windows PowerShell)

Adding Instance Store Volumes to an Instance

When you launch an instance, the default block device mapping is provided by the specified AMI. If you need additional instance store volumes, you must add them to the instance as you launch it. Note that you can also omit devices specified in the AMI block device mapping.

Important

For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.

Important

For HS1 instances, no matter how many instance store volumes you specify in the block device mapping of an AMI, the block device mapping for an instance launched from the

AMI automatically includes the maximum number of supported instance store volumes. You must explicitly remove the instance store volumes that you don't want from the block device mapping for the instance before you launch it.

To update the block device mapping for an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, select the AMI to use and choose **Select**.
4. Follow the wizard to complete **Step 1: Choose an Amazon Machine Image (AMI)**, **Step 2: Choose an Instance Type**, and **Step 3: Configure Instance Details**.
5. In **Step 4: Add Storage**, modify the existing entries as needed. For each instance store volume to add, click **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. The number of available instance store volumes depends on the instance type.

Type	Device	Snapshot	Size (GiB)
Root	/dev/xvda	snap-bfb086e1	8
Instance Store 0	/dev/sdb	N/A	N/A
Instance Store 1	/dev/sdc	Search (case-insensitiv	8

6. Complete the wizard to launch the instance.

To update the block device mapping for an instance using the command line

You can use one of the following options commands with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--block-device-mappings` with `run-instances` (AWS CLI)
- `-BlockDeviceMapping` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Making Instance Store Volumes Available on Your Instance

After you launch an instance, the instance store volumes are available to the instance, but you can't access them until they are mounted. For Linux instances, the instance type determines which instance store volumes are mounted for you and which are available for you to mount yourself. For Windows instances, the EC2Config service mounts the instance store volumes for an instance. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

Many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see [Instance Store Volume TRIM Support \(p. 722\)](#). For Windows instances, the EC2Config service reformats the instance store volumes with the NTFS file system.

You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 739\)](#).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see [Listing the Disks Using Windows Disk Management](#).

For Linux instances, you can view and mount the instance store volumes as described in the following procedure.

To make an instance store volume available on Linux

1. Connect to the instance using an SSH client.
2. Use the `df -h` command to view the volumes that are formatted and mounted. Use the `lsblk` to view any volumes that were mapped at launch but not formatted and mounted.
3. To format and mount an instance store volume that was mapped only, do the following:
 - a. Create a file system on the device using the `mkfs` command.
 - b. Create a directory on which to mount the device using the `mkdir` command.
 - c. Mount the device on the newly created directory using the `mount` command.

SSD Instance Store Volumes

The following instances support instance store volumes that use solid state drives (SSD) to deliver very high random I/O performance: C3, G2, H1, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see [Instance Store Volumes \(p. 717\)](#).

To ensure the best IOPS performance from your SSD instance store volumes on Linux, we recommend that you use the most recent version of the [Amazon Linux AMI](#), or another Linux AMI with a kernel version of 3.8 or later. If you do not use a Linux AMI with a kernel version of 3.8 or later, your instance will not achieve the maximum IOPS performance available for these instance types.

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it, and the data on an SSD instance volume persists only for the life of its associated instance. For more information, see [Add Instance Store Volumes to Your EC2 Instance \(p. 719\)](#).

Instance Store Volume TRIM Support

The following instances support SSD volumes with TRIM: I2, R3.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information about using TRIM commands, see the documentation for the operating system for your instance.

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should specify the file system-specific option that skips the TRIM operation when you format them. On Linux, you should also add the `discard` option to your mount command or `/etc/fstab` file entries for the devices that support TRIM so that they use this feature effectively. On Windows, use the following command: `fsutil behavior set DisableDeleteNotify 1`.

To make an instance store volume with TRIM support available for use on Linux

1. Map the instance store volume when you launch the instance. For more information, see [Add Instance Store Volumes to Your EC2 Instance \(p. 719\)](#).
2. From the instance, list the available devices using the `lsblk` command or [view the instance store volumes using instance metadata \(p. 739\)](#).

- Verify that your operating system and device support TRIM using the following command (replacing `xvdb` with the name of your device):

```
[ec2-user ~]$ sudo cat /sys/block/xvdb/queue/discard_max_bytes
322122547200
```

If this command returns a value other than 0, then your operating system and device support TRIM.

- Format the volume with the file system of your choice.
 - (EXT4) To format the volume with the `ext4` file system, use the following command (replacing `xvdc` with the name of your device):

```
[ec2-user ~]$ sudo mkfs.ext4 -E nodiscard /dev/xvdc
```

- (XFS) To format the volume with the `xfs` file system, use the following command (replacing `xvdb` with the name of your device):

```
[ec2-user ~]$ sudo mkfs.xfs -K /dev/xvdb
```

Note

You might need to install XFS file system support on your operating system for this command to work. For Amazon Linux, use the `sudo yum install -y xfsprogs` command.

- Mount the device using the `discard` option. Be sure to specify the device name of the volume. You can select an existing directory or create a new one using the `mkdir` command.

```
[ec2-user ~]$ sudo mount -o discard /dev/xvdb /mnt/my-data
```

- (Optional) If you want the device to mount at boot time, you can add or modify an entry in the `/etc/fstab` file with the `discard` option.

```
/dev/xvdb /mnt/xvdb xfs defaults,nofail,discard 0 2
/dev/xvdc /mnt/xvdc ext4 defaults,nofail,discard 0 2
```

Important

After you edit the `/etc/fstab` file, verify that there are no errors running the `sudo mount -a` command. If there are any errors in this file, the system may not boot properly or at all.

Instance Store Swap Volumes

Swap space in Linux can be used when a system requires more memory than it has been physically allocated. When swap space is enabled, Linux systems can swap infrequently used memory pages from physical memory to swap space (either a dedicated partition or a swap file in an existing file system) and free up that space for memory pages that require high speed access.

Note

Using swap space for memory paging is not as fast or efficient as using RAM. If your workload is regularly paging memory into swap space, you should consider migrating to a larger instance type with more RAM. For more information, see [Resizing Your Instance \(p. 189\)](#).

The `c1.medium` and `m1.small` instance types have a limited amount of physical memory to work with, and they are given a 900 MiB swap volume at launch time to act as virtual memory for Linux

Amazon Elastic Compute Cloud User Guide for Linux Instances Instance Store Swap Volumes

AMIs. Although the Linux kernel sees this swap space as a partition on the root device, it is actually a separate instance store volume, regardless of your root device type.

Amazon Linux AMIs automatically enable and use this swap space, but your AMI may require some additional steps to recognize and use this swap space. To see if your instance is using swap space, you can use the **swapon -s** command.

```
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type      Size      Used
Priority
/dev/xvda3                             partition 917500    0        -1
```

The above instance has a 900 MiB swap volume attached and enabled. If you don't see a swap volume listed with this command, you may need to enable swap space for the device. Check your available disks using the **lsblk** command.

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1   0    8G  0 disk /
xvda3 202:3   0   896M 0 disk
```

Here, the swap volume `xvda3` is available to the instance, but it is not enabled (notice that the `MOUNTPOINT` field is empty). You can enable the swap volume with the **swapon** command.

Note

You need to prepend `/dev/` to the device name listed by **lsblk**. Your device may be named differently, such as `sda3`, `sde3`, or `xvde3`. Use the device name for your system in the command below.

```
[ec2-user@ip-12-34-56-78 ~]$ sudo swapon /dev/xvda3
```

Now the swap space should show up in **lsblk** and **swapon -s** output.

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1   0    8G  0 disk /
xvda3 202:3   0   896M 0 disk [SWAP]
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type      Size      Used
Priority
/dev/xvda3                             partition 917500    0        -1
```

You will also need to edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user@ip-12-34-56-78 ~]$ sudo vim /etc/fstab
```

Append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvda3    none    swap    sw    0    0
```

To use an instance store volume as swap space

Any instance store volume can be used as swap space. For example, the `m3.medium` instance type includes a 4 GB SSD instance store volume that is appropriate for swap space. If your instance store

volume is much larger (for example, 350 GB), you may consider partitioning the volume with a smaller swap partition of 4-8 GB and the rest for a data volume.

Note

This procedure applies only to instance types that support instance storage. For a list of supported instance types, see [Instance Store Volumes \(p. 717\)](#).

1. List the block devices attached to your instance to get the device name for your instance store volume.

```
[ec2-user ~]$ lsblk -p
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/xvdb   202:16  0    4G  0 disk /media/ephemeral0
/dev/xvda1  202:1   0    8G  0 disk /
```

In this example, the instance store volume is `/dev/xvdb`. Because this is an Amazon Linux instance, the instance store volume is formatted and mounted at `/media/ephemeral0`; not all Linux operating systems do this automatically.

2. (Optional) If your instance store volume is mounted (it will list a `MOUNTPOINT` in the `lsblk` command output), you need to unmount it with the following command.

```
[ec2-user ~]$ sudo umount /dev/xvdb
```

3. Set up a Linux swap area on the device with the `mkswap` command.

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swspace version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. Enable the new swap space.

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. Verify that the new swap space is being used.

```
[ec2-user ~]$ swapon -s
Filename      Type      Size Used Priority
/dev/xvdb    partition 4188668 0  -1
```

6. Edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

If your `/etc/fstab` file has an entry for `/dev/xvdb` (or `/dev/sdb`) change it to match the line below; if it does not have an entry for this device, append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvdb    none    swap    sw    0    0
```

Important

Instance store volume data is lost when an instance is stopped; this includes the instance store swap space formatting created in [Step 3 \(p. 725\)](#). If you stop and restart an instance that has been configured to use instance store swap space, you must repeat [Step 1 \(p. 725\)](#) through [Step 5 \(p. 725\)](#) on the new instance store volume.

Optimizing Disk Performance for Instance Store Volumes

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on most instance store volumes performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you initialize your drives by writing once to every drive location before production use.

Note

Some instance types with direct-attached solid state drives (SSD) and TRIM support provide maximum performance at launch time, without initialization. For information about the instance store for each instance type, see [Instance Store Volumes \(p. 717\)](#).

If you require greater flexibility in latency or throughput, we recommend using Amazon EBS.

To initialize the instance store volumes, use the following `dd` commands, depending on which store you want to initialize (for example, `/dev/sdb`).

Note

Make sure to unmount the drive before performing this command.
Initialization can take a long time (about 8 hours for an extra large instance).

To initialize the instance store volumes, use the following commands on the `m1.large`, `m1.xlarge`, `c1.xlarge`, `m2.xlarge`, `m2.2xlarge`, and `m2.4xlarge` instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

To perform initialization on all instance store volumes at the same time, use the following command:

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

Configuring drives for RAID initializes them by writing to every drive location. When configuring software-based RAID, make sure to change the minimum reconstruction speed:

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

Amazon Simple Storage Service (Amazon S3)

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easy by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures.

You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see [Amazon Elastic Block Store \(p. 634\)](#).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address. For example, if an object with a key value `/photos/mygarden.jpg` is stored in the `myawsbucket` bucket, then it is addressable using the URL `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg`.

For more information about the features of Amazon S3, see the [Amazon S3 product page](#).

Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
wget http://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

This method requires that the object you request is public; if the object is not public, you receive an `ERROR 403: Forbidden` message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the [Amazon Simple Storage Service Developer Guide](#).

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI allows users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the [AWS Command Line Interface detail page](#).

The **aws s3 cp** command is similar to the Unix **cp** command (the syntax is: **aws s3 cp source destination**). You can copy files from Amazon S3 to your instance, you can copy files from your instance to Amazon S3, and you can even copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

Use the following command to copy an object from one Amazon S3 location to another.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext  
s3://my_bucket/my_folder/my_file2.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. The command syntax is: **aws s3 sync source destination**. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
$ aws s3 sync s3://remote_S3_bucket local_directory
```

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the [AWS Tools for Windows PowerShell](#) to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -  
LocalFile my_copied_file.ext
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the [Amazon Simple Storage Service Developer Guide](#). You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the `botocore` Python interface.

Instance Volume Limits

The maximum number of volumes that your instance can have depends on the operating system. When considering how many volumes to add to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Contents

- [Linux-Specific Volume Limits \(p. 728\)](#)
- [Windows-Specific Volume Limits \(p. 729\)](#)
- [Bandwidth vs Capacity \(p. 729\)](#)

Linux-Specific Volume Limits

Attaching more than 40 volumes can cause boot failures. Note that this number includes the root volume, plus any attached instance store volumes and EBS volumes. If you experience boot problems

on an instance with a large number of volumes, stop the instance, detach any volumes that are not essential to the boot process, and then reattach the volumes after the instance is running.

Important

Attaching more than 40 volumes to a Linux instance is supported on a best effort basis only and is not guaranteed.

Windows-Specific Volume Limits

The following table shows the volume limits for Windows instances based on the driver used. Note that these numbers include the root volume, plus any attached instance store volumes and EBS volumes.

Important

Attaching more than the following volumes to a Windows instance is supported on a best effort basis only and is not guaranteed.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We do not recommend that you give a Windows instance more than 26 volumes with AWS PV or Citrix PV drivers, as it is likely to cause performance issues.

To determine which PV drivers your instance is using, or to upgrade your Windows instance from Red Hat to Citrix PV drivers, see [Upgrading PV Drivers on Your Windows Instance](#).

For more information about how device names related to volumes, see [Mapping Disks to Volumes on Your Windows EC2 Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Bandwidth vs Capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose SSD or Provisioned IOPS SSD volumes. Follow the guidance in [Amazon EC2 Instance Configuration \(p. 701\)](#) to match the IOPS you have provisioned for your volumes to the bandwidth available from your instances for maximum performance. For RAID configurations, many administrators find that arrays larger than 8 volumes have diminished performance returns due to increased I/O overhead. Test your individual application performance and tune it as required.

Device Naming on Linux Instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

Contents

- [Available Device Names \(p. 730\)](#)
- [Device Name Considerations \(p. 730\)](#)

For information about device names on Windows instances, see [Device Naming on Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Available Device Names

The following table lists the available device names for Linux instances. The number of volumes that you can attach to your instance is determined by the operating system. For more information, see [Instance Volume Limits \(p. 728\)](#).

Virtualization Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
Paravirtual	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[b-e] /dev/sd[b-y] (hs1.8xlarge)	/dev/sd[f-p] /dev/sd[f-p][1-6]
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	Differs by AMI* /dev/sda1 or /dev/xvda	/dev/sd[b-e] /dev/sd[b-y] (hs1.8xlarge) /dev/sd[b-i] (i2.8xlarge)	/dev/sd[f-p]

Note that you can determine the root device name for your particular AMI with the following AWS CLI command:

```
aws ec2 describe-images --image-ids image_id --query Images[.].RootDeviceName
```

For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 715\)](#). For information about the root device storage, see [Amazon EC2 Root Device Volume \(p. 14\)](#).

Device Name Considerations

Keep the following in mind when selecting a device name:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of `/dev/sdh`, your device might be renamed `/dev/xvdh` or `/dev/hdh` by the kernel; in most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), even the trailing letter might also change (where `/dev/sda` could become `/dev/xvde`). In these cases, each device name trailing letter is incremented the same number of times. For example, `/dev/sdb` would become `/dev/xvdf` and `/dev/sdc` would become `/dev/xvdg`. Amazon Linux AMIs create a symbolic link with the name you specify at launch that points to the renamed device path, but other AMIs might behave differently.
- There are two types of virtualization available for Linux instances: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type of an instance is determined by the AMI used to launch the instance. Some instance types support both PV and HVM, some support HVM only,

and others support PV only. Be sure to note the virtualization type of your AMI, because the recommended and available device names that you can use depend on the virtualization type of your instance. For more information, see [Linux AMI Virtualization Types \(p. 74\)](#).

- You cannot attach volumes that share the same device letters both with and without trailing digits. For example, if you attach a volume as `/dev/sdc` and another volume as `/dev/sdc1`, only `/dev/sdc` is visible to the instance. To use trailing digits in device names, you must use trailing digits on all device names that share the same base letters (such as `/dev/sdc1`, `/dev/sdc2`, `/dev/sdc3`).
- Hardware virtual machine (HVM) AMIs don't support the use of trailing numbers on device names.
- Some custom kernels might have restrictions that limit use to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`. If you're having trouble using `/dev/sd[q-z]` or `/dev/sd[q-z][1-6]`, try switching to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`.

Block Device Mapping

Each instance that you launch has an associated root device volume, either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach them as the instance is launched.

For more information about root device volumes, see [Changing the Root Device Volume to Persist \(p. 17\)](#).

Contents

- [Block Device Mapping Concepts \(p. 731\)](#)
- [AMI Block Device Mapping \(p. 734\)](#)
- [Instance Block Device Mapping \(p. 736\)](#)

Block Device Mapping Concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer. Amazon EC2 supports two types of block devices:

- Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- EBS volumes (remote storage devices)

A *block device mapping* defines the block devices (instance store volumes and EBS volumes) to attach to an instance. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance.

