

Image Management Service

User Guide (Paris Region)

Issue	04
Date	2021-03-30



Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

Contents

1 Overview.....	1
1.1 What Is Image Management Service?.....	1
1.2 Supported OSs.....	2
1.2.1 Formats and OSs Supported for External Image Files.....	2
1.2.2 OSs Supporting UEFI Boot Mode.....	7
1.3 Basic Concepts.....	9
1.3.1 Region and AZ.....	9
1.3.2 Common Image Formats.....	10
1.4 Related Services.....	12
2 Creating a Private Image.....	14
2.1 Introduction.....	14
2.2 Creating a System Disk Image from a Windows ECS.....	14
2.3 Creating a System Disk Image from a Linux ECS.....	16
2.4 Creating a Windows System Disk Image from an External Image File.....	18
2.4.1 Overview.....	18
2.4.2 Preparing an Image File.....	19
2.4.3 Uploading an External Image File.....	21
2.4.4 Registering an External Image File as a Private Image.....	22
2.4.5 Creating a Windows ECS from an Image.....	24
2.5 Creating a Linux System Disk Image from an External Image File.....	25
2.5.1 Overview.....	25
2.5.2 Preparing an Image File.....	25
2.5.3 Uploading an External Image File.....	28
2.5.4 Registering an External Image File as a Private Image.....	28
2.5.5 Creating a Linux ECS from an Image.....	31
2.6 Creating a BMS System Disk Image.....	32
2.7 Creating a Full-ECS Image from an ECS.....	32
2.8 Creating a Full-ECS Image from a CSBS Backup.....	34
2.9 Quickly Importing an Image File.....	35
2.9.1 Overview.....	35
2.9.2 Quickly Importing an Image File (Linux).....	37
2.9.3 Quickly Importing an Image File (Windows).....	41
3 Managing Private Images.....	44

3.1 Modifying Image Information.....	44
3.2 Checking the Disk Capacity of an Image.....	45
3.3 Creating an ECS from an Image.....	46
3.4 Deleting Images.....	46
3.5 Sharing Images.....	47
3.5.1 Overview.....	47
3.5.2 Obtaining the Project ID.....	48
3.5.3 Sharing Specified Images.....	48
3.5.4 Accepting or Rejecting Shared Images.....	49
3.5.5 Rejecting Accepted Images.....	50
3.5.6 Accepting Rejected Images.....	50
3.5.7 Stopping Sharing Images.....	51
3.5.8 Adding Tenants Who Can Use Shared Images.....	51
3.5.9 Deleting Image Recipients Who Can Use Shared Images.....	52
3.6 Exporting Images.....	52
3.7 Optimizing a Windows Private Image.....	54
3.7.1 Optimization Process.....	54
3.7.2 Viewing the Virtualization Type of a Windows ECS.....	54
3.7.3 Obtaining Required Software Packages.....	55
3.7.4 Installing the PV Driver.....	56
3.7.5 Installing UVP VMTools.....	58
3.7.6 Clearing System Logs.....	61
3.8 Optimizing a Linux Private Image.....	61
3.8.1 Optimization Process.....	61
3.8.2 Viewing the Virtualization Type of a Linux ECS.....	62
3.8.3 Uninstalling the PV Driver from a Linux ECS.....	62
3.8.4 Changing the Disk Identifier in the GRUB Configuration File to UUID.....	63
3.8.5 Changing the Disk Identifier in the fstab File to UUID.....	68
3.8.6 Installing Native KVM Drivers.....	69
3.8.7 Clearing System Logs.....	75
3.9 Encrypting Images.....	75
3.9.1 Overview.....	75
3.9.2 Creating Encrypted Images.....	75
3.10 Replicating Images.....	76
3.11 Tagging an Image.....	77
3.12 Exporting Image Information.....	78
3.13 Auditing Key Operations.....	79
3.13.1 IMS Operations Recorded by CTS.....	79
3.13.2 Viewing Traces.....	80
3.14 Converting the Image Format Using qemu-img.....	81
3.15 Converting the Image Format Using qemu-img-hw.....	85
4 Windows Operations.....	88