Contents

- [Block Device Mapping Entries \(p. 732\)](#)
- [Block Device Mapping Instance Store Caveats \(p. 732\)](#)
- [Example Block Device Mapping \(p. 733\)](#)

- [How Devices Are Made Available in the Operating System \(p. 733\)](#)

Block Device Mapping Entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. For more information, see [Device Naming on Linux Instances \(p. 729\)](#).

Important

The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

- [Instance store volumes] The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.
- [EBS volumes] The ID of the snapshot to use to create the block device (`snap-xxxxxxx`). This value is optional as long as you specify a volume size.
- [EBS volumes] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [EBS volumes] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true` for the root device volume and `false` for attached volumes. When you create an AMI, its block device mapping inherits this setting from the instance. When you launch an instance, it inherits this setting from the AMI.
- [EBS volumes] The volume type, which can be `gp2` for General Purpose SSD, `io1` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic. The default value is `gp2` in the Amazon EC2 console, and `standard` in the AWS SDKs and the AWS CLI.
- [EBS volumes] The number of input/output operations per second (IOPS) that the volume supports. (Not used with `gp2`, `st1`, `sc1`, or `standard` volumes.)

Block Device Mapping Instance Store Caveats

There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

Note

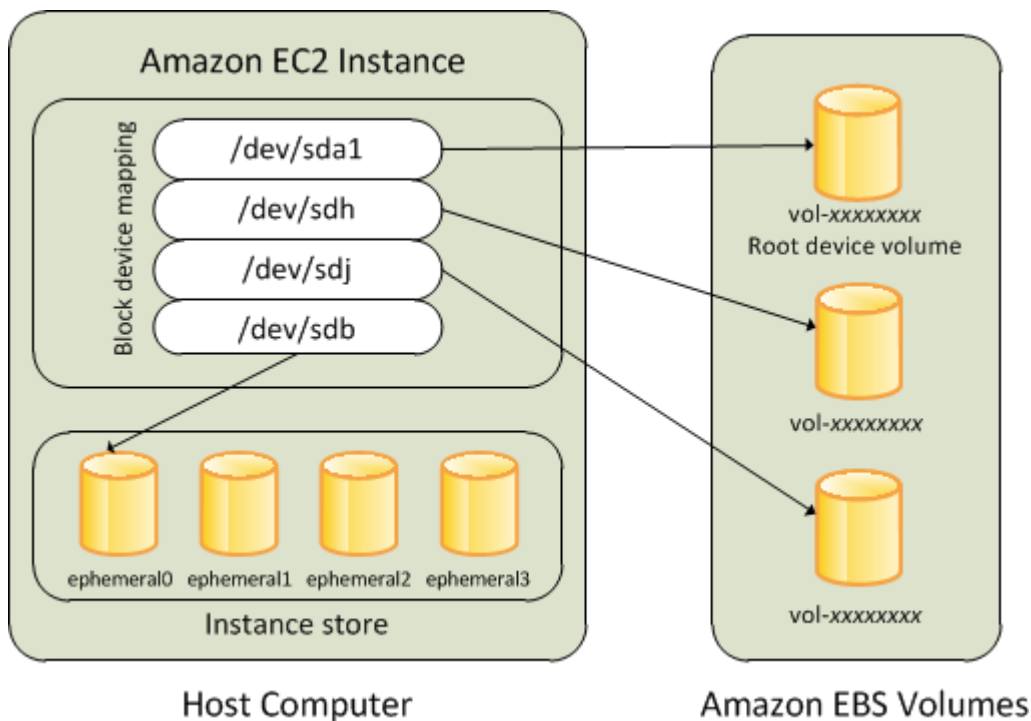
When an instance is stopped, all data on the instance store volumes is lost.

- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance

store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example Block Device Mapping

This figure shows an example block device mapping for an EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings in [Specifying a Block Device Mapping for an AMI \(p. 734\)](#) and [Updating the Block Device Mapping when Launching an Instance \(p. 736\)](#).

How Devices Are Made Available in the Operating System

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Linux instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots. The instance type determines which instance store volumes are formatted and mounted by default. You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes available for your instance type. For more information, see [Amazon EC2 Instance Store \(p. 715\)](#). The block device driver for the instance determines which devices are used when the volumes are formatted and mounted. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 654\)](#).

AMI Block Device Mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add more block devices to an AMI, you must create your own AMI.

Contents

- [Specifying a Block Device Mapping for an AMI \(p. 734\)](#)
- [Viewing the EBS Volumes in an AMI Block Device Mapping \(p. 735\)](#)

Specifying a Block Device Mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add instance store volumes only by modifying the block device mapping entries in the image manifest file when registering the image.

Note

For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select an instance and choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, choose **Add New Volume**.
5. Select a volume type from the **Type** list and a device name from the **Device** list. For an EBS volume, you can optionally specify a snapshot, volume size, and volume type.
6. Choose **Create Image**.

To add volumes to an AMI using the command line

Use the [create-image](#) AWS CLI command to specify a block device mapping for an EBS-backed AMI. Use the [register-image](#) AWS CLI command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

To add an instance store volume, use the following mapping:

```
{  
  "DeviceName": "/dev/sdf",  
  "VirtualName": "ephemeral0"
```

```
}
```

To add an empty 100 GiB Magnetic volume, use the following mapping:

```
{  
  "DeviceName": "/dev/sdg",  
  "Ebs": {  
    "VolumeSize": 100  
  }  
}
```

To add an EBS volume based on a snapshot, use the following mapping:

```
{  
  "DeviceName": "/dev/sdh",  
  "Ebs": {  
    "SnapshotId": "snap-xxxxxxxx"  
  }  
}
```

To omit a mapping for a device, use the following mapping:

```
{  
  "DeviceName": "/dev/sdj",  
  "NoDevice": ""  
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the following commands (AWS Tools for Windows PowerShell):

- [New-EC2Image](#)
- [Register-EC2Image](#)

Viewing the EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:
 - **Root Device Type** (ebs)
 - **Root Device Name** (for example, `/dev/sda1`)
 - **Block Devices** (for example, `/dev/sda1=snap-1234567890abcdef0:8:true`)

If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (Recall that this screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the command line

Use the [describe-images](#) (AWS CLI) command or [Get-EC2Image](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance Block Device Mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI. However,

Limits

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

Contents

- [Updating the Block Device Mapping when Launching an Instance](#) (p. 736)
- [Updating the Block Device Mapping of a Running Instance](#) (p. 737)
- [Viewing the EBS Volumes in an Instance Block Device Mapping](#) (p. 738)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes](#) (p. 739)

Updating the Block Device Mapping when Launching an Instance

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

To add volumes to an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
 - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
 - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
 - To add an EBS volume, choose **Add New Volume**, choose **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and choose its **Delete** icon.

- To add an instance store volume, choose **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.
6. Complete the remaining wizard pages, and choose **Launch**.

To add volumes to an instance using the comamnd line

Use the [run-instances](#) AWS CLI command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- /dev/sdb=ephemeral0
- /dev/sdh=snap-1234567890abcdef0
- /dev/sdj=:100

To prevent /dev/sdj from attaching to an instance launched from this AMI, use the following mapping:

```
{  
  "DeviceName": "/dev/sdj",  
  "NoDevice": ""  
}
```

To increase the size of /dev/sdh to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for /dev/sdh, because specifying the device name is enough to identify the volume.

```
{  
  "DeviceName": "/dev/sdh",  
  "Ebs": {  
    "VolumeSize": 300  
  }  
}
```

To attach an additional instance store volume, /dev/sdc, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect.

```
{  
  "DeviceName": "/dev/sdc",  
  "VirtualName": "ephemeral1"  
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

Updating the Block Device Mapping of a Running Instance

You can use the following [modify-instance-attribute](#) AWS CLI command to update the block device mapping of a running instance. Note that you do not need to stop the instance before changing this attribute.


```
$ aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings file://mapping.json
```

For example, to preserve the root volume at instance termination, specify the following in `mapping.json`:

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the `Edit-EC2InstanceAttribute` command (AWS Tools for Windows PowerShell).

Viewing the EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the EBS volumes mapped to an instance.

Note

For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. In the search bar, type **Root Device Type**, and then choose **EBS**. This displays a list of EBS-backed instances.
4. Select the desired instance and look at the details displayed in the **Description** tab. At a minimum, the following information is available for the root device:

- **Root device type** (ebs)
- **Root device** (for example, `/dev/sda1`)
- **Block devices** (for example, `/dev/sda1`, `/dev/sdh`, and `/dev/sdj`)

If the instance was launched with additional EBS volumes using a block device mapping, the **Block devices** field displays those additional volumes as well as the root device. (Recall that this dialog box doesn't display instance store volumes.)

Root device type	ebs
Root device	<code>/dev/sda1</code>
Block devices	<code>/dev/sda1</code> <code>/dev/sdf</code>

5. To display additional information about a block device, select its entry next to **Block devices**. This displays the following information for the block device:

- **EBS ID** (vol-xxxxxxxx)

- **Root device type** (*ebs*)
- **Attachment time** (*yyyy-mmTth:mm:ss.ssTZD*)
- **Block device status** (*attaching, attached, detaching, detached*)
- **Delete on termination** (*Yes, No*)

To view the EBS volumes for an instance using the command line

Use the [describe-instances](#) (AWS CLI) command or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an instance.

Viewing the Instance Block Device Mapping for Instance Store Volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. You can use instance metadata to query the complete block device mapping. The base URI for all requests for instance metadata is <http://169.254.169.254/latest/>.

First, connect to your running instance.

Use this query on a running instance to get its block device mapping.

```
$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store-backed `m1.small` instance looks like this.

```
ami
ephemeral0
root
swap
```

The `ami` device is the root device as seen by the instance. The instance store volumes are named `ephemeral[0-23]`. The `swap` device is for the page file. If you've also mapped EBS volumes, they appear as `ebs1`, `ebs2`, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

For more information, see [Instance Metadata and User Data](#) (p. 337).

Using Public Data Sets

Amazon Web Services provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

Contents

- [Public Data Set Concepts](#) (p. 740)
- [Finding Public Data Sets](#) (p. 740)
- [Creating a Public Data Set Volume from a Snapshot](#) (p. 741)
- [Attaching and Mounting the Public Data Set Volume](#) (p. 742)

Public Data Set Concepts

Previously, large data sets such as the mapping of the Human Genome and the US Census data required hours or days to locate, download, customize, and analyze. Now, anyone can access these data sets from an EC2 instance and start computing on the data within minutes. You can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, you can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.

For more information, go to the [Public Data Sets on AWS Page](#).

Available Public Data Sets

Public data sets are currently available in the following categories:

- **Biology**—Includes Human Genome Project, GenBank, and other content.
- **Chemistry**—Includes multiple versions of PubChem and other content.
- **Economics**—Includes census data, labor statistics, transportation statistics, and other content.
- **Encyclopedic**—Includes Wikipedia content from multiple sources and other content.

Finding Public Data Sets

Before you can use a public data set, you must locate the data set and determine which format the data set is hosted in. The data sets are available in two possible formats: Amazon EBS snapshots or Amazon S3 buckets.

To find a public data set and determine its format

1. Go to the [Public Data Sets Page](#) to see a listing of all available public data sets. You can also enter a search phrase on this page to query the available public data set listings.
2. Click the name of a data set to see its detail page.
3. On the data set detail page, look for a snapshot ID listing to identify an Amazon EBS formatted data set or an Amazon S3 URL.

Data sets that are in snapshot format are used to create new EBS volumes that you attach to an EC2 instance. For more information, see [Creating a Public Data Set Volume from a Snapshot](#) (p. 741).

For data sets that are in Amazon S3 format, you can use the AWS SDKs or the HTTP query API to access the information, or you can use the AWS CLI to copy or synchronize the data to and from your instance. For more information, see [Amazon S3 and Amazon EC2 \(p. 727\)](#).

You can also use Amazon EMR to analyze and work with public data sets. For more information, see [What is Amazon EMR?](#).

Creating a Public Data Set Volume from a Snapshot

To use a public data set that is in snapshot format, you create a new volume, specifying the snapshot ID of the public data set. You can create your new volume using the AWS Management Console as follows. If you prefer, you can use the [create-volume](#) AWS CLI command instead.

To create a public data set volume from a snapshot

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your data set snapshot is located in.

Important

Snapshot IDs are constrained to a single region, and you cannot create a volume from a snapshot that is located in another region. In addition, you can only attach an EBS volume to an instance in the same Availability Zone. For more information, see [Resource Locations \(p. 743\)](#).

If you need to create this volume in a different region, you can copy the snapshot to your required region and then restore it to a volume in that region. For more information, see [Copying an Amazon EBS Snapshot \(p. 688\)](#).

3. In the navigation pane, click **Volumes**.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Type** list, select **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types \(p. 638\)](#).
6. In the **Snapshot** field, start typing the ID or description of the snapshot for your data set. Select the snapshot from the list of suggested options.

Note

If the snapshot ID you are expecting to see does not appear, you may have a different region selected in the Amazon EC2 console. If the data set you identified in [Finding Public Data Sets \(p. 740\)](#) does not specify a region on its detail page, it is likely contained in the `us-east-1` US East (N. Virginia) region.

7. In the **Size** field, enter the size of the volume (in GiB or TiB), or verify that the default size of the snapshot is adequate.

Note

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list.

8. For Provisioned IOPS SSD volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
9. In the **Availability Zone** list, select the Availability Zone in which to launch the instance.

Important

EBS volumes can only be attached to instances in the same Availability Zone.

10. Click **Yes, Create**.

Important

If you created a larger volume than the default size for that snapshot (by specifying a size in [Step 7 \(p. 741\)](#)), you need to extend the file system on the volume to take advantage

of the extra space. For more information, see [Expanding the Storage Space of an EBS Volume on Linux](#) (p. 671).

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an EC2 instance to access the data (this instance must also be in the same Availability Zone as the new volume). For more information, see [Attaching an Amazon EBS Volume to an Instance](#) (p. 654).

After you have attached the volume to an instance, you need to mount the volume on the instance. For more information, see [Making an Amazon EBS Volume Available for Use](#) (p. 655).

Resources and Tags

Amazon EC2 provides different *resources* that you can create and use. Some of these resources include images, instances, volumes, and snapshots. When you create a resource, we assign the resource a unique resource ID.

Some resources can be tagged with values that you define, to help you organize and identify them.

The following topics describe resources and tags, and how you can work with them.

Topics

- [Resource Locations \(p. 743\)](#)
- [Resource IDs \(p. 744\)](#)
- [Listing and Filtering Your Resources \(p. 748\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 751\)](#)
- [Amazon EC2 Service Limits \(p. 760\)](#)
- [Amazon EC2 Usage Reports \(p. 762\)](#)

Resource Locations

The following table describes which Amazon EC2 resources are global, regional, or based on Availability Zone.

Resource	Type	Description
AWS account	Global	You can use the same AWS account in all regions.
Key pairs	Global or Regional	You can use the key pairs that you create using Amazon EC2 only in the region where you created them. You can create and upload an RSA key pair that you can use in all regions. For more information, see Amazon EC2 Key Pairs (p. 481) .

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Resource IDs**

Resource	Type	Description
Amazon EC2 resource identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.
User-supplied resource names	Regional	Each resource name, such as a security group name or key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within Amazon S3. You can copy an AMI from one region to another. For more information, see Copying an AMI (p. 135) .
Elastic IP addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.
EBS snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another. For more information, see Copying an Amazon EBS Snapshot (p. 688) .
EBS volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.

Resource IDs

When resources are created, we assign each resource a unique resource ID. You can use resource IDs to find your resources in the Amazon EC2 console. If you are using a command line tool or the Amazon EC2 API to work with Amazon EC2, resource IDs are required for certain commands. For example, if you are using the [stop-instances](#) AWS CLI command to stop an instance, you must specify the instance ID in the command.

Resource ID Length

A resource ID takes the form of a resource identifier (such as `snap` for a snapshot) followed by a hyphen and a unique combination of letters and numbers. Starting in January 2016, we're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. The length of the alphanumeric character combination was in an 8-character format; the new IDs are in a 17-character format, for example, `i-1234567890abcdef0` for an instance ID.

Supported resource types will have an opt-in period, during which you can enable the longer ID format. After you've enabled longer IDs for a resource type, any new resources that you create are created with a longer ID unless you explicitly disable the longer ID format. A resource ID does not change after it's created; therefore, your existing resources with shorter IDs are not affected. Similarly, if you disable longer IDs for a resource type, any resources that you created with the longer IDs are not affected.

All supported resource types will have a deadline date, after which all new resources of this type default to the longer ID format, and you can no longer disable the longer ID format. You can enable or disable longer IDs per IAM user and IAM role. By default, an IAM user or role defaults to the same settings as the root user.

Depending on when you created your account, supported resource types may default to using longer IDs. However, you can opt out of using longer IDs until the deadline date for that resource type. For more information, see [Longer EC2 and EBS Resource IDs](#) in the *Amazon EC2 FAQs*.

Resources created with longer IDs are visible to all IAM users and IAM roles, regardless of individual settings and provided that they have permissions to view the relevant resource types.

Topics

- [Working with Longer IDs \(p. 745\)](#)
- [Controlling Access to Longer ID Settings \(p. 748\)](#)

Working with Longer IDs

You can view and modify the longer ID settings for yourself, or for a different IAM user, IAM role, or the root user of the account.

Topics

- [Viewing and Modifying Your Longer ID Settings \(p. 745\)](#)
- [Viewing and Modifying Longer ID Settings for Users or Roles \(p. 747\)](#)

Viewing and Modifying Your Longer ID Settings

You can use the Amazon EC2 console or the AWS CLI to view the resource types that support long IDs, and enable or disable the longer ID format for yourself. The procedures in this section apply to the IAM user or IAM role that's logged into the console or that makes the request; they do not apply to the entire AWS account.

To view and modify the longer ID settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for which you want to view or change the longer ID settings. Settings are not shared between regions.
3. From the dashboard, under **Account Attributes**, choose **Resource ID length management**. The resource types that support longer IDs are listed. The date at which you're automatically switched over to using longer IDs for each resource type is displayed in the **Deadline** column.
4. To enable the longer ID format for a supported resource type, choose the check box for the **Use Longer IDs** column. To disable the longer ID format, clear the check box.

Important

If you're logged in as the root user, these settings apply to the entire account, unless an IAM user or role logs in and explicitly overrides these settings for themselves. Resources created with longer IDs are visible to all IAM users, regardless of individual settings and provided that they have permissions to view the relevant resource types.

To view and modify longer ID settings using the AWS CLI

To view the longer ID settings of all supported resources, use the [describe-id-format](#) AWS CLI command:

```
aws ec2 describe-id-format

{
  "Statuses": [
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "instance"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": true,
      "Resource": "reservation"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "volume"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "snapshot"
    }
  ]
}
```

The results apply to the IAM user, IAM role, or root user that makes the request; they do not apply to the entire AWS account. The results above indicate that the `instance`, `reservation`, `volume`, and `snapshot` resource types can be enabled or disabled for longer IDs; the `reservation` resource is already enabled. The `Deadline` field indicates the date (in UTC) at which you will be automatically switched over to using longer IDs for that resource. If a deadline date is not yet available, this value is not returned.

To enable longer IDs for a specified resource, use the [modify-id-format](#) AWS CLI command:

```
aws ec2 modify-id-format --resource resource-type --use-long-ids
```

To disable longer IDs for a specified resource, use the [modify-id-format](#) AWS CLI command:

```
aws ec2 modify-id-format --resource resource-type --no-use-long-ids
```

If you're using these actions as the root user, then these settings apply to the entire account, unless an IAM user or role explicitly overrides these settings for themselves. These commands are per-region only. To modify the settings for other regions, use the `--region` parameter in the command.

Note

In the 2015-10-01 version of the Amazon EC2 API, if you call `describe-id-format` or `modify-id-format` using IAM role credentials, the results apply to the entire AWS account, and not the specific IAM role. In the current version of the Amazon EC2 API, the results apply to the IAM role only.

Alternatively, you can use the following commands:

To describe the ID format

- [DescribeIdFormat](#) (Amazon EC2 API)
- [Get-EC2IdFormat](#) (AWS Tools for Windows PowerShell)

To modify the ID format

- [ModifyIdFormat](#) (Amazon EC2 API)
- [Edit-EC2IdFormat](#) (AWS Tools for Windows PowerShell)

Viewing and Modifying Longer ID Settings for Users or Roles

You can view supported resource types and enable the longer ID settings for a specific IAM user, IAM role, or the root user of your account by using the [describe-identity-id-format](#) and [modify-identity-id-format](#) AWS CLI commands. To use these commands, you must specify the ARN of an IAM user, IAM role, or root account user in the request. For example, the ARN of the role 'EC2Role' in account 123456789012 is `arn:aws:iam::123456789012:role/EC2Role`. For more information, see [Principal](#) in the *IAM User Guide*.

To view the longer ID settings of all supported resources for a specific IAM user or IAM role, use the following AWS CLI command:

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal
```

To enable the longer ID settings for a resource type for a specific IAM user or IAM role, use the following AWS CLI command:

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --resource resource-type --use-long-ids
```

These commands apply to the ARN specified in the request, they do not apply to the IAM user, IAM role, or root user that made the request.

You can enable the longer ID settings for all IAM users, IAM roles, and the root user of your account by using the following AWS CLI command:

```
aws ec2 modify-identity-id-format --principal-arn all --resource resource-type --use-long-ids
```

Alternatively, you can use the following commands:

To describe the ID format

- [DescribeIdentityIdFormat](#) (Amazon EC2 API)
- [Get-EC2IdentityIdFormat](#) (AWS Tools for Windows PowerShell)

To modify the ID format

- [ModifyIdentityIdFormat](#) (Amazon EC2 API)

- [Edit-EC2IdentityIdFormat](#) (AWS Tools for Windows PowerShell)

Controlling Access to Longer ID Settings

By default, IAM users and roles do not have permission to use the `ec2:DescribeIdFormat`, `ec2:DescribeIdentityIdFormat`, `ec2:ModifyIdFormat`, and `ec2:ModifyIdentityIdFormat` actions unless they're explicitly granted permission through their associated IAM policies. For example, an IAM role may have permission to use all Amazon EC2 actions through an `"Action": "ec2:*"` element in the policy statement.

To prevent IAM users and roles from viewing or modifying the longer resource ID settings for themselves or other users and roles in your account, ensure that the IAM policy contains the following statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:ModifyIdFormat",
        "ec2:DescribeIdFormat",
        "ec2:ModifyIdentityIdFormat",
        "ec2:DescribeIdentityIdFormat"
      ],
      "Resource": "*"
    }
  ]
}
```

We do not support resource-level permissions for the `ec2:DescribeIdFormat`, `ec2:DescribeIdentityIdFormat`, `ec2:ModifyIdFormat`, and `ec2:ModifyIdentityIdFormat` actions.

Listing and Filtering Your Resources

You can get a list of some types of resource using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include only the resources that match certain criteria.

Topics

- [Advanced Search](#) (p. 748)
- [Listing Resources Using the Console](#) (p. 749)
- [Filtering Resources Using the Console](#) (p. 750)
- [Listing and Filtering Using the CLI and API](#) (p. 751)

Advanced Search

Advanced search allows you to search using a combination of filters to achieve precise results. You can filter by keywords, user-defined tag keys, and predefined resource attributes.

The specific search types available are:

- **Search by keyword**

To search by keyword, type or paste what you're looking for in the search box, and then choose Enter. For example, to search for a specific instance, you can type the instance ID.

- **Search by fields**

You can also search by fields, tags, and attributes associated with a resource. For example, to find all instances in the stopped state:

1. In the search box, start typing `Instance state`. As you type, you'll see a list of suggested fields.
2. Select **Instance State** from the list.
3. Select **Stopped** from the list of suggested values.
4. To further refine your list, select the search box for more search options.

- **Advanced search**

You can create advanced queries by adding multiple filters. For example, you can search by tags and see instances for the Flying Mountain project running in the Production stack, and then search by attributes to see all t2.micro instances, or all instances in us-west-2a, or both.

- **Inverse search**

You can search for resources that do not match a specified value. For example, to list all instances that are not terminated, search by the **Instance State** field, and prefix the Terminated value with an exclamation mark (!).

- **Partial search**

When searching by field, you can also enter a partial string to find all resources that contain the string in that field. For example, search by **Instance Type**, and then type `t2` to find all t2.micro, t2.small or t2.medium instances.

- **Regular expression**

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, search by the Name tag, and then type `^s.*` to see all instances with a Name tag that starts with an 's'. Regular expression search is not case-sensitive.

After you have the precise results of your search, you can bookmark the URL for easy reference. In situations where you have thousands of instances, filters and bookmarks can save you a great deal of time; you don't have to run searches repeatedly.

Combining search filters

In general, multiple filters with the same key field (e.g., tag:Name, search, Instance State) are automatically joined with OR. This is intentional, as the vast majority of filters would not be logical if they were joined with AND. For example, you would get zero results for a search on Instance State=running AND Instance State=stopped. In many cases, you can granulate the results by using complementary search terms on different key fields, where the AND rule is automatically applied instead. If you search for tag: Name:=All values and tag:Instance State=running, you get search results that contain both those criteria. To fine-tune your results, simply remove one filter in the string until the results fit your requirements.

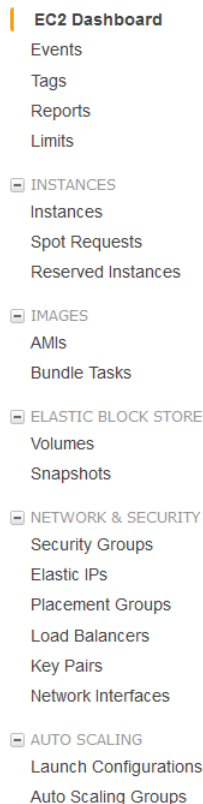
Listing Resources Using the Console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose the option that corresponds to the resource, such as **AMIs** or **Instances**.



3. The page displays all the available resources.

Filtering Resources Using the Console

You can perform filtering and sorting of the most common resource types using the Amazon EC2 console. For example, you can use the search bar on the instances page to sort instances by tags, attributes, or keywords.

You can also use the search field on each page to find resources with specific attributes or values. You can use regular expressions to search on partial or multiple strings. For example, to find all instances that are using the MySG security group, enter `MySG` in the search field. The results will include any values that contain `MySG` as a part of the string, such as `MySG2` and `MySG3`. To limit your results to `MySG` only, enter `\bMySG\b` in the search field. To list all the instances whose type is either `m1.small` or `m1.large`, enter `m1.small|m1.large` in the search field.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, choose **Volumes**.
2. Click on the search box, select **Attachment Status** from the menu, and then select **Detached**. (A detached volume is available to be attached to an instance in the same Availability Zone.)
3. Click on the search box again, select **State**, and then select **Available**.
4. Click on the search box again, select **Availability Zone**, and then select `us-east-1b`.
5. Any volumes that meet this criteria are displayed.

To list public 64-bit Linux AMIs backed by Amazon EBS

1. In the navigation pane, choose **AMIs**.
2. In the **Filter** pane, select **Public images**, **EBS images**, and then your Linux distribution from the **Filter** lists.
3. Enter `x86_64` in the search field.
4. Any AMIs that meet this criteria are displayed.

Listing and Filtering Using the CLI and API

Each resource type has a corresponding CLI command or API request that you use to list resources of that type. For example, you can list Amazon Machine Images (AMI) using `ec2-describe-images` or `DescribeImages`. The response contains information for all your resources.

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. You can specify multiple filter values, and you can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

Note

If you use a tag filter, the response includes the tags for your resources; otherwise, tags may be omitted in the response.

You can also use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches exactly one character. For example, you can use `*database*` as a filter value to get all EBS snapshots that include `database` in the description. If you were to specify `database` as the filter value, then only snapshots whose description equals `database` would be returned. Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards). If a resulting list of resources is long, using an exact string filter may return the response faster.

Tip

Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `*amazon?\` searches for the literal string `*amazon?\`.

For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the [AWS Command Line Interface Reference](#).
- For Windows PowerShell, see the relevant `Get` command in the [AWS Tools for Windows PowerShell Reference](#).
- For the Query API, see the relevant `Describe` API action in the [Amazon EC2 API Reference](#).

Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon EC2 resources, you can optionally assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

Contents

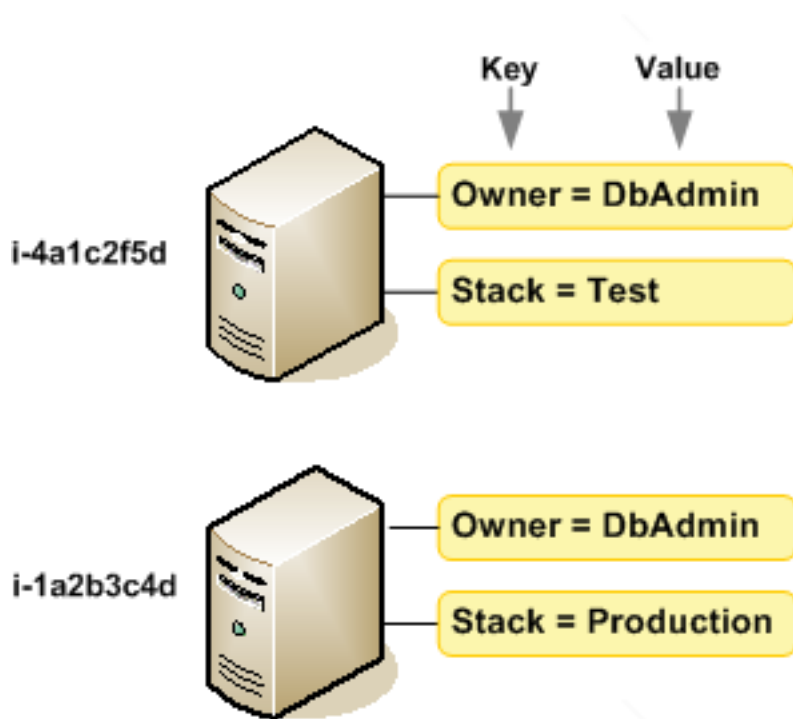
- [Tag Basics \(p. 752\)](#)
- [Tag Restrictions \(p. 753\)](#)

- [Tagging Your Resources for Billing](#) (p. 754)
- [Working with Tags Using the Console](#) (p. 755)
- [Working with Tags Using the CLI or API](#) (p. 760)

Tag Basics

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you've assigned to it. Each tag consists of a key and an optional value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances, one called `Owner` and another called `Stack`. Each of the tags also has an associated value.



Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources.

You can work with tags using the AWS Management Console, the Amazon EC2 command line interface (CLI), the AWS CLI, and the Amazon EC2 API.

You can assign tags only to resources that already exist. You cannot assign tags when you create a resource; for example, when you use the `run-instances` AWS CLI command. When you use the Amazon EC2 console, some resource creation screens enable you to specify tags which are applied immediately after the resource is created. If you add a tag that has the same key as an existing tag on

that resource, the new value overwrites the old value. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set a tag's value to the empty string, but you can't set a tag's value to null.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information about IAM, see [Controlling Access to Amazon EC2 Resources \(p. 498\)](#).

Tag Restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource—50
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.
- If your tagging schema will be used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: `+ - = . _ : / @`.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must use the `DeleteSnapshots` action with the resource identifiers of the snapshots, such as `snap-1234567890abcdef0`. To identify resources by their tags, you can use the `DescribeTags` action to list all of your tags and their associated resources. You can also filter by resource type or tag keys and values. You can't call `DeleteSnapshots` with a filter that specified the tag. For more information about using filters when listing your resources, see [Listing and Filtering Your Resources \(p. 748\)](#).

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources, and some you can only tag using API actions or the command line. The following table lists all Amazon EC2 resources and the tagging restrictions that apply to them, if any. Resources with tagging restrictions of None can be tagged with API actions, the CLI, and the console.

Resource	Tagging support	Tagging restrictions
AMI	Yes	None
Bundle task	No	
Customer gateway	Yes	None
Dedicated Host	No	
DHCP option	Yes	None
EBS volume	Yes	None
Instance store volume	No	
Elastic IP	No	
Instance	Yes	None

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Tagging Your Resources for Billing**

Resource	Tagging support	Tagging restrictions
Internet gateway	Yes	None
Key pair	No	
NAT gateway	No	
Network ACL	Yes	None
Network interface	Yes	None
Placement group	No	
Reserved Instance	Yes	None
Reserved Instance listing	No	
Route table	Yes	None
Spot instance request	Yes	None
Security group - EC2-Classical	Yes	None
Security group - VPC	Yes	None
Snapshot	Yes	None
Subnet	Yes	None
Virtual private gateway	Yes	None
VPC	Yes	None
VPC endpoint	No	
VPC flow log	No	
VPC peering connection	Yes	None
VPN connection	Yes	None

For more information about tagging using the AWS Management Console, see [Working with Tags Using the Console \(p. 755\)](#). For more information about tagging using the API or command line, see [Working with Tags Using the CLI or API \(p. 760\)](#).

Tagging Your Resources for Billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see [Setting Up Your Monthly Cost Allocation Report](#) in *About AWS Account Billing*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Cost Allocation and Tagging](#) in *About AWS Account Billing*.

Note

If you've just enabled reporting, the current month's data will be available for viewing in about 24 hours.