4.1 Setting the NIC to DHCP.....	88
4.2 Enabling Remote Desktop Connection.....	90
4.3 Installing and Configuring Cloudbase-Init.....	91
4.4 Running Sysprep.....	96
4.5 Installing Special Windows Drivers.....	98
5 Linux Operations.....	100
5.1 Setting the NIC to DHCP.....	100
5.2 Deleting Files in the Network Rule Directory.....	102
5.3 Installing Cloud-Init.....	103
5.4 Configuring Cloud-Init.....	108
5.5 Detaching Data Disks from an ECS.....	114
6 FAQs.....	116
6.1 Image Consulting.....	116
6.1.1 How Do I Select an Image?.....	116
6.1.2 How Do I Increase the Image Quota?.....	117
6.1.3 Can I Use Private Images of Other Tenants?.....	118
6.2 Image Creation.....	118
6.2.1 Image Creation FAQs.....	118
6.2.2 How Do I Create a Full-ECS Image Using an ECS That Has a Spanned Volume?.....	119
6.2.3 Why Is Sysprep Required for Creating a Private Image from a Windows ECS?.....	119
6.2.4 What Do I Do If I Cannot Create an Image in ZVHD2 Format Using an API?.....	120
6.3 Image Sharing.....	120
6.3.1 Image Sharing.....	120
6.4 OS.....	121
6.4.1 How Is BIOS Different from UEFI?.....	121
6.4.2 How Do I Delete Redundant Network Connections from a Windows ECS?.....	121
6.4.3 What Do I Do If an ECS Starts Slowly?.....	122
6.4.4 What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?.....	123
6.5 Image Importing.....	124
6.5.1 Can I Use Images in Formats Other Than Those Specified in This Document?.....	124
6.5.2 What Are the Impacts If I Do Not Pre-configure an ECS Used to Create a Private Image?.....	124
6.5.3 What Do I Do If I Configure an Incorrect OS or System Disk Size During Private Image Registration Using an Image File?.....	125
6.5.4 Why Does the Error Message Displayed on Task Center Indicates That the System Disk Size of the External Image File Exceeds the Maximum System Disk Size When a VHD Image File Failed to Be Uploaded?.....	125
6.6 Image Optimization.....	125
6.6.1 Must I Install Guest OS Drivers on an ECS?.....	125
6.6.2 Why Do I Need to Install and Update VMTools for Windows?.....	126
6.6.3 What Changes Will Be Made to an Image File Used for Registering a Private Image?.....	126
6.6.4 What Initial Configuration Needs to Be Performed on the ECS, BMS, or Image File Before It Is Used to Create an Image?.....	128

6.6.5 What Do I Do If the Initial Configurations of a Windows External Image File Are Not Completed Before the File Is Exported?.....	131
6.6.6 What Do I Do If the Initial Configurations of a Linux External Image File Are Not Completed Before the File Is Exported?.....	134
6.6.7 How Do I Set NIC Multi-Queue for an Image?.....	136
6.6.8 How Do I Optimize a System Disk Image So That It Can Be Used to Create ECSs Quickly?.....	142
6.6.9 What Is the Cause of the Failure to Install a Guest OS Driver on a Windows ECS?.....	142
6.6.10 How Do I Install Native Xen and KVM Drivers?.....	143
6.7 Image Encryption.....	150
6.8 Cloud-Init.....	150
6.8.1 What Can I Do with a Cloud-Init ECS?.....	150
6.8.2 What Do I Do If Injecting the Key or Password Using Cloud-Init Failed After NetworkManager Is Installed?.....	151
6.8.3 How Do I Install growpart for SUSE 11 SP4?.....	151
6.8.4 How Do I Configure a Linux Private Image That Can Automatically Expand Its Root Partition?.....	152
6.9 ECS Creation.....	158
6.9.1 Can an ECS Created from a Private Image Have Different Hardware Specifications from the ECS Used to Create the Private Image?.....	158
6.9.2 Can I Specify the System Disk Size When I Create an ECS Using an Image?.....	158
6.9.3 What Do I Do If No Partition Is Found During the Startup of an ECS Created from an Imported Private Image?.....	158
6.9.4 What Do I Do If the Disks of an ECS Created from a CentOS Image Cannot Be Found?.....	161
6.9.5 What Do I Do If an ECS Created from a Windows Image Failed to Start When I Have Selected Enable Automatic Configuration During Image Registration?.....	162
6.9.6 What Do I Do If an Exception Occurs When I Start an ECS Created from an Image Using the UEFI Boot Mode?.....	163
A Change History.....	164

1 Overview

1.1 What Is Image Management Service?

Overview

An image is an Elastic Cloud Server (ECS) or a Bare Metal Server (BMS) template that contains an operating system (OS) or service data and necessary software, such as database software. IMS provides public, private, and shared images.