Working with Tags Using the Console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

For ease of use and best results, use Tag Editor in the AWS Management Console, which provides a central, unified way to create and manage your tags. For more information, see [Working with Tag Editor](#) in [Getting Started with the AWS Management Console](#).

Contents

- [Displaying Tags](#) (p. 755)
- [Adding and Deleting Tags on an Individual Resource](#) (p. 756)
- [Adding and Deleting Tags to a Group of Resources](#) (p. 757)
- [Adding a Tag When You Launch an Instance](#) (p. 758)
- [Filtering a List of Resources by Tag](#) (p. 759)

Displaying Tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

To display tags for individual resources

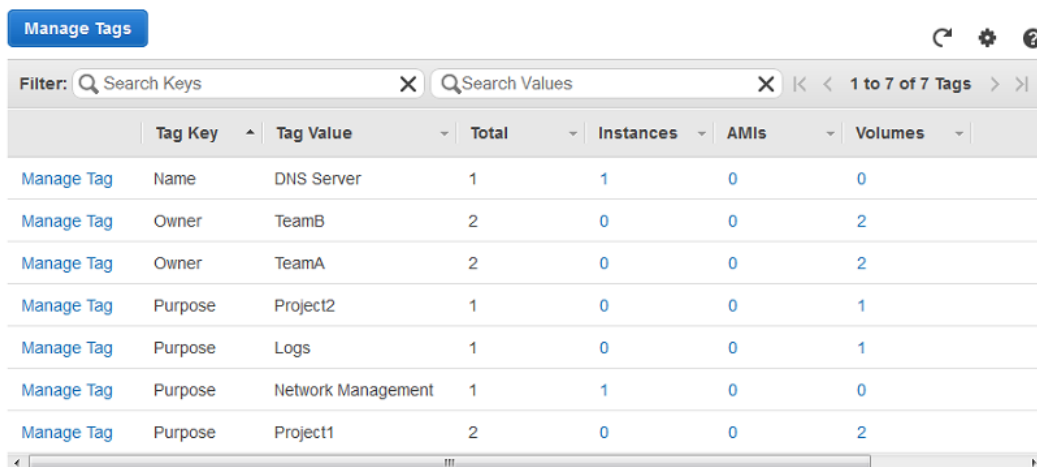
When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags in the **Tags** tab on the details pane.

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags.

- On the **Tags** tab, select **Show Column**. A new column will be added to the console.
- Choose the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

To display tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.



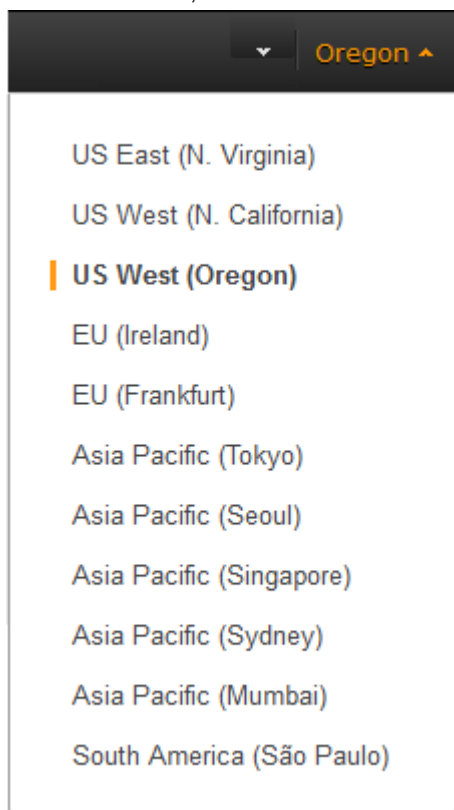
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

Adding and Deleting Tags on an Individual Resource

You can manage tags for an individual resource directly from the resource's page.

To add a tag to an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 743).



3. In the navigation pane, select a resource type (for example, **Instances**).

4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Choose the **Add/Edit Tags** button.
7. In the **Add/Edit Tags** dialog box, specify the key and value for each tag, and then choose **Save**.

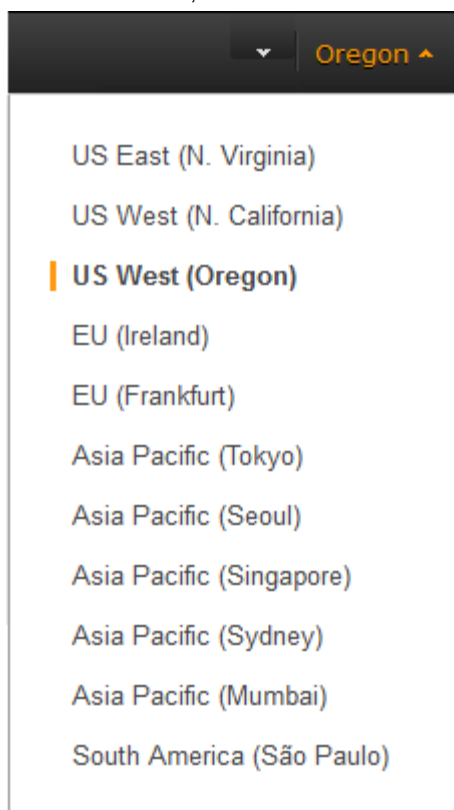
To delete a tag from an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 743).
3. In the navigation pane, choose a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Choose **Add/Edit Tags**, select the **Delete** icon for the tag, and choose **Save**.

Adding and Deleting Tags to a Group of Resources

To add a tag to a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 743).



3. In the navigation pane, choose **Tags**.

4. At the top of the content pane, choose **Manage Tags**.
5. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to add tags to.
6. In the resources list, select the check box next to each resource that you want to add tags to.
7. In the **Key** and **Value** boxes under **Add Tag**, type the tag key and values you want, and then choose **Add Tag**.

Note

If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

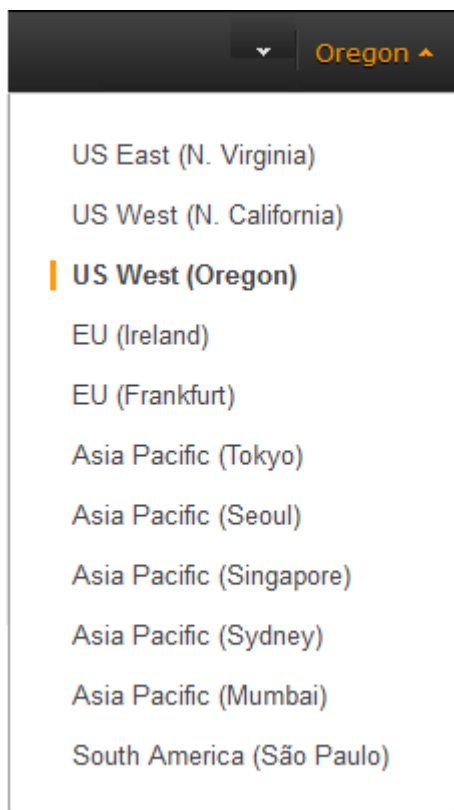
To remove a tag from a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 743\)](#).
3. In the navigation pane, choose **Tags**.
4. At the top of the content pane, choose **Manage Tags**.
5. To view the tags in use, select the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys you want to view, and then choose **Close**.
6. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to remove tags from.
7. In the resource list, select the check box next to each resource that you want to remove tags from.
8. Under **Remove Tag**, type the tag's name in the **Key** box, and then choose **Remove Tag**.

Adding a Tag When You Launch an Instance

To add a tag using the Launch Wizard

1. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations \(p. 743\)](#).



2. Choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Choose the AMI that you want to use and choose **Select**. For more information about selecting an AMI, see [Finding a Linux AMI \(p. 75\)](#).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then choose **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Choose **Next: Tag Instance** when done.
6. On the **Tag Instance** page, specify tags for the instance by providing key and value combinations. Choose **Create Tag** to add more than one tag to your instance. Choose **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Choose **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, choose **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then choose **Launch Instances**.

Filtering a List of Resources by Tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag

1. Display a column for the tag as follows:
 - a. Select one of the resources.

- b. Select the **Tags** tab in the details pane.
- c. Locate the tag in the list and choose **Show Column**.
2. Choose the filter icon in the top right corner of the column for the tag to display the filter list.
3. Select the tag values, and then choose **Apply Filter** to filter the results list.

Note

For more information about filters see [Listing and Filtering Your Resources](#) (p. 748).

Working with Tags Using the CLI or API

Use the following to add, update, list, and delete the tags for your resources. The corresponding documentation provides examples.

Task	AWS CLI	AWS Tools for Windows PowerShell	API Action
Add or overwrite one or more tags.	create-tags	New-EC2Tag	CreateTags
Delete one or more tags.	delete-tags	Remove-EC2Tag	DeleteTags
Describe one or more tags.	describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. The following examples demonstrate how to filter your instances using tags with the [describe-instances](#) command.

Example 1: Describe instances with the specified tag key

The following command describes the instances with a Stack tag, regardless of the value of the tag.

```
aws ec2 describe-instances --filters Name=tag-key,Values=Stack
```

Example 2: Describe instances with the specified tag

The following command describes the instances with the tag Stack=production.

```
aws ec2 describe-instances --filters Name=tag:Stack,Values=production
```

Example 3: Describe instances with the specified tag value

The following command describes the instances with a tag with the value production, regardless of the tag key.

```
aws ec2 describe-instances --filters Name=tag-value,Values=production
```

Important

If you describe resources without using a tag filter, the results may not return the tags for your resources. To ensure that tags are returned in results, we recommend that you either describe tags (and use a resource filter if necessary), or describe your resources and use one or more tag filters.

Amazon EC2 Service Limits

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default limits on these

resources on a per-region basis. For example, there is a limit on the number of instances that you can launch in a region. Therefore, when you launch an instance in the US West (Oregon) Region, the request must not cause your usage to exceed your current instance limit in that region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

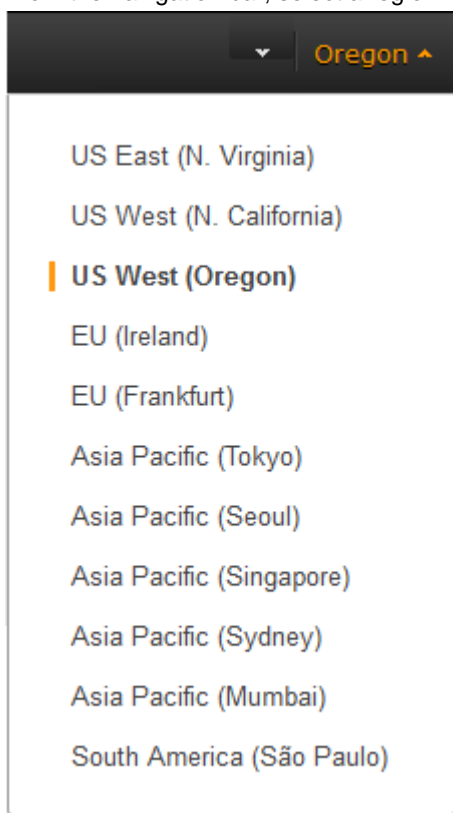
For more information about the limits for other services, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

Viewing Your Current Limits

Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.

To view your current limits

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region.



3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. The **Current Limit** column displays the current maximum for that resource for your account.

Requesting a Limit Increase

Use the **Limits** page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2 or Amazon VPC, on a per-region basis.

To request a limit increase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region.
3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. Choose **Request limit increase**.
5. Complete the required fields on the limit increase form. We'll respond to you using the contact method that you specified.

Amazon EC2 Usage Reports

The usage reports provided by Amazon EC2 enable you to analyze the usage of your instances in depth. The data in the usage reports is updated multiple times each day. You can filter the reports by AWS account, region, Availability Zone, operating system, instance type, purchasing option, tenancy, and tags.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see [Pay Bills for Multiple Accounts with Consolidated Billing](#).

Topics

- [Available Reports \(p. 762\)](#)
- [Getting Set Up for Usage Reports \(p. 762\)](#)
- [Granting IAM Users Access to the Amazon EC2 Usage Reports \(p. 764\)](#)
- [Instance Usage Report \(p. 764\)](#)
- [Reserved Instance Utilization Reports \(p. 767\)](#)

Available Reports

You can generate the following reports:

- [Instance usage report \(p. 764\)](#). This report covers your usage of On-Demand instances, Spot instances, and Reserved Instances.
- [Reserved Instances utilization report \(p. 767\)](#). This report covers the usage of your capacity reservation.

To access the reports, open the AWS Management Console. In the navigation pane, choose **Reports** then choose the report you'd like to view.

Getting Set Up for Usage Reports

Before you begin, enable detailed billing reports with resources and tags as shown in the following procedure. After you complete this procedure, we'll start collecting usage data for your instances.

If you've already enabled detailed billing reports, you can access the usage data that we've been collecting since you enabled them.

Important

To complete these procedures, you must log in using your AWS account credentials. You can't complete these procedures if you log in using IAM user credentials.

To enable detailed billing reports

1. Select an existing Amazon S3 bucket to receive your usage data. Be sure to manage access to this bucket as it contains your billing data. (We don't require that you keep these files; in fact, you can delete them immediately if you don't need them.) If you don't have a bucket, create one as follows:
 - a. Open the Amazon S3 console.
 - b. Select **Create Bucket**.
 - c. In the **Create a Bucket** dialog box, enter a name for your bucket (for example, *username-ec2-usage-data*), select a region, and then choose **Create**. For more information about the requirements for bucket names, see [Creating a Bucket](#) in the *Amazon Simple Storage Service Console User Guide*.
2. Open the Billing and Cost Management console at <https://console.aws.amazon.com/billing/home?#>.
3. Choose **Preferences** in the navigation pane.
4. Select **Receive Billing Reports**.
5. Specify the name of your Amazon S3 bucket in **Save to S3 Bucket**, and then select **Verify**.
6. Grant AWS permission to publish usage data to your Amazon S3 bucket.
 - a. Under **Receive Billing Reports**, choose **sample policy**. Copy the sample policy. Notice that the sample policy uses the bucket name you specified.
 - b. Open the Amazon S3 console in another browser tab. Select your bucket, choose **Properties**, and then expand **Permissions**. In the **Permissions** section, choose **Add bucket policy**. Paste the sample policy into the text area and choose **Save**. In the **Permissions** section, choose **Save**.
 - c. Return to the browser tab with the sample policy and choose **Done**.
7. Under **Report**, select **Detailed billing report with resources and tags**.
8. Choose **Save preferences**.

Note

It can take up to a day before you can see your data in the reports.

You can categorize your instances using tags. After you tag your instances, you must enable reporting on these tags.

To enable usage reporting by tag

1. Tag your instances. For best results, ensure that you add each tag you plan to use for reporting to each of your instances. For more information about how to tag an instance, see [Tagging Your Amazon EC2 Resources](#) (p. 751).
2. Open the Billing and Cost Management console at <https://console.aws.amazon.com/billing/home?#>.
3. Select **Preferences** in the navigation pane.
4. Under **Report**, choose **Manage report tags**.
5. The page displays the list of tags that you've created. Select the tags that you'd like to use to filter or group your instance usage data, and then click **Save**. We automatically exclude any tags that you don't select from your instance usage report.

Note

We apply these changes only to the data for the current month. It can take up to a day for these changes to take effect.

Granting IAM Users Access to the Amazon EC2 Usage Reports

By default, IAM users can't access the Amazon EC2 usage reports. You must create an IAM policy that grants IAM users permission to access these reports.

The following policy allows users to view both Amazon EC2 usage reports.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:*",
    "Resource": "*"
  }]
}
```

The following policy allows users to view the instance usage report.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewInstanceUsageReport",
    "Resource": "*"
  }]
}
```

The following policy allows users to view the Reserved Instances utilization report.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",
    "Resource": "*"
  }]
}
```

For more information, see [Permissions and Policies](#) in the *IAM User Guide*.

Instance Usage Report

You can use the instance usage report to view your instance usage and cost trends. You can see your usage data in either instance hours or cost. You can choose to see hourly, daily and monthly

aggregates of your usage data. You can filter or group the report by region, Availability Zone, instance type, AWS account, platform, tenancy, purchase option, or tag. After you configure a report, you can bookmark it so that it's easy to get back to later.

Here's an example of some of the questions that you can answer by creating an instance usage report:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?

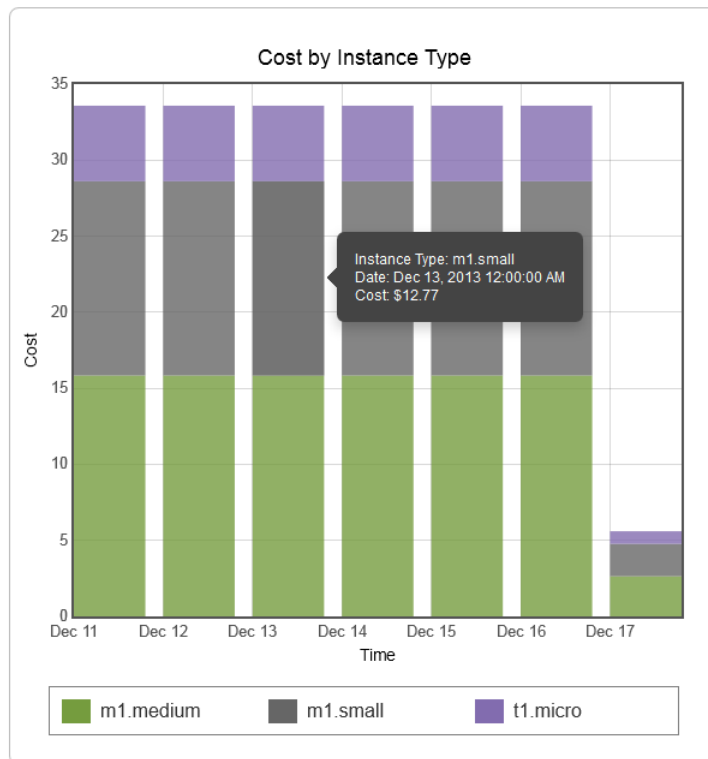
Topics

- [Report Formats \(p. 765\)](#)
- [Viewing Your Instance Usage \(p. 766\)](#)
- [Bookmarking a Customized Report \(p. 767\)](#)
- [Exporting Your Usage Data \(p. 767\)](#)

Report Formats

We display the usage data that you request as both a graph and a table.

For example, the following graph displays cost by instance type. The key for the graph indicates which color represents which instance type. To get detailed information about a segment of a bar, hover over it.



The corresponding table displays one column for each instance type. Notice that we include a color band in the column head that is the same color as the instance type in the graph.

Amazon Elastic Compute Cloud User Guide for Linux Instances Instance Usage

Time (UTC)	m1.medium	m1.small	t1.micro
12/11/13	\$15.84	\$12.77	\$4.97
12/12/13	\$15.84	\$12.77	\$4.97
12/13/13	\$15.84	\$12.77	\$4.97
12/14/13	\$15.84	\$12.77	\$4.97
12/15/13	\$15.84	\$12.77	\$4.97
12/16/13	\$15.84	\$12.77	\$4.97
12/17/13	\$2.64	\$2.13	\$0.83
Total	\$97.68	\$78.75	\$30.65

Viewing Your Instance Usage

The following procedures demonstrate how to generate usage reports using some of the capabilities we provide.

Before you begin, you must get set up. For more information, see [Getting Set Up for Usage Reports](#) (p. 762).

To filter and group your instance usage by instance type

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Reports** and then select **EC2 Instance Usage Report**.
3. Select an option for **Unit**. To view the time that your instances have been running, in hours, select *Instance Hours*. To view the cost of your instance usage, select *Cost*.
4. Select options for **Granularity** and **Time range**.
 - To view the data summarized for each hour in the time range, select *Hourly* granularity. You can select a time range of up to 2 days when viewing hourly data.
 - To view the data summarized for each day in the time range, select *Daily* granularity. You can select a time range of up to 2 months when viewing daily data.
 - To view the data summarized for each month in the time range, select *Monthly* granularity.
5. In the **Filter** list, select *Instance Type*. In the **Group by** list, select *Instance Type*.
6. In the filter area, select one or more instance types and then select **Update Report**. The filters you specify appear under **Applied Filters**.

[EC2 Management Console](#) > [Reports](#) > EC2 Instance Usage

Instance Usage Reports

Download ▾

Granularity: **Daily** ▾

Time range: **Last 7 Days** ▾

Unit: **Cost** ▾

Filter: **Select** ▾

Group by: **Instance Type** ▾

Applied Filters: [Instance Type: t1.micro; m1.small; m1.medium;](#)

[Clear All Filters](#)

[Update Report](#)

Notice that you can return to the Amazon EC2 console by choosing either **Reports** or **EC2 Management Console** at the top of the page.

To group your instance usage based on tags

1. Open the Instance Usage Reports page.
2. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
3. Select options for **Granularity** and **Time range**.
 - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
 - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
 - To view the data summarized for each month in the time range, select `Monthly` granularity.
4. In the **Group by** list, select **Tag**.
5. Choose the **Key Name** box, select a name from the list, and then choose **Update Report**. If there are no items in this list, you must enable usage reporting by tag. For more information, see [To enable usage reporting by tag \(p. 763\)](#).

The screenshot shows the 'Instance Usage Reports' interface. At the top, it says 'Instance Usage Reports'. Below that, there are several controls: 'Granularity' set to 'Daily', 'Time range' set to 'Last 14 Days', and 'Unit' set to 'Instance Hours'. There is also a 'Filter' dropdown set to 'Select', a 'Group by' dropdown set to 'Tag', and a 'Key Name' text box containing 'Project'. At the bottom left, it says 'Applied Filters: None'. At the bottom right, there are two buttons: 'Clear All Filters' and 'Update Report'.

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=filter_list`.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Reserved Instance Utilization Reports

The Reserved Instance utilization report describes the utilization over time of each group (or *bucket*) of Amazon EC2 Reserved Instances that you own. Each bucket has a unique combination of region,

Availability Zone, instance type, tenancy, offering type, and platform. You can specify the time range that the report covers, from a custom range to weeks, months, a year, or three years. The available data depends on when you enable detailed billing reports for the account (see [Getting Set Up for Usage Reports](#) (p. 762)). The Reserved Instance utilization report compares the Reserved Instance prices paid for instance usage in the bucket with On-Demand prices and shows your savings for the time range covered by the report.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see [Pay Bills for Multiple Accounts with Consolidated Billing](#).

Note

The Reserved Instance buckets aggregate Reserved Instances across EC2-VPC and EC2-Classic network platform types in the same way that your bill is calculated. Additionally, Reserved Instances in a bucket may have different upfront and hourly prices.

Here are examples of some of the questions that you can answer using the Reserved Instance utilization report:

- How well am I utilizing my Reserved Instances?
- Are my Reserved Instances helping me save money?

For information about Reserved Instances, see [Reserved Instances](#) (p. 194).

Before you begin, you must get set up. For more information, see [Getting Set Up for Usage Reports](#) (p. 762).

Topics

- [Getting to Know the Report](#) (p. 768)
- [Viewing Your Reserved Instance Utilization](#) (p. 770)
- [Bookmarking a Customized Report](#) (p. 770)
- [Exporting Your Usage Data](#) (p. 771)
- [Options Reference](#) (p. 771)

Getting to Know the Report

The Reserved Instance utilization report displays your requested utilization data in graph and table formats.

To access the report, open the AWS Management Console. In the navigation pane, choose **Reports** and then select **EC2 Reserved Instance Usage Report**.

The report aggregates Reserved Instance usage data for a given period by bucket. In the report, each row in the table represents a bucket and provides the following metrics:

- **Count**—The highest number of Reserved Instances owned at the same time during the period of the report.
- **Usage Cost**—The total Reserved Instance usage fees applied to instance usage covered by the Reserved Instance bucket.
- **Total Cost**—The usage cost plus the amortized upfront fee for the usage period associated with the Reserved Instance bucket.

Note

If the bucket contains a Reserved Instance that you sold in the Reserved Instance Marketplace and that Reserved Instance was active at any point during the period of the report, the total cost of the bucket might be inflated and your savings might be underestimated.

- **Savings**—The difference between what your usage for the period would have cost at On-Demand prices and what it actually cost using Reserved Instances (Total Cost).
- **Average Utilization**—The average hourly utilization rate for the Reserved Instance bucket over the period.
- **Maximum Utilization**—The highest utilization rate of any hour during the period covered by the report.

For each row—or Reserved Instance bucket—in the table, the graph represents data based on your selected **Show** metric over the selected **Time range** for the report. Each point in the graph represents a metric at a point in time. For information about report options, see [Options Reference \(p. 771\)](#).

A color band at the edge of each selected row in the table corresponds to a report line in the graph. You can show a row in the graph by selecting the checkbox at the beginning of the row.

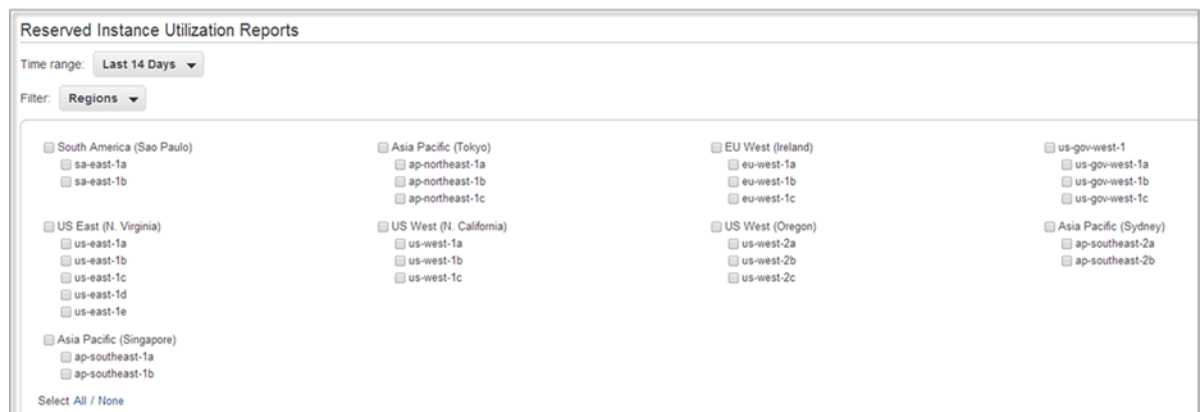
By default, the Reserved Instance utilization report returns data over the last 14 days for all Reserved Instance buckets. The graph shows the average utilization for the first five buckets in the table. You can customize the report graph to show different utilization (average utilization, maximum utilization) or cost (total cost, usage cost) data over a period ranging from 7 days to weeks, months, or years.

Customizing the Report

You can customize the Reserved Instance utilization report with **Time range** and **Filter** options.

Time range provides a list of common relative time ranges, ranging from **Last 7 Days** to **Last 3 Years**. Select the time range that works best for your needs, and then click **Update Report** to apply the change. To apply a time range that is not on the list, select **Custom** and enter the start date and end date for which you want to run the report.

Filter lets you scope your Reserved Instance utilization report by one or more of the following Reserved Instance qualities: region, instance type, accounts, platforms, tenancy, and offering types. For example, you can filter by region or by specific Availability Zones in a region, or both. To filter by region, select **Regions**, then select the regions and Availability Zones you want to include in the report, and choose **Update Report**.

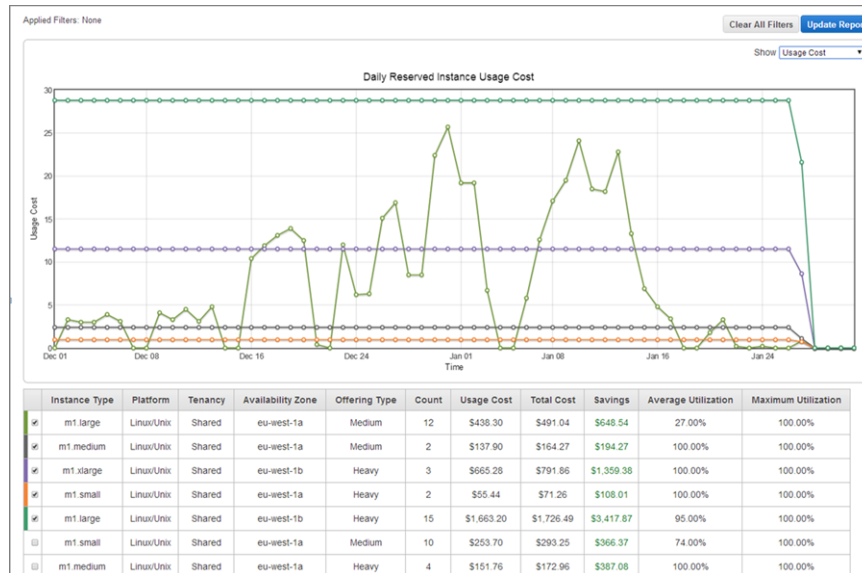


The report will return all results if no filter is applied.

For information about report options, see [Options Reference \(p. 771\)](#).

Viewing Your Reserved Instance Utilization

In this section, we will highlight aspects of your Reserved Instance utilization that the graph and table capture. For the purposes of this discussion, we'll use the following report, which is based on test data.



This Reserved Instance utilization report displays the average utilization of Reserved Instances in the last two months. This report reveals the following information about the account's Reserved Instances and how they have been utilized.

- Average Utilization

Most of the Reserved Instances in the table were utilized well. Standouts were the two m1.medium medium utilization Reserved Instances (row 2), which were utilized all the time at 100% average utilization, and the m1.xlarge (row 3) and m1.small (row 4) heavy utilization Reserved Instances, which also were utilized all the time. In contrast, the high-count heavy utilization Reserved Instances (row 5) had lower average utilization rates.

It is also worth noting that the 12 m1.large medium utilization Reserved Instances (row 1) were utilized on average only 27 percent of the time.

- Maximum Utilization

At some point during the two-month period, all of the Reserved Instances were used 100 percent.

- Savings

All across the board, the report shows that for this test account, using Reserved Instances instead of On-Demand instances results in savings for the account owner.

- Question

Does the account have too many m1.large medium utilization Reserved Instances (row 1)?

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=filter_list`.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Options Reference

Use the **Show** options to specify the metric to be displayed by the report graph.

- Average Utilization

Shows the average of the utilization rates for each hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.

- Maximum Utilization

Shows the highest of the utilization rates of any hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.

- Total Cost

Shows the usage cost plus the amortized portion of the upfront cost of the Reserved Instances in the bucket over the period for which the report is generated.

- Usage Cost

Shows the total cost based on hourly fees for a selected bucket of Reserved Instances.

Use **Time range** to specify the period on which the report will be based.

Note

All times are specified in UTC time.

- Last 7 Days

Shows data for usage that took place during the current and previous six calendar days. Can be used with daily or monthly granularities.

- Last 14 Days

Shows data for usage that took place during the current and previous 13 calendar days. Can be used with daily or monthly granularities.

- This Month

Shows data for usage that took place during the current calendar month. Can be used with daily or monthly granularities.

- Last 3 Months

Shows data for usage that took place during the current and previous two calendar months. Can be used with daily or monthly granularities.

- Last 6 Months

Shows data for usage that took place during the current and previous five calendar months. Can be used with monthly granularities.

- Last 12 Months

Shows data for usage that took place during the current and previous 11 calendar months. Can be used with monthly granularity.

- This Year

Shows data for usage that took place during the current calendar year. Can be used with monthly granularity.

- Last 3 Years

Shows data for usage that took place during the current and previous two calendar years. Can be used with monthly granularity.

- Custom

Shows data for the time range for the entered **Start** and **End** dates specified in the following format: mm/dd/yyyy. Can be used with hourly, daily, or monthly granularities, but you can only specify a maximum time range of two days for hourly data, two months for daily data, and three years for monthly data.

Use **Filter** to scope the data displayed in the report.

- Regions
- Instance Type
- Accounts
- Platforms
- Tenancy
- Offering Types

Troubleshooting Instances

The following documentation can help you troubleshoot problems that you might have with your instance.

Contents

- [What To Do If An Instance Immediately Terminates \(p. 773\)](#)
- [Troubleshooting Connecting to Your Instance \(p. 774\)](#)
- [Troubleshooting Stopping Your Instance \(p. 779\)](#)
- [Troubleshooting Terminating \(Shutting Down\) Your Instance \(p. 781\)](#)
- [Troubleshooting Instance Recovery Failures \(p. 781\)](#)
- [Troubleshooting Instances with Failed Status Checks \(p. 782\)](#)
- [Troubleshooting Instance Capacity \(p. 805\)](#)
- [Getting Console Output and Rebooting Instances \(p. 805\)](#)
- [My Instance is Booting from the Wrong Volume \(p. 807\)](#)

For additional help with Windows instances, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

You can also search for answers and post questions on the [Amazon EC2 forum](#).

What To Do If An Instance Immediately Terminates

After you launch an instance, we recommend that you check its status to confirm that it goes from the `pending` state to the `running` state, not the `terminated` state.

The following are a few reasons why an instance might immediately terminate:

- You've reached your EBS volume limit. For information about the volume limit, and to submit a request to increase your volume limit, see [Request to Increase the Amazon EBS Volume Limit](#).

- An EBS snapshot is corrupt.
- The instance store-backed AMI you used to launch the instance is missing a required part (an image.part.xx file).

Getting the Reason for Instance Termination

You can use the Amazon EC2 console, CLI, or API to get information about the reason that the instance terminated.

To get the reason that an instance terminated using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select your instance.
3. In the **Description** tab, locate the reason next to the label **State transition reason**. If the instance is still running, there's typically no reason listed. If you've explicitly stopped or terminated the instance, the reason is `User initiated shutdown`.

To get the reason that an instance terminated using the command line

1. Use the `describe-instances` command:

```
$ aws ec2 describe-instances --instance-id instance_id
```

2. In the JSON response that's displayed, locate the `StateReason` element. It looks similar to the following example.