Image Management Service (IMS) allows you to manage the entire lifecycle of your images. You can create ECSs or BMSs from public, private, or shared images. You can also create a private image from a cloud server or an external image file to make it easier to migrate workloads to the cloud or on the cloud.

Image Types

Images are classified as public, private, and shared. Public images are provided by the cloud platform, private images are those you created yourself, and shared images are private images that other tenants have shared with you.

Image Type	Description
Public image	A public image is a standard, widely used image. It contains an OS and preinstalled public applications and is available to all users. Public images are very stable and their OS and any included software have been officially authorized for use. If a public image does not contain the application environments or software you need, you can use a public image to create an ECS and then deploy required software as needed.
Private image	A private image is an image that contains an OS or service data, pre-installed public applications, and user's personal applications. Private images are only available to the users who created them.
Shared image	A shared image is a private image another user has shared with you.

IMS Functions

IMS provides:

- Public images that contain common OSs
- Creation of a private image from an ECS or external image file
- Public image management, such as searching for images by OS type, name, or ID, and viewing the image ID, system disk size, and image features such as user data injection and disk hot swap
- Private image management, such as modifying image attributes, sharing images, and replicating images
- Creation of ECSs using an image

Access Methods

The public cloud provides a web-based service management platform (a management console). You can access the IMS service through HTTPS APIs or from the management console.

- API
If you need to integrate IMS into a third-party system for secondary development, use APIs to access the IMS service. For details, see *Image Management Service API Reference*.
- Management console
If no integration with a third party system is needed, use the management console.

1.2 Supported OSs

1.2.1 Formats and OSs Supported for External Image Files

Supported File Formats

Image files in VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, or ZVHD format can be used to create private images. Select whichever format best meeting your requirements.

Supported OSs

When you upload an external image file to an OBS bucket on the management console, the OS contained in the image file will be checked. [Table 1-1](#) lists the OSs supported for external image files. (If you use a public image to create an ECS and then use the ECS to create a private image, the private image also supports the OSs listed in [Table 1-1](#).)

- For details about OSs supported by BIOS image files, see [Table 1-1](#). If the OS is not included in the table, either **Other Windows(64bit)** or **Other**

Windows(32bit) will be used for Windows, and either **Other Linux(64bit)** or **Other Linux(32bit)** for Linux during image registration.

- For more information about the OSs supported by UEFI image files, see [OSs Supporting UEFI Boot Mode](#).

 NOTE

- Uploading image files containing OSs not listed in [Table 1-1](#) may fail. You are advised to contact the customer service before uploading these image files.
- When uploading a CoreOS image file, set the OS type to CoreOS. Otherwise, the OS type will be set to **Other (64bit)**. In addition, ensure that coreos-cloudinit has been installed and configured for CoreOS. Automatic system upgrades must be disabled. Otherwise, they may make ECSs created using this image unavailable.

Table 1-1 Supported OSs

OS	OS Version
Windows	Windows 10 64bit Windows Server 2016 Standard 64bit Windows Server 2016 Datacenter 64bit Windows Server 2012 R2 Standard 64bit Windows Server 2012 Essentials R2 64bit Windows Server 2012 R2 Datacenter 64bit Windows Server 2012 Datacenter 64bit Windows Server 2012 Standard 64bit Windows Server 2008 R2 WEB 64bit Windows Server 2008 R2 Standard 64bit Windows Server 2008 R2 Enterprise 64bit Windows Server 2008 R2 Datacenter 64bit
SUSE	SUSE Linux Enterprise Server 15 SP1 64bit SUSE Linux Enterprise Server 15 64bit SUSE Linux Enterprise Server 12 SP5 64bit SUSE Linux Enterprise Server 12 SP3 64bit SUSE Linux Enterprise Server 12 SP2 64bit SUSE Linux Enterprise Server 12 SP1 64bit SUSE Linux Enterprise Server 12 64bit SUSE Linux Enterprise Server 11 SP4 64bit SUSE Linux Enterprise Server 11 SP3 64bit SUSE Linux Enterprise Server 11 SP3 32bit SUSE Linux Enterprise Server 11 SP1 32bit