```
"StateReason": {  
  "Message": "Client.UserInitiatedShutdown: User initiated shutdown",  
  "Code": "Client.UserInitiatedShutdown"  
},
```

This example response shows the reason code that you'll see after you stop or terminate a running instance. If the instance terminated immediately, you'll see `code` and `message` elements that describe the reason that the instance terminated (for example, `VolumeLimitExceeded`).

Troubleshooting Connecting to Your Instance

The following are possible problems you may have and error messages you may see while trying to connect to your instance.

Contents

- [Error connecting to your instance: Connection timed out \(p. 775\)](#)
- [Error: User key not recognized by server \(p. 776\)](#)
- [Error: Host key not found, Permission denied \(publickey\), or Authentication failed, permission denied \(p. 777\)](#)
- [Error: Unprotected Private Key File \(p. 778\)](#)
- [Error: Server refused our key or No supported authentication methods available \(p. 778\)](#)
- [Error using MindTerm on Safari Browser \(p. 779\)](#)
- [Error Using Mac OS X RDP Client \(p. 779\)](#)
- [Cannot Ping Instance \(p. 779\)](#)

For additional help with Windows instances, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

You can also search for answers and post questions on the [Amazon EC2 forum](#).

Error connecting to your instance: Connection timed out

If you try to connect to your instance and get an error message `Network error: Connection timed out` or `Error connecting to [instance], reason: -> Connection timed out: connect`, try the following:

- Check your security group rules. You need a security group rule that allows inbound traffic from your public IP address on the proper port.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. In the **Description** tab, next to **Security groups**, choose **view rules** to display the list of rules that are in effect.
4. For Linux instances: Verify that there is a rule that allows traffic from your computer to port 22 (SSH).

For Windows instances: Verify that there is a rule that allows traffic from your computer to port 3389 (RDP).

If your security group has a rule that allows inbound traffic from a single IP address, this address may not be static if your computer is on a corporate network or if you are connecting through an Internet service provider (ISP). Instead, specify the range of IP addresses used by client computers. If your security group does not have a rule that allows inbound traffic as described in the previous step, add a rule to your security group. For more information, see [Authorizing Network Access to Your Instances](#) (p. 541).

- [EC2-VPC] Check the route table for the subnet. You need a route that sends all traffic destined outside the VPC (0.0.0.0/0) to the Internet gateway for the VPC.
 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the **Description** tab, write down the values of **VPC ID** and **Subnet ID**.
 3. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 4. In the navigation pane, choose **Internet Gateways**. Verify that there is an Internet gateway attached to your VPC. Otherwise, choose **Create Internet Gateway** and follow the directions to create an Internet gateway, select the Internet gateway, and then choose **Attach to VPC** and follow the directions to attach it to your VPC.
 5. In the navigation pane, choose **Subnets**, and then select your subnet.
 6. On the **Route Table** tab, verify that there is a route with 0.0.0.0/0 as the destination and the Internet gateway for your VPC as the target. Otherwise, choose the ID of the route table (rtb-xxxxxxx) to navigate to the **Routes** tab for the route table, choose **Edit**, **Add another route**, enter 0.0.0.0/0 in **Destination**, select your Internet gateway from **Target**, and then choose **Save**.
- [EC2-VPC] Check the network access control list (ACL) for the subnet. The network ACLs must allow inbound and outbound traffic from your public IP address on the proper port. The default network ACL allows all inbound and outbound traffic.
 1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 2. In the navigation pane, choose **Your VPCs**.
 3. On the **Summary** tab, find **Network ACL**, choose the ID (acl-xxxxxxx), and select the ACL.

4. On the **Inbound Rules** tab, verify that the rules allow traffic from your computer. Otherwise, delete or modify the rule that is blocking traffic from your computer.
 5. On the **Outbound Rules** tab, verify that the rules allow traffic to your computer. Otherwise, delete or modify the rule that is blocking traffic to your computer.
- If your computer is on a corporate network, ask your network administrator whether the internal firewall allows inbound and outbound traffic from your computer on port 22 (for Linux instances) or port 3389 (for Windows instances).

If you have a firewall on your computer, verify that it allows inbound and outbound traffic from your computer on port 22 (for Linux instances) or port 3389 (for Windows instances).

- Check that your instance has a public IP address. If not, you can associate an Elastic IP address with your instance. For more information, see [Elastic IP Addresses \(p. 581\)](#).
- Check the CPU load on your instance; the server may be overloaded. AWS automatically provides data such as Amazon CloudWatch metrics and instance status, which you can use to see how much CPU load is on your instance and, if necessary, adjust how your loads are handled. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 419\)](#).
 - If your load is variable, you can automatically scale your instances up or down using [Auto Scaling](#) and [Elastic Load Balancing](#).
 - If your load is steadily growing, you can move to a larger instance type. For more information, see [Resizing Your Instance \(p. 189\)](#).

Error: User key not recognized by server

If you use SSH to connect to your instance

- Use `ssh -vvv` to get triple verbose debugging information while connecting:

```
#ssh -vvv -i [your key name].pem ec2-user@[public DNS address of your instance].compute-1.amazonaws.com
```

The following sample output demonstrates what you might see if you were trying to connect to your instance with a key that was not recognized by the server:

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-
interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
```

Amazon Elastic Compute Cloud User Guide for Linux Instances

Error: Host key not found, Permission denied (publickey), or

```
debug1: Next authentication method: publickey
debug1: Authentication failed, permission denied
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA
 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

If you use SSH (MindTerm) to connect to your instance

- If Java is not enabled, the server does not recognize the user key. To enable Java, go to [How do I enable Java in my web browser?](#) in the Java documentation.

If you use PuTTY to connect to your instance

- Verify that your private key (.pem) file has been converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see [Connecting to Your Linux Instance from Windows Using PuTTY \(p. 293\)](#).

Note

In PuTTYgen, load your private key file and select **Save Private Key** rather than **Generate**.

- Verify that you are connecting with the appropriate user name for your AMI. Enter the user name in the **Host name** box in the **PuTTY Configuration** window.
 - For an Amazon Linux AMI, the user name is `ec2-user`.
 - For a RHEL5 AMI, the user name is either `root` or `ec2-user`.
 - For an Ubuntu AMI, the user name is `ubuntu`.
 - For a Fedora AMI, the user name is either `fedora` or `ec2-user`.
 - For SUSE Linux, the user name is either `root` or `ec2-user`.
 - Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.
- Verify that you have an inbound security group rule to allow inbound traffic to the appropriate port. For more information, see [Authorizing Network Access to Your Instances \(p. 541\)](#).

Error: Host key not found, Permission denied (publickey), or Authentication failed, permission denied

If you connect to your instance using SSH and get any of the following errors, `Host key not found in [directory]`, `Permission denied (publickey)`, or `Authentication failed, permission denied`, verify that you are connecting with the appropriate user name for your AMI *and* that you have specified the proper private key (.pem) file for your instance. For MindTerm clients, enter the user name in the **User name** box in the **Connect To Your Instance** window.

The appropriate user names are as follows:

- For an Amazon Linux AMI, the user name is `ec2-user`.
- For a RHEL5 AMI, the user name is either `root` or `ec2-user`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

- For SUSE Linux, the user name is either `root` or `ec2-user`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Confirm that you are using the private key file that corresponds to the key pair that you selected when you launched the instance.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select your instance. In the **Description** tab, verify the value of **Key pair name**.
3. If you did not specify a key pair when you launched the instance, you can terminate the instance and launch a new instance, ensuring that you specify a key pair. If this is an instance that you have been using but you no longer have the `.pem` file for your key pair, you can replace the key pair with a new one. For more information, see [Connecting to Your Linux Instance if You Lose Your Private Key](#) (p. 486).

Error: Unprotected Private Key File

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key `.ssh/my_private_key.pem` with file permissions of `0777`, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key. To fix the error, execute the following command, substituting the path for your private key file.

```
$ chmod 0400 .ssh/my_private_key.pem
```

Error: Server refused our key or No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, `Error: Server refused our key` or `Error: No supported authentication methods available`, verify that you are connecting with the appropriate user name for your AMI. Enter the user name in the **User name** box in the **PuTTY Configuration** window.

The appropriate user names are as follows:

- For an Amazon Linux AMI, the user name is `ec2-user`.
- For a RHEL5 AMI, the user name is either `root` or `ec2-user`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

- For SUSE Linux, the user name is either `root` or `ec2-user`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

You should also verify that your private key (`.pem`) file has been correctly converted to the format recognized by PuTTY (`.ppk`). For more information about converting your private key, see [Connecting to Your Linux Instance from Windows Using PuTTY](#) (p. 293).

Error using MindTerm on Safari Browser

If you use MindTerm to connect to your instance, and are using the Safari web browser, you may get the following error:

```
Error connecting to your_instance_ip, reason:  
-> Key exchange failed: Host authentication failed
```

You need to update the browser's security settings to allow the AWS Management Console to run the Java plugin in unsafe mode.

To enable the Java plugin to run in unsafe mode

1. In Safari, keep the Amazon EC2 console open, and choose **Safari, Preferences, Security**.
2. Choose **Plug-in Settings** (or **Manage Website Settings** on older versions of Safari).
3. Choose the **Java** plugin on the left, then locate the AWS Management Console URL in the **Currently Open Websites** list. Select **Run in Unsafe Mode** from its associated list.
4. When prompted, choose **Trust** in the warning dialog. Choose **Done** to return the browser.

Error Using Mac OS X RDP Client

If you are connecting to a Windows Server 2012 R2 instance using the Remote Desktop Connection client from the Microsoft website, you may get the following error:

```
Remote Desktop Connection cannot verify the identity of the computer that you  
want to connect to.
```

Download the Microsoft Remote Desktop app from the Apple iTunes store, and use the app to connect to your instance.

Cannot Ping Instance

The `ping` command is a type of ICMP traffic — if you are unable to ping your instance, ensure that your inbound security group rules allow ICMP traffic for the `Echo Request` message from all sources, or from the computer or instance from which you are issuing the command. If you are unable to issue a `ping` command from your instance, ensure that your outbound security group rules allow ICMP traffic for the `Echo Request` message to all destinations, or to the host that you are attempting to ping.

Troubleshooting Stopping Your Instance

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, there may be an issue with the underlying host computer.

First, try stopping the instance again. If you are using the [stop-instances](#) (AWS CLI) command be sure to use the `--force` option.

If you can't force the instance to stop, you can create an AMI from the instance and launch a replacement instance.

You are not billed for any instance hours while an instance is not in the `running` state.

To create a replacement instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, fill in the following fields and then choose **Create Image**:
 - a. Specify a name and description for the AMI.
 - b. Choose **No reboot**.
5. Launch an instance from the AMI and verify that the instance is working.
6. Select the stuck instance, choose **Actions, Instance State, Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

If you are unable to create an AMI from the instance as described in the previous procedure, you can set up a replacement instance as follows:

To create a replacement instance (if the previous procedure fails)

1. Select the instance, open the **Description** tab, and view the **Block devices** list. Select each volume and write down its volume ID. Be sure to note which volume is the root volume.
2. In the navigation pane, choose **Volumes**. Select each volume for the instance, and choose **Actions, Create Snapshot**.
3. In the navigation pane, choose **Snapshots**. Select the snapshot that you just created, and choose **Actions, Create Volume**.
4. Launch an instance of the same type as the stuck instance (Amazon Linux, Windows, and so on). Note the volume ID and device name of its root volume.
5. In the navigation pane, choose **Instances**, select the instance that you just launched, choose **Actions, Instance State**, and then choose **Stop**.
6. In the navigation pane, choose **Volumes**, select the root volume of the stopped instance, and choose **Actions, Detach Volume**.
7. Select the root volume that you created from the stuck instance, choose **Actions, Attach Volume**, and attach it to the new instance as its root volume (using the device name that you wrote down). Attach any additional non-root volumes to the instance.
8. In the navigation pane, choose **Instances** and select the replacement instance. Choose **Actions, Instance State, Start**. Verify that the instance is working.
9. Select the stuck instance, choose **Actions, Instance State, Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

If you're unable to complete these procedures, you can post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the instance ID and describe the steps that you've already taken.

Troubleshooting Terminating (Shutting Down) Your Instance

You are not billed for any instance hours while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Delayed Instance Termination

If your instance remains in the `shutting-down` state longer than a few minutes, it might be delayed due to shutdown scripts being run by the instance.

Another possible cause is a problem with the underlying host computer. If your instance remains in the `shutting-down` state for several hours, Amazon EC2 treats it as a stuck instance and forcibly terminates it.

If it appears that your instance is stuck terminating and it has been longer than several hours, post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the instance ID and describe the steps that you've already taken.

Terminated Instance Still Displayed

After you terminate an instance, it remains visible for a short while before being deleted. The status shows as `terminated`. If the entry is not deleted after several hours, contact Support.

Automatically Launch or Terminate Instances

If you terminate all your instances, you may see that we launch a new instance for you. If you launch an instance, you may see that we terminate one of your instances. If you stop an instance, you may see that we terminate the instance and launch a new instance. Generally, these behaviors mean that you've used Auto Scaling or Elastic Beanstalk to scale your computing resources automatically based on criteria that you've defined.

For more information, see the [Auto Scaling User Guide](#) or the [AWS Elastic Beanstalk Developer Guide](#).

Troubleshooting Instance Recovery Failures

The following issues can cause automatic recovery of your instance to fail:

- Temporary, insufficient capacity of replacement hardware.
- The instance has an attached instance store storage, which is an unsupported configuration for automatic instance recovery.
- There is an ongoing Service Health Dashboard event that prevented the recovery process from successfully executing. Refer to <http://status.aws.amazon.com> for the latest service availability information.
- The instance has reached the maximum daily allowance of three recovery attempts.

The automatic recovery process will attempt to recover your instance for up to three separate failures per day. If the instance system status check failure persists, we recommend that you manually start and stop the instance. For more information, see [Stop and Start Your Instance](#) (p. 299).

Your instance may subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause for the original system status check failure.

Troubleshooting Instances with Failed Status Checks

Topics

- [Initial Steps](#) (p. 782)
- [Retrieving System Logs](#) (p. 783)
- [Troubleshooting System Log Errors for Linux-Based Instances](#) (p. 783)
- [Out of memory: kill process](#) (p. 784)
- [ERROR: mmu_update failed \(Memory management update failed\)](#) (p. 785)
- [I/O error \(Block device failure\)](#) (p. 786)
- [IO ERROR: neither local nor remote disk \(Broken distributed block device\)](#) (p. 787)
- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\)](#) (p. 788)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\)](#) (p. 789)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\)](#) (p. 789)
- [ERROR Invalid kernel \(EC2 incompatible kernel\)](#) (p. 791)
- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\)](#) (p. 792)
- [fsck: No such file or directory while trying to open... \(File system not found\)](#) (p. 793)
- [General error mounting filesystems \(Failed mount\)](#) (p. 794)
- [VFS: Unable to mount root fs on unknown-block \(Root filesystem mismatch\)](#) (p. 796)
- [Error: Unable to determine major/minor number of root device... \(Root file system/device mismatch\)](#) (p. 797)
- [XENBUS: Device with no driver...](#) (p. 798)
- [... days without being checked, check forced \(File system check required\)](#) (p. 798)
- [fsck died with exit status... \(Missing device\)](#) (p. 799)
- [GRUB prompt \(grubdom>\)](#) (p. 800)
- [Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. \(Hard-coded MAC address\)](#) (p. 802)
- [Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. \(SELinux misconfiguration\)](#) (p. 803)
- [XENBUS: Timeout connecting to devices \(Xenbus timeout\)](#) (p. 804)

Initial Steps

If your instance fails a status check, first determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, follow these steps:

To investigate impaired instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.

3. In the details pane, choose **Status Checks** to see the individual results for all **System Status Checks** and **Instance Status Checks**.

If a system status check has failed, you can try one of the following options:

- Create an instance recovery alarm. For more information, see [Create Alarms That Stop, Terminate, or Recover an Instance](#) in the *Amazon CloudWatch Developer Guide*.
- For an instance using an Amazon EBS-backed AMI, stop and restart the instance.
- For an instance using an instance-store backed AMI, terminate the instance and launch a replacement.
- Wait for Amazon EC2 to resolve the issue.
- Post your issue to the [Amazon EC2 forum](#).
- Retrieve the system log and look for errors.

Retrieving System Logs

If an instance status check fails, you can reboot the instance and retrieve the system logs. The logs may reveal an error that can help you troubleshoot the issue. Rebooting clears unnecessary information from the logs.

To reboot an instance and retrieve the system log

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select your instance.
3. Choose **Actions**, **Instance State**, **Reboot**. It may take a few minutes for your instance to reboot.
4. Verify that the problem still exists; in some cases, rebooting may resolve the problem.
5. When the instance is in the `running` state, choose **Actions**, **Instance Settings**, **Get System Log**.
6. Review the log that appears on the screen, and use the list of known system log error statements below to troubleshoot your issue.
7. If your experience differs from our check results, or if you are having an issue with your instance that our checks did not detect, choose **Submit feedback** on the **Status Checks** tab to help us improve our detection tests.
8. If your issue is not resolved, you can post your issue to the [Amazon EC2 forum](#).

Troubleshooting System Log Errors for Linux-Based Instances

For Linux-based instances that have failed an instance status check, such as the instance reachability check, verify that you followed the steps above to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error .

Memory Errors

- [Out of memory: kill process](#) (p. 784)
- [ERROR: mmu_update failed \(Memory management update failed\)](#) (p. 785)

Device Errors

- [I/O error \(Block device failure\)](#) (p. 786)

- [IO ERROR: neither local nor remote disk \(Broken distributed block device\) \(p. 787\)](#)

Kernel Errors

- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 788\)](#)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\) \(p. 789\)](#)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\) \(p. 789\)](#)
- [ERROR Invalid kernel \(EC2 incompatible kernel\) \(p. 791\)](#)

File System Errors

- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 792\)](#)
- [fsck: No such file or directory while trying to open... \(File system not found\) \(p. 793\)](#)
- [General error mounting filesystems \(Failed mount\) \(p. 794\)](#)
- [VFS: Unable to mount root fs on unknown-block \(Root filesystem mismatch\) \(p. 796\)](#)
- [Error: Unable to determine major/minor number of root device... \(Root file system/device mismatch\) \(p. 797\)](#)
- [XENBUS: Device with no driver... \(p. 798\)](#)
- [... days without being checked, check forced \(File system check required\) \(p. 798\)](#)
- [fsck died with exit status... \(Missing device\) \(p. 799\)](#)

Operating System Errors

- [GRUB prompt \(grubdom>\) \(p. 800\)](#)
- [Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. \(Hard-coded MAC address\) \(p. 802\)](#)
- [Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. \(SELinux misconfiguration\) \(p. 803\)](#)
- [XENBUS: Timeout connecting to devices \(Xenbus timeout\) \(p. 804\)](#)

Out of memory: kill process

An out of memory error is indicated by a system log entry similar to the one shown below.

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879  
or a child  
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-  
rss:101196kB, file-rss:204kB
```