OS	OS Version
Oracle Linux	Oracle Linux Server release 7.6 64bit Oracle Linux Server release 7.5 64bit Oracle Linux Server release 7.4 64bit Oracle Linux Server release 7.3 64bit Oracle Linux Server release 7.2 64bit Oracle Linux Server release 7.1 64bit Oracle Linux Server release 7.0 64bit Oracle Linux Server release 6.10 64bit Oracle Linux Server release 6.9 64bit Oracle Linux Server release 6.8 64bit Oracle Linux Server release 6.7 64bit Oracle Linux Server release 6.5 64bit
Red Hat	Red Hat Linux Enterprise 8.0 64bit Red Hat Linux Enterprise 7.6 64bit Red Hat Linux Enterprise 7.5 64bit Red Hat Linux Enterprise 7.4 64bit Red Hat Linux Enterprise 7.3 64bit Red Hat Linux Enterprise 7.2 64bit Red Hat Linux Enterprise 7.1 64bit Red Hat Linux Enterprise 7.0 64bit Red Hat Linux Enterprise 6.10 64bit Red Hat Linux Enterprise 6.9 64bit Red Hat Linux Enterprise 6.8 64bit Red Hat Linux Enterprise 6.7 64bit Red Hat Linux Enterprise 6.6 64bit Red Hat Linux Enterprise 6.6 32bit Red Hat Linux Enterprise 6.5 64bit Red Hat Linux Enterprise 6.4 64bit Red Hat Linux Enterprise 6.4 32bit

OS	OS Version
Ubuntu	Ubuntu 20.04 Server 64bit Ubuntu 19.04 Server 64bit Ubuntu 18.04 Server 64bit Ubuntu 16.04 Server 64bit Ubuntu 14.04.4 Server 64bit Ubuntu 14.04.4 Server 32bit Ubuntu 14.04.3 Server 64bit Ubuntu 14.04.3 Server 32bit Ubuntu 14.04.1 Server 64bit Ubuntu 14.04.1 Server 32bit Ubuntu 14.04 Server 64bit Ubuntu 14.04 Server 32bit
openSUSE	openSUSE 42.3 64bit openSUSE 42.2 64bit openSUSE 42.1 64bit openSUSE 15.1 64bit openSUSE 15.0 64bit openSUSE 13.2 64bit openSUSE 11.3 64bit

OS	OS Version
CentOS	CentOS 8.0 64bit CentOS 7.9 64bit CentOS 7.8 64bit CentOS 7.7 64bit CentOS 7.6 64bit CentOS 7.5 64bit CentOS 7.4 64bit CentOS 7.3 64bit CentOS 7.2 64bit CentOS 7.1 64bit CentOS 7.0 64bit CentOS 7.0 32bit CentOS 6.10 64bit CentOS 6.10 32bit CentOS 6.9 64bit CentOS 6.8 64bit CentOS 6.7 64bit CentOS 6.7 32bit CentOS 6.6 64bit CentOS 6.6 32bit CentOS 6.5 64bit CentOS 6.5 32bit CentOS 6.4 64bit CentOS 6.4 32bit CentOS 6.3 64bit CentOS 6.3 32bit
Debian	Debian GNU/Linux 10.0.0 64bit Debian GNU/Linux 9.3.0 64bit Debian GNU/Linux 9.0.0 64bit Debian GNU/Linux 8.8.0 64bit Debian GNU/Linux 8.7.0 64bit Debian GNU/Linux 8.6.0 64bit Debian GNU/Linux 8.5.0 64bit Debian GNU/Linux 8.4.0 64bit Debian GNU/Linux 8.2.0 64bit Debian GNU/Linux 8.1.0 64bit

OS	OS Version
Fedora	Fedora 30 64bit Fedora 28 64bit Fedora 27 64bit Fedora 26 64bit Fedora 25 64bit Fedora 24 64bit Fedora 23 64bit Fedora 22 64bit
EulerOS	EulerOS 2.9 64bit EulerOS 2.5 64bit EulerOS 2.3 64bit EulerOS 2.2 64bit EulerOS 2.1 64bit
CoreOS	CoreOS 1068.10.0 CoreOS 1010.5.0 CoreOS 1298.6.0
openEuler	openEuler 20.03 64bit

Related Operations

For how to upload an external image file, see [Uploading an External Image File](#) and [Uploading an External Image File](#).

After an external image file is successfully uploaded, you can register this image file as a private image on the cloud platform. For details, see [Registering an External Image File as a Private Image](#) and [Registering an External Image File as a Private Image](#).

1.2.2 OSs Supporting UEFI Boot Mode

The ECS boot mode can be BIOS or UEFI. For details about the differences between the two modes, see [How Is BIOS Different from UEFI?](#)

Table 1-2 lists the OSs that support the UEFI boot mode.

Table 1-2 OSs supporting UEFI boot mode

OS	OS Version
Windows	Windows Server 2019 Datacenter 64bit
	Windows Server 2019 Standard 64bit
	Windows Server 2016 Standard 64bit

OS	OS Version
	Windows Server 2016 Datacenter 64bit
	Windows Server 2012 R2 Standard 64bit
	Windows Server 2012 R2 Datacenter 64bit
	Windows Server 2012 Essentials R2 64bit
	Windows Server 2012 Standard 64bit
	Windows Server 2012 Datacenter 64bit
	Windows 10 64bit
Ubuntu	Ubuntu 19.04 Server 64bit
	Ubuntu 18.04 Server 64bit
	Ubuntu 16.04 Server 64bit
	Ubuntu 14.04 Server 64bit
Red Hat	Red Hat Linux Enterprise 7.4 64bit
	Red Hat Linux Enterprise 7.3 64bit
	Red Hat Linux Enterprise 7.1 64bit
	Red Hat Linux Enterprise 7.0 64bit
	Red Hat Linux Enterprise 6.9 64bit
	Red Hat Linux Enterprise 6.6 32bit
	Red Hat Linux Enterprise 6.5 64bit
Oracle Linux	Oracle Linux Server release 7.4 64bit
	Oracle Linux Server release 6.9 64bit
openSUSE	openSUSE 42.1 64bit
SUSE	SUSE Linux Enterprise Server 12 SP5 64bit
	SUSE Linux Enterprise Server 12 SP1 64bit
	SUSE Linux Enterprise Server 11 SP3 64bit
Fedora	Fedora 29 64bit
	Fedora 24 64bit
Debian	Debian GNU/Linux 8.8.0 64bit
CentOS	CentOS 7.6 64bit
	CentOS 7.5 64bit
	CentOS 7.4 64bit

OS	OS Version
	CentOS 7.0 64bit
	CentOS 6.9 64bit
	CentOS 6.6 64bit
EulerOS	EulerOS 2.8 64bit
	EulerOS 2.5 64bit
	EulerOS 2.3 64bit
	EulerOS 2.2 64bit
openEuler	openEuler 20.03 64bit

1.3 Basic Concepts

1.3.1 Region and AZ

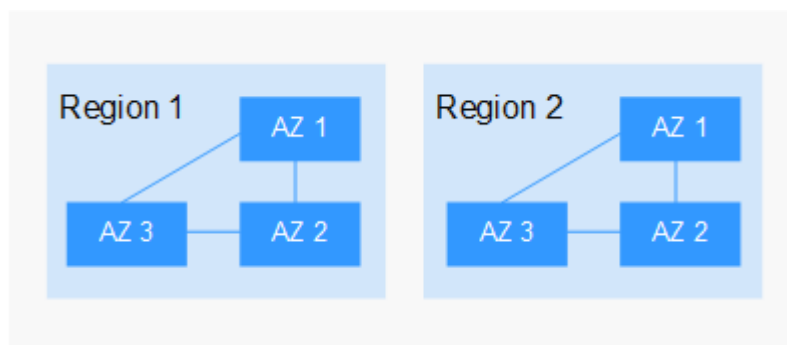
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- A region is a physical data center, which is completely isolated to improve fault tolerance and stability. The region that is selected during resource creation cannot be changed after the resource is created.
- An AZ is a physical location where resources use independent power supplies and networks. A region contains one or more AZs that are physically isolated but interconnected through internal networks. Because AZs are isolated from each other, any fault that occurs in one AZ will not affect others.

Figure 1-1 shows the relationship between regions and AZs.

Figure 1-1 Regions and AZs



Selecting a Region

Select a region closest to your target users for lower network latency and quick access.

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

Regions and Endpoints

Before you use an API to call resources, specify its region and endpoint. For more details, see [Regions and Endpoints](#).

1.3.2 Common Image Formats

IMS supports multiple image formats, but the system uses ZVHD or ZVHD2 by default.

[Table 1-3](#) lists the common image formats.

Table 1-3 Common image formats

Image Format	Description	Remarks
ZVHD	This format uses the ZLIB compression algorithm and supports sequential read and write.	A universal format supported by IaaS OpenStack; a format supported for imported and exported images
ZVHD2	This format uses the ZSTD algorithm and supports lazy loading.	A format for the lazy loading feature; a format supported for imported images

Image Format	Description	Remarks
QCOW2	This is a disk image supported by the QEMU simulator. It is a file that indicates a block device disk of a fixed size. Compared with the RAW format, the QCOW2 format has the following features: • Supports a lower disk usage. • Supports Copy-On-Write (CoW). The image file only reflects disk changes. • Supports snapshots. • Supports zlib compression and encryption by following Advanced Encryption Standard (AES).	A format supported for imported and exported images
VMDK	VMDK is a virtual disk format from VMware. A VMDK file represents a physical disk drive of the virtual machine file system (VMFS) on an ECS.	A format supported for imported and exported images
VHD	VHD is a virtual disk file format from Microsoft. A VHD file is a compressed file stored in the file system of the host machine. It mainly contains a file system required for starting ECSs.	A format supported for imported and exported images
VHDX	VHDX is a new VHD format introduced into Hyper-V of Windows Server 2012 by Microsoft. Compared with the VHD format, VHDX has a larger storage capacity. It provides protection against data damage during power supply failures, and the disk structure alignment has been optimized to prevent performance degradation of new physical disks in a large sector.	A format supported for imported images
RAW	A RAW file can be directly read and written by ECSs. This format does not support dynamic space expansion and has the best I/O performance.	A format supported for imported images

Image Format	Description	Remarks
QCOW	QCOW manages the space allocation of an image through the secondary index table. The secondary index uses the memory cache technology and needs the query operation, which results in performance loss. The performance of QCOW is inferior to that of QCOW2, and the read and write performance is inferior to that of RAW.	A format supported for imported images
VDI	VDI is the disk image file format used by the VirtualBOX virtualization software from Oracle. It supports snapshots.	A format supported for imported images
QED	The QED format is an evolved version of the QCOW2 format. Its storage location query mode and data block size are the same as those of the QCOW2 format. However, QED implements Copy-On-Write (CoW) in a different way as it uses a dirty flag to replace the reference count table of QCOW2.	A format supported for imported images

1.4 Related Services

Table 1-4 Related services

Service	Relationship with IMS	Related Operation
Elastic Cloud Server (ECS)	You can use an image to create ECSs or use an ECS to create an image.	• Creating an ECS from an Image • Creating a System Disk Image from a Windows ECS • Creating a System Disk Image from a Linux ECS
Bare Metal Server (BMS)	You can use an image to create BMSs or use a BMS to create an image.	Creating a BMS System Disk Image

Service	Relationship with IMS	Related Operation
Object Storage Service (OBS)	Images are stored in OBS buckets. External image files to be uploaded to the system are stored in OBS buckets, and private images are exported to OBS buckets.	Exporting Images
Key Management Service (KMS)	KMS provides the keys used for encrypting images.	Encrypting Images
Cloud Server Backup Service (CSBS)	You can use a CSBS backup to create a full-ECS image.	Creating a Full-ECS Image from a CSBS Backup
Tag Management Service (TMS)	You can add tags to images for convenient classification and search.	Tagging an Image
Cloud Trace Service (CTS)	CTS records IMS operations for query, auditing, or backtracking.	Auditing Key Operations

2 Creating a Private Image

2.1 Introduction

A private image is an image available only to the user who created it. It contains an OS, preinstalled public applications, and a user's personal applications. A private image can be a system disk image, data disk image, or full-ECS image. It can be created from a cloud server or an external image file.

This section describes how to create a private image using any of the following methods:

- [Creating a System Disk Image from a Windows ECS](#)
- [Creating a System Disk Image from a Linux ECS](#)
- [Creating a Windows System Disk Image from an External Image File](#)
- [Creating a Linux System Disk Image from an External Image File](#)
- [Creating a BMS System Disk Image](#)
- [Creating a Full-ECS Image from an ECS](#)
- [Creating a Full-ECS Image from a CSBS Backup](#)

2.2 Creating a System Disk Image from a Windows ECS

Scenarios

If you have created and configured a Windows ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software and environment preinstalled.

Prerequisites

Before creating a private image from an ECS:

- Delete any sensitive data the ECS may contain.

- Ensure that the ECS is in the **Running** or **Stopped** state.
- Check the network configuration of the ECS and ensure that DHCP is configured for the NICs. Enable remote desktop connection if needed. For details, see [Setting the NIC to DHCP](#) and [Enabling Remote Desktop Connection](#).
- Install special drivers. The normal running and advanced functions of some ECSs depend on certain drivers. For example, GPU-accelerated ECSs depend on the Tesla and GRID/vGPU drivers. For details, see [Installing Special Windows Drivers](#).
- Check whether Cloudbase-Init has been installed on the ECS. The user data injection function on the management console is only available for new ECSs that this tool is installed. You can use data injection to, for example, set the login password for a new ECS. For details, see [Installing and Configuring Cloudbase-Init](#).
- Check and install the PV driver and UVP VMTools driver to ensure that new ECSs created from the image support both KVM and XEN virtualization and to improve network performance.
For details, see steps [2](#) to [5](#) in [Optimization Process](#).
- Run Sysprep to ensure that the SIDs of the new ECSs created from the image are unique within their domain. In a cluster deployment scenario, the SIDs must be unique. For details, see [Running Sysprep](#).

 NOTE

If an ECS is created from a public image, Cloudbase-Init has been installed by default. You can follow the guide in the prerequisites to verify the installation.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. Set image parameters.

[Table 2-1](#) and [Table 2-2](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-1 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select ECS and select an ECS with required configurations.

Table 2-2 Image information

Parameter	Description
Encryption	This parameter specifies whether the image will be encrypted. The value is provided by the system and cannot be changed. • Only an unencrypted private image can be created from an unencrypted ECS. • Only an encrypted private image can be created from an encrypted ECS.
Name	Set a name for the image.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

5. Click **Create Now**.
6. Confirm the parameters and click **Submit**.
7. Go back to the private image list and monitor the image status.

The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks. When the image status changes to **Normal**, the image creation is complete.

NOTE

- Do not perform any operations on the selected ECS or its associated resources during image creation.
- An ECS created from an encrypted image is also encrypted. The key used for encrypting the ECS is the same as that used for encrypting the image.
- An image created from an encrypted ECS is also encrypted. The key used for encrypting the image is the same as that used for encrypting the ECS.

2.3 Creating a System Disk Image from a Linux ECS

Scenarios

If you have created and configured a Linux ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software and environment preinstalled.

Prerequisites

Before creating a private image from an ECS:

- Delete any sensitive data the ECS may contain.

- Ensure that the ECS is in the **Running** or **Stopped** state.
- Check network configuration of the ECS and ensure that DHCP is configured for the NICs. For details, see [Setting the NIC to DHCP](#).
- Check whether Cloud-Init has been installed on the ECS. The user data injection function on the management console is only available for new ECSs that have this tool installed. You can use data injection, for example, to set the login password for a new ECS. For details, see [Installing Cloud-Init](#) and [Configuring Cloud-Init](#).
- Delete any network rules to prevent NIC name drift on the ECSs created from the image. For details, see [Deleting Files in the Network Rule Directory](#).
- To ensure that the ECSs created from the image support both Xen and KVM virtualization, the Linux ECS used to create the image has to be modified. For instance, the disk IDs in the GRUB and fstab files need to be UUID and native Xen and KVM drivers need to be installed.
For details, see steps 2 to 6 in [Optimization Process](#).
- If multiple data disks are attached to an ECS used to create a private image, the ECSs created from the image may be unavailable. You need to detach all data disks from the ECS before using it to create an image. For details, see [Detaching Data Disks from an ECS](#).
- If data disks have been attached to the ECS and automatic partition mounting has been configured in the fstab file for the ECS, delete these configurations from the file before using the ECS to create a system disk image.

 NOTE

If an ECS is created from a public image, Cloud-Init has been installed by default. You can follow the guide to verify the installation.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. Set image parameters.

[Table 2-3](#) and [Table 2-4](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-3 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select ECS and select an ECS with required configurations.

Table 2-4 Image information

Parameter	Description
Encryption	This parameter specifies whether the image will be encrypted. The value is provided by the system and cannot be changed. - Only an unencrypted private image can be created from an unencrypted ECS. - Only an encrypted private image can be created from an encrypted ECS.
Name	Set a name for the image.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

- Click **Create Now**.
- Confirm the parameters and click **Submit**.
- Go back to the private image list and monitor the image status.

The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks. When the image status changes to **Normal**, the image creation is complete.

NOTE

- Do not perform any operations on the selected ECS or its associated resources during image creation.
- An ECS created from an encrypted image is also encrypted. The key used for encrypting the ECS is the same as that used for encrypting the image.
- An image created from an encrypted ECS is also encrypted. The key used for encrypting the image is the same as that used for encrypting the ECS.

2.4 Creating a Windows System Disk Image from an External Image File

2.4.1 Overview

You can import a local image or a system disk image from another cloud platform to the current cloud. After an image is imported, you can use it to create ECSs or reinstall the OSs of existing ECSs.

Creation Process

Figure 2-1 shows the process of creating a private image.

Figure 2-1 Creating a Windows system disk image



As shown in the figure, the following steps are required to register an external image file as a private image:

1. Prepare an external image file that meets platform requirements. For details, see [Preparing an Image File](#).
2. Upload the external image file to your OBS bucket. For details, see [Uploading an External Image File](#).
3. On the management console, select the uploaded image file and register it as a private image. For details, see [Registering an External Image File as a Private Image](#).
4. After the private image is registered, you can use it to create ECSs. For details, see [Creating a Windows ECS from an Image](#).

2.4.2 Preparing an Image File

You need to prepare an image file that meets the platform requirements.

NOTE

- You are advised to complete the network, tool, and driver configurations in [Table 2-5](#) on the ECS and then export the image file. You can also complete the configurations on the created ECSs. For details, see [What Do I Do If the Initial Configurations of a Windows External Image File Are Not Completed Before the File Is Exported?](#)
- Currently, only RAW and ZVHD2 files can be imported (maximum file size: 1 TB). In addition to the requirements described in [Table 2-5](#), a bitmap file needs to be generated for each RAW image file. The bitmap file is uploaded together with the image file. For details, see [Quickly Importing an Image File](#).

Table 2-5 Windows image file requirements

Image File Property	Requirement
OS	• Windows Server 2008, Windows Server 2012, Windows Server 2016 • 32-bit or 64-bit • The OS cannot be bound to specific hardware. • The OS must support full virtualization. For details about the supported OS versions, see Formats and OSs Supported for External Image Files. These OSs support automatic configuration. For details, see What Changes Will Be Made to an Image File Used for Registering a Private Image? For other OSs, check and install the Guest OS driver. On the image registration page, select Other Windows. After the image is imported, whether the system is started depends on the driver integrity.
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	Maximum file size: 128 GB If the image size is between 128 GB and 1 TB, convert the image file into the RAW or ZVHD2 format and import the image using fast import. • For details about how to convert the image file format, see Converting the Image Format Using qemu