Potential Cause

Exhausted memory

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following:

Amazon Elastic Compute Cloud
User Guide for Linux Instances
ERROR: mmu_update failed
(Memory management update failed)

For this instance type	Do this
	<ul style="list-style-type: none">• Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type.• Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.
Instance store-backed	Do one of the following: <ul style="list-style-type: none">• Terminate the instance and launch a new instance, specifying a different instance type. For example, a larger or a memory-optimized instance type.• Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management update failed)

Memory management update failures are indicated by a system log entry similar to the following:

```
...  
Press `ESC' to enter the menu... 0 [H[J Booting 'Amazon Linux 2011.09  
(2.6.35.14-95.38.amzn1.i686)'  
  
root (hd0)  
  
Filesystem type is ext2fs, using whole disk  
  
kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0  
LANG=  
  
en_US.UTF-8 KEYTABLE=us  
  
initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img  
  
ERROR: mmu_update failed with rc=-22
```

Potential Cause

Issue with Amazon Linux

Suggested Action

Post your issue to the [Developer Forums](#) or contact [AWS Support](#).

I/O error (Block device failure)

An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Attempt to recover the volume. <p>Note It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure.</p> <ol style="list-style-type: none">4. Re-attach the volume to the instance.5. Detach the volume.
Instance store-backed	Terminate the instance and launch a new instance.

Amazon Elastic Compute Cloud
User Guide for Linux Instances
**IO ERROR: neither local nor remote
disk (Broken distributed block device)**

For this instance type	Do this
	<p>Note Data cannot be recovered. Recover from backups.</p> <p>Note It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.</p>

IO ERROR: neither local nor remote disk (Broken distributed block device)

An input/output error on the device is indicated by a system log entry similar to the following example:

```
...  
block drbd1: Local IO failed in request_timer_fn. Detaching...  
  
Aborting journal on device drbd1-8.  
  
block drbd1: IO ERROR: neither local nor remote disk  
  
Buffer I/O error on device drbd1, logical block 557056  
  
lost page write due to I/O error on drbd1  
  
JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested Action

Terminate the instance and launch a new instance.

For an Amazon EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot cannot be recovered.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 0000000026700000 (usable)

OMB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
```

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options: Option 1: Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters. Option 2: <ol style="list-style-type: none">1. Stop the instance.2. Modify the kernel and ramdisk attributes to use a newer kernel.3. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.

"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36
SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

Potential Causes

Incompatible kernel and userland

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Modify the configuration to use a newer kernel.3. Start the instance.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none">1. Create an AMI that uses a newer kernel.2. Terminate the instance.3. Start a new instance from the AMI you created.

"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)

This condition is indicated by a system log similar to the one shown below.

```
[ 0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file
or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for
writing: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
"FATAL: Could not load /lib/modules"
or "BusyBox" (Missing kernel modules)**

```
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device. Common problems:
- Boot args (cat /proc/cmdline)
  - Check rootdelay= (did the system wait long enough?)
  - Check root= (did the system wait for the right device?)
- Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
ALERT! /dev/sda1 does not exist. Dropping to a shell!

BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

Potential Causes

One or more of the following conditions can cause this problem:

- Missing ramdisk
- Missing correct modules from ramdisk
- Amazon EBS root volume not correctly attached as `/dev/sda1`

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none"> 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none"> 1. Terminate the instance and launch a new instance with the correct ramdisk.

For this instance type	Do this
	2. Create a new AMI with the correct ramdisk.

ERROR Invalid kernel (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

Potential Causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Replace with working kernel.3. Install a fallback kernel.4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none">1. Terminate the instance and launch a new instance with the correct kernel.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**
**request_module: runaway loop modprobe (Looping
legacy kernel modprobe on older Linux versions)**

For this instance type	Do this
	<ol style="list-style-type: none"> 2. Create an AMI with the correct kernel. 3. (Optional) Seek technical assistance for data recovery using AWS Support.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 0000000026700000 (usable)

OMB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
```

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	<p>Use a newer kernel, either GRUB-based or static, using one of the following options:</p> <p>Option 1: Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.</p> <p>Option 2:</p> <ol style="list-style-type: none"> 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
fsck: No such file or directory while
trying to open... (File system not found)**

For this instance type	Do this
Instance store-backed	Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.

fsck: No such file or directory while trying to open... (File system not found)

This condition is indicated by a system log similar to the one shown below.

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]

Starting udev: [ OK ]

Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

Potential Causes

- A bug exists in ramdisk filesystem definitions `/etc/fstab`
- Misconfigured filesystem definitions in `/etc/fstab`
- Missing/failed drive

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance, detach the root volume, repair/modify <code>/etc/fstab</code> the volume, attach the volume to the instance, and start the instance.2. Fix ramdisk to include modified <code>/etc/fstab</code> (if applicable).3. Modify the AMI to use a newer ramdisk. <p>Tip The sixth field in the <code>fstab</code> defines availability requirements of the mount – a nonzero value implies that an <code>fsck</code> will be done on that volume and <i>must</i> succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt which is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for <code>fstab</code>.</p>
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Terminate the instance and launch a new instance.2. Detach any errant Amazon EBS volumes and the reboot instance.3. (Optional) Seek technical assistance for data recovery using AWS Support.

General error mounting filesystems (Failed mount)

This condition is indicated by a system log similar to the one shown below.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds
```

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
General error mounting filesystems (Failed mount)**

```
EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none">• Detached or failed Amazon EBS volume.• Corrupted filesystem.• Mismatched ramdisk and AMI combination (e.g., Debian ramdisk with a SUSE AMI).
Instance store-backed	<ul style="list-style-type: none">• A failed drive.• A corrupted file system.• A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Detach the root volume.3. Attach the root volume to a known working instance.4. Run filesystem check (<code>fsck -a /dev/...</code>).5. Fix any errors.6. Detach the volume from the known working instance.7. Attach the volume to the stopped instance.8. Start the instance.9. Recheck the instance status.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
VFS: Unable to mount root fs on unknown-
block (Root filesystem mismatch)**

For this instance type	Do this
Instance store-backed	Try one of the following: <ul style="list-style-type: none"> • Start a new instance. • (Optional) Seek technical assistance for data recovery using AWS Support.

VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
 20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-
block(8,1)
```

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none"> • Device not attached correctly. • Root device not attached at correct device point. • Filesystem not in expected format. • Use of legacy kernel (e.g., 2.6.16-XenU).
Instance store-backed	Hardware device failure.

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: <ul style="list-style-type: none"> • Stop and then restart the instance. • Modify root volume to attach at the correct device point, possible /dev/sda1 instead of /dev/sda. • Stop and modify to use modern kernel.
Instance store-backed	Terminate the instance and launch a new instance using a modern kernel.

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)

This condition is indicated by a system log similar to the one shown below.

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udev[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential Causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda or sda instead of sda1)
- Incorrect choice of DomU kernel

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.4. Start the instance.5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none">1. Create a new AMI with the appropriate fix (map block device correctly).2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

This condition is indicated by a system log similar to the one shown below.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udev[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential Causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda)
- Incorrect choice of DomU kernel

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.4. Start the instance.5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none">1. Create an AMI with the appropriate fix (map block device correctly).2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

This condition is indicated by a system log similar to the one shown below.

```
...  
Checking filesystems  
Checking all file systems.  
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1  
/dev/sda1 has gone 361 days without being checked, check forced
```

Potential Causes

Filesystem check time passed; a filesystem check is being forced

Suggested Actions

- Wait until the filesystem check completes. Note that a filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

fsck died with exit status... (Missing device)

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....  
Loading kernel modules...done.  
...  
Activating lvm and md swap...done.  
Checking file systems...fsck from util-linux-ng 2.16.2  
/sbin/fsck.xfs: /dev/sdh does not exist  
fsck died with exit status 8  
[31mfailed (code 8).[39;49m
```

Potential Causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Try one or more of the following to resolve the issue: <ul style="list-style-type: none">• Stop the instance, attach the volume to an existing running instance.• Manually run consistency checks.• Fix ramdisk to include relevant utilities.• Modify filesystem tuning parameters to remove consistency requirements (not recommended).
Instance store-backed	Try one or more of the following to resolve the issue:

Amazon Elastic Compute Cloud
User Guide for Linux Instances
GRUB prompt (grubdom>)

For this instance type	Do this
	<ul style="list-style-type: none">• Rebundle ramdisk with correct tooling.• Modify file system tuning parameters to remove consistency requirements (not recommended).• Terminate the instance and launch a new instance.• (Optional) Seek technical assistance for data recovery using AWS Support.

GRUB prompt (grubdom>)

This condition is indicated by a system log similar to the one shown below.

```
GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

[ Minimal BASH-like line editing is supported.  For
  the first word, TAB lists possible command
  completions.  Anywhere else TAB lists the possible
  completions of a device/filename. ]

grubdom>
```

Potential Causes

Instance type	Potential causes
Amazon EBS-backed	<ul style="list-style-type: none">• Missing GRUB configuration file.• Incorrect GRUB image used, expecting GRUB configuration file at a different location.• Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).
Instance store-backed	<ul style="list-style-type: none">• Missing GRUB configuration file.• Incorrect GRUB image used, expecting GRUB configuration file at a different location.• Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	<p>Option 1: Modify the AMI and relaunch the instance:</p> <ol style="list-style-type: none">1. Modify the source AMI to create a GRUB configuration file at the standard location (/boot/grub/menu.lst).2. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.3. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition).4. Terminate the instance and launch a new one using the AMI the you created. <p>Option 2: Fix the existing instance:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the root filesystem.3. Attach the root filesystem to a known working instance.4. Mount filesystem.5. Create a GRUB configuration file.6. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.7. Detach filesystem.8. Attach to the original instance.9. Modify kernel attribute to use the appropriate GRUB image (1st disk or 1st partition on 1st disk).10. Start the instance.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Bringing up interface eth0: Device eth0
has different MAC address than expected,**

For this instance type	Do this
Instance store-backed	<p>Option 1: Modify the AMI and relaunch the instance:</p> <ol style="list-style-type: none"> 1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 4. Terminate the instance and launch a new instance using the AMI you created. <p>Option 2: Terminate the instance and launch a new instance, specifying the correct kernel.</p> <p>Note To recover data from the existing instance, contact AWS Support.</p>

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)

This condition is indicated by a system log similar to the one shown below.

```
...
Bringing up loopback interface: [ OK ]

Bringing up interface eth0: Device eth0 has different MAC address than
expected, ignoring.
[FAILED]

Starting auditd: [ OK ]
```

Potential Causes

There is a hard-coded interface MAC in the AMI configuration

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Modify the AMI to remove the hard coding and relaunch the instance. • Modify the instance to remove the hard-coded MAC address.

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
Unable to load SELinux Policy.
Machine is in enforcing mode. Halting**

For this instance type	Do this
	<p>OR</p> <p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Stop the instance. 2. Detach the root volume. 3. Attach the volume to another instance and modify the volume to remove the hard-coded MAC address. 4. Attach the volume to the original instance. 5. Start the instance.
Instance store-backed	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Modify the instance to remove the hard-coded MAC address. • Terminate the instance and launch a new instance.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

Potential Causes

SELinux has been enabled in error:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Stop the failed instance. 2. Detach the failed instance's root volume. 3. Attach the root volume to another running Linux instance (later referred to as a recovery instance). 4. Connect to the recovery instance and mount the failed instance's root volume.

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
**XENBUS: Timeout connecting
 to devices (Xenbus timeout)**

For this instance type	Do this
	5. Disable SELinux on the mounted root volume. This process varies across Linux distributions; for more information, consult your OS-specific documentation. <p>Note On some systems, you disable SELinux by setting <code>SELINUX=disabled</code> in the <code>/mount_point/etc/sysconfig/selinux</code> file, where <code>mount_point</code> is the location that you mounted the volume on your recovery instance.</p> 6. Unmount and detach the root volume from the recovery instance and reattach it to the original instance. 7. Start the instance.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none"> 1. Terminate the instance and launch a new instance. 2. (Optional) Seek technical assistance for data recovery using AWS Support.

XENBUS: Timeout connecting to devices (Xenbus timeout)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to
kernel.
```

Potential Causes

- The block device not is connected to the instance
- This instance is using a very old DomU kernel

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: <ul style="list-style-type: none"> • Modify the AMI and instance to use a modern kernel and relaunch the instance.

For this instance type	Do this
	<ul style="list-style-type: none">• Reboot the instance.
Instance store-backed	Do one of the following: <ul style="list-style-type: none">• Terminate the instance.• Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshooting Instance Capacity

The following errors are related to instance capacity.

Error: InsufficientInstanceCapacity

If you get an `InsufficientInstanceCapacity` error when you try to launch an instance, AWS does not currently have enough available capacity to service your request. Try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- Submit a new request without specifying an Availability Zone.
- Submit a new request using a different instance type (which you can resize at a later stage). For more information, see [Resizing Your Instance \(p. 189\)](#).
- Try purchasing Reserved Instances. Reserved Instances are a long-term capacity reservation. For more information, see: [Amazon EC2 Reserved Instances](#).

Error: InstanceLimitExceeded

If you get an `InstanceLimitExceeded` error when you try to launch an instance, you have reached your concurrent running instance limit. For new AWS accounts, the default limit is 20. If you need additional running instances, complete the form at [Request to Increase Amazon EC2 Instance Limit](#).

Getting Console Output and Rebooting Instances

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

Similarly, the ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

EC2 instances do not have a physical monitor through which you can view their console output. They also lack physical controls that allow you to power up, reboot, or shut them down. Instead, you perform these tasks through the Amazon EC2 API and the command line interface (CLI).

Instance Reboot

Just as you can reset a computer by pressing the reset button, you can reset EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see [Reboot Your Instance \(p. 302\)](#).

Caution

For Windows instances, this operation performs a hard reboot that might result in data corruption.

Instance Console Output

For Linux/Unix instances, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. This output is buffered because the instance produces it and then posts it to a store where the instance's owner can retrieve it.

For Windows instances, the instance console output displays the last three system event log errors.

The posted output is not continuously updated; only when it is likely to be of the most value. This includes shortly after instance boot, after reboot, and when the instance terminates.

Note

Only the most recent 64 KB of posted output is stored, which is available for at least 1 hour after the last posting.

Only the instance owner can access the console output. You can retrieve the console output for your instances using the console or the command line.

To get console output using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**, and select the instance.
3. Choose **Actions, Instance Settings, Get System Log**.

To get console output using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [get-console-output](#) (AWS CLI)
- [Get-EC2ConsoleOutput](#) (AWS Tools for Windows PowerShell)

For more information about common system log errors, see [Troubleshooting System Log Errors for Linux-Based Instances \(p. 783\)](#).

Capture a Screenshot of an Unreachable Instance

If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

There is no data transfer cost for this screenshot. The image is generated in JPG format, no larger than 100kb.

To access the instance console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the left navigation pane, choose **Instances**.
3. Select the instance to capture.
4. Choose **Actions, Instance Settings**.
5. Choose **Get Instance Screenshot**.

Right-click on the image to download and save it.

To capture a screenshot using the command line

You can use one of the following commands. The returned content is base64-encoded. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [get-console-screenshot](#) (AWS CLI)
- [GetConsoleScreenshot](#) (Amazon EC2 Query API)

Instance Recovery When a Host Computer Fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You'll be notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.
5. [EC2-Classic] If the instance had an associated Elastic IP address, you must reassociate it with the instance.

For more information, see [Stop and Start Your Instance \(p. 299\)](#).

To recover an instance store-backed instance running on a host computer that failed

1. Create an AMI from the instance.
2. Upload the image to Amazon S3.
3. Back up important data to Amazon EBS or Amazon S3.
4. Terminate the instance.
5. Launch a new instance from the AMI.
6. Restore any important data to the new instance.
7. [EC2-Classic] If the original instance had an associated Elastic IP address, you must associate it with the new instance.

For more information, see [Creating an Instance Store-Backed Linux AMI \(p. 92\)](#).

My Instance is Booting from the Wrong Volume

In some situations, you may find that a volume other than the volume attached to `/dev/xvda` or `/dev/sda` has become the root volume of your instance. This can happen when you have attached

Amazon Elastic Compute Cloud User Guide for Linux Instances My Instance is Booting from the Wrong Volume

the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume.

This is due to how the initial ramdisk in Linux works. It will choose the volume defined as `/` in the `/etc/fstab`, and in some distributions, including Amazon Linux, this is determined by the label attached to the volume partition. Specifically, you will find that your `/etc/fstab` looks something like the following:

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

And if you were to check the label of both volumes, you would see that they both contain the `/` label:

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

In this example, you could end up having `/dev/xvdf1` become the root device that your instance boots to after the initial ramdisk runs, instead of the `/dev/xvda1` volume you had intended to boot from. Solving this is fairly simple; you can use the same **e2label** command to change the label of the attached volume that you do not want to boot from.

Note

In some cases, specifying a UUID in `/etc/fstab` can resolve this, however, if both volumes come from the same snapshot, or the secondary is created from a snapshot of the primary volume, they will share a UUID.

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778
TYPE="ext4" PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-
b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778
TYPE="ext4" PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-
b514-7084e28f7334
```

To change the label of an attached volume

1. Use the **e2label** command to change the label of the volume to something other than `/`.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. Verify that the volume has the new label.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1
old/
```

After making this change, you should be able to reboot the instance and have the proper volume selected by the initial ramdisk when the instance boots.

Important

If you intend to detach the volume with the new label and return it to another instance to use as the root volume, you must perform the above procedure again and change the volume

**Amazon Elastic Compute Cloud
User Guide for Linux Instances
My Instance is Booting from the Wrong Volume**

label back to its original value; otherwise, the other instance will not boot because the ramdisk will be unable to find the volume with the label / .

Making API Requests

We provide the Query API for Amazon EC2, as well as software development kits (SDK) for Amazon Web Services (AWS) that enable you to access Amazon EC2 from your preferred programming language.

For more information about working with the Query API for Amazon EC2, see [Making API Requests](#) in the *Amazon EC2 API Reference*.

Document History

The following table describes important additions to the Amazon EC2 documentation. We also update the documentation frequently to address the feedback that you send us.

Current API version: 2016-04-01.

Feature	API Version	Description	Release Date
Automatic scaling for Spot fleet		You can now set up scaling policies for your Spot fleet. For more information, see Automatic Scaling for Spot Fleet (p. 253) .	1 September 2016
Run Command support for managed instances	2016-04-01	Amazon EC2 Run Command now supports the management of on-premises servers and virtual machines (VMs) and VMs from other cloud providers. For more information, see Setting Up Run Command On Managed Instances (p. 371)	30 June 2016
Elastic Network Adapter (ENA)	2016-04-01	You can now use ENA for enhanced networking. For more information, see Enhanced Networking Types (p. 608) .	28 June 2016
Enhanced support for viewing and modifying longer IDs	2016-04-01	You can now view and modify longer ID settings for other IAM users, IAM roles, or the root user. For more information, see Resource IDs (p. 744) .	23 June 2016
Copy encrypted Amazon EBS snapshots between AWS accounts	2016-04-01	You can now copy encrypted EBS snapshots between AWS accounts. For more information, see Copying an Amazon EBS Snapshot (p. 688) .	21 June 2016
Capture a screenshot of an instance console	2015-10-01	You can now obtain additional information when debugging instances that are unreachable. For more information, see Capture a Screenshot of an Unreachable Instance (p. 806) .	24 May 2016
X1 instances	2015-10-01	Memory-optimized instances designed for running in-memory databases, big data processing engines, and high performance computing (HPC) applications. For more information, see X1 Instances (p. 187) .	18 May 2016

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Two new EBS volume types	2015-10-01	You can now create Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see Amazon EBS Volume Types (p. 638) .	19 April 2016
Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2		Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2. For more information, see Instance Metrics (p. 421) .	23 March 2016
CloudWatch metrics for Spot fleet		You can now get CloudWatch metrics for your Spot fleet. For more information, see CloudWatch Metrics for Spot Fleet (p. 251) .	21 March 2016
Scheduled Instances	2015-10-01	Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration. For more information, see Scheduled Reserved Instances (p. 215) .	13 January 2016
Longer resource IDs	2015-10-01	We're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. During the opt-in period, you can enable the longer ID format for supported resource types. For more information, see Resource IDs (p. 744) .	13 January 2016
ClassicLink DNS support	2015-10-01	You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For more information, see Enabling ClassicLink DNS Support (p. 555) .	11 January 2016
New t2.nano instance type	2015-10-01	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 160) .	15 December 2015
Dedicated hosts	2015-10-01	An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Dedicated Hosts (p. 264) .	23 November 2015
Spot instance duration	2015-10-01	You can now specify a duration for your Spot instances. For more information, see Specifying a Duration for Your Spot Instances (p. 231) .	6 October 2015

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Spot fleet modify request	2015-10-01	You can now modify the target capacity of your Spot fleet request. For more information, see Modifying a Spot Fleet Request (p. 242) .	29 September 2015
Spot fleet diversified allocation strategy	2015-04-15	You can now allocate Spot instances in multiple Spot pools using a single Spot fleet request. For more information, see Spot Fleet Allocation Strategy (p. 224) .	15 September 2015
Spot fleet instance weighting	2015-04-15	You can now define the capacity units that each instance type contributes to your application's performance, and adjust your bid price for each Spot pool accordingly. For more information, see Spot Fleet Instance Weighting (p. 225) .	31 August 2015
New reboot alarm action and new IAM role for use with alarm actions		Added the reboot alarm action and new IAM role for use with alarm actions. For more information, see Create Alarms That Stop, Terminate, Reboot, or Recover an Instance (p. 450) .	23 July 2015
New t2.large instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 160) .	16 June 2015
M4 instances		The next generation of general-purpose instances that provide a balance of compute, memory, and network resources. M4 instances are powered by a custom Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processor with AVX2.	11 June 2015
Spot fleets	2015-04-15	You can manage a collection, or fleet, of Spot instances instead of managing separate Spot instance requests. For more information, see How Spot Fleet Works (p. 224) .	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 583) .	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports importing VMs with multiple disks as AMIs. For more information, see Importing a VM as an Image Using VM Import/Export in the <i>VM Import/Export User Guide</i> .	23 April 2015

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
New g2.8xlarge instance type		The new g2.8xlarge instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015
D2 instances		Next generation Amazon EC2 dense-storage instances that are optimized for applications requiring sequential access to large amount of data on direct attached instance storage. D2 instances are designed to offer best price/performance in the dense-storage family. Powered by 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processors, D2 instances improve on HS1 instances by providing additional compute power, more memory, and Enhanced Networking. In addition, D2 instances are available in four instance sizes with 6TB, 12TB, 24TB, and 48TB storage options. For more information, see D2 Instances (p. 170) .	24 March 2015
Automatic recovery for EC2 instances		You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata. For more information, see Recover Your Instance (p. 310) .	12 January 2015
C4 instances		Next-generation compute-optimized instances that provide very high CPU performance at an economical price. C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors. With additional Turbo boost, the processor clock speed in C4 instances can reach as high as 3.5Ghz with 1 or 2 core turbo. Expanding on the capabilities of C3 compute-optimized instances, C4 instances offer customers the highest processor performance among EC2 instances. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For more information, see C4 Instances (p. 163) .	11 January 2015

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
ClassicLink	2014-10-01	ClassicLink enables you to link your EC2-Classic instance to a VPC in your account. You can associate VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. For more information, see ClassicLink (p. 550) .	7 January 2015
Spot instance termination notices		The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance. For more information, see Spot Instance Termination Notices (p. 261) .	5 January 2015
DescribeVolumes pagination support	2014-09-01	The <code>DescribeVolumes</code> API call now supports the pagination of results with the <code>MaxResults</code> and <code>NextToken</code> parameters. For more information, see DescribeVolumes in the <i>Amazon EC2 API Reference</i> .	23 October 2014
T2 instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 160) .	30 June 2014
New EC2 Service Limits page		Use the EC2 Service Limits page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose SSD volumes can range in size from 1 GiB to 1 TiB. For more information, see General Purpose SSD (gp2) Volumes (p. 640) .	16 June 2014

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more information, see Amazon EBS Encryption (p. 695) .	21 May 2014
R3 instances	2014-02-01	Next generation memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	9 April 2014
New Amazon Linux AMI release		Amazon Linux AMI 2014.03 is released.	27 March 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 Usage Reports (p. 762) .	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes <code>m3.medium</code> and <code>m3.large</code> are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS and support TRIM on Linux instances for better successive SSD write performance. I2 instances also support enhanced networking that delivers improve inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see I2 Instances (p. 169) .	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	19 December 2013

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Importing Linux virtual machines	2013-10-15	The VM Import process now supports the importation of Linux instances. For more information, see the VM Import/Export User Guide .	16 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Controlling Access to Amazon EC2 Resources (p. 498).	20 November 2013
C3 instances	2013-10-15	Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 287).	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power. For more information, see Linux GPU Instances (p. 166).	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an Instance (p. 279).	10 October 2013
Modifying Instance Types of Amazon EC2 Reserved Instances	2013-10-01	You can now modify the instance type of Linux Reserved Instances within the same family (for example, M1, M2, M3, C1). For more information, see Modifying Your Reserved Instances (p. 210).	09 October 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.09 is released.	30 September 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region. For more information, see Modifying Your Reserved Instances (p. 210).	11 September 2013

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assigning a Public IP Address (p. 575) .	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM Policies for Amazon EC2 (p. 500) .	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS Snapshot (p. 688) .	11 June 2013
New Tags page		There is a new Tags page in the Amazon EC2 console. For more information, see Tagging Your Amazon EC2 Resources (p. 751) .	04 April 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.03 is released.	27 March 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: <code>c1.xlarge</code> , <code>m2.2xlarge</code> , <code>m3.xlarge</code> , and <code>m3.2xlarge</code> . For more information, see Amazon EBS-Optimized Instances (p. 692) .	19 March 2013
Copy an AMI from one region to another	2013-02-01	You can copy an AMI from one region to another, enabling you to launch consistent instances in more than one AWS region quickly and easily. For more information, see Copying an AMI (p. 135) .	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either the EC2-Classic or EC2-VPC platform, or only into the EC2-VPC platform, on a region-by-region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance. For more information, see Supported Platforms (p. 549) .	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see HS1 Instances (p. 174) .	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS Snapshot (p. 688) .	17 December 2012
Updated EBS metrics and status checks for Provisioned IOPS SSD volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with CloudWatch (p. 658) . Also added new status checks for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with Status Checks (p. 661) .	20 November 2012
Linux Kernels		Updated AKI IDs; reorganized distribution kernels; updated PVOps section.	13 November 2012
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	31 October 2012
Spot instance request status	2012-10-01	Spot instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
New Amazon Linux AMI release		Amazon Linux AMI 2012.09 is released.	11 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see Amazon EBS Volume Types (p. 638) .	31 July 2012

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage. For more information, see H1 Instances (p. 173) .	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <ul style="list-style-type: none"> • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services. 	11 June 2012
Spot instance features that make it easier to get started and handle the potential of interruption.		You can now manage your Spot instances as follows: <ul style="list-style-type: none"> • Place bids for Spot instances using Auto Scaling launch configurations, and set up a schedule for placing bids for Spot instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the <i>Auto Scaling User Guide</i>. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot instances in a stack with AWS resources. 	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for <code>cc2.8xlarge</code> instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
New Linux AMI release		Amazon Linux AMI 2012.03 is released.	28 March 2012
New AKI version		We've released AKI version 1.03 and AKIs for the AWS GovCloud (US) region.	28 March 2012

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Medium instances, support for 64-bit on all AMIs, and a Java-based SSH Client	2011-12-15	Added support for a new instance type and 64-bit information. Added procedures for using the Java-based SSH client to connect to Linux instances.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic Network Interfaces (ENI) (p. 588) .	21 December 2011
New GRU Region and AKIs		Added information about the release of new AKIs for the SA-East-1 Region. This release deprecates the AKI version 1.01. AKI version 1.02 will continue to be backward compatible.	14 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitoring the Status of Your Instances (p. 411) .	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
New PDX Region and AKIs		Added information about the release of new AKIs for the new US-West 2 Region.	8 November 2011
Spot instances in Amazon VPC	2011-07-15	Added information about the support for Spot instances in Amazon VPC. With this update, users can launch Spot instances a virtual private cloud (VPC). By launching Spot instances in a VPC, users of Spot instances can enjoy the benefits of Amazon VPC.	11 October 2011
New Linux AMI release		Added information about the release of Amazon Linux AMI 2011.09. This update removes the beta tag from the Amazon Linux AMI, supports the ability to lock the repositories to a specific version, and provides for notification when updates are available to installed packages including security updates.	26 September 2011

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of <code>ImportInstance</code> and <code>ImportVolume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of <code>ResumeImport</code> , users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide .	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes.	27 June 2011
Enabling Linux AMI to run user-provided kernels		Added information about the AKI version change from 1.01 to 1.02. This version updates the PVGRUB to address launch failures associated with t1.micro Linux instances. For more information, see User Provided Kernels (p. 149) .	20 June 2011
Spot instances Availability Zone pricing changes	2011-05-15	Added information about the Spot instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot instance requests and Spot price history. These additions make it easier to determine the price required to launch a Spot instance into a particular Availability Zone.	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Controlling Access to Amazon EC2 Resources (p. 498) .	26 April 2011
Enabling Linux AMI to run user-provided kernels		Added information about enabling a Linux AMI to use PVGRUB Amazon Kernel Image (AKI) to run a user-provided kernel. For more information, see User Provided Kernels (p. 149) .	26 April 2011

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Using EC2 Dedicated Instances in the <i>Amazon VPC User Guide</i> .	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances. For more information, see Reserved Instances (p. 194) .	27 March 2011
New Amazon Linux reference AMI		The new Amazon Linux reference AMI replaces the CentOS reference AMI. Removed information about the CentOS reference AMI, including the section named Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI. For more information, see AMIs for GPU Instances (p. 167) .	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance Metadata and User Data (p. 337) and Instance Metadata Categories (p. 344) .	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2.	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 669) .	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enabling Termination Protection for an Instance (p. 306) .	23 February 2011
Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI		Added information about how to correct clock drift for cluster instances running on Amazon's CentOS 5.4 AMI.	25 January 2011

**Amazon Elastic Compute Cloud
User Guide for Linux Instances**

Feature	API Version	Description	Release Date
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see the VM Import/Export User Guide .	15 December 2010
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Cluster GPU instances	2010-08-31	Amazon EC2 offers cluster GPU instances (cg1.4xlarge) for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	14 November 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 748) and Tagging Your Amazon EC2 Resources (p. 751) .	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances. For more information, see Ensuring Idempotency in the <i>Amazon EC2 API Reference</i> .	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the <code>t1.micro</code> instance type for certain types of applications. For more information, see T1 Micro Instances (p. 175) .	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Controlling Access to Amazon EC2 Resources (p. 498) .	2 September 2010
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Monitoring Volumes with CloudWatch (p. 658) .	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	22 February 2010

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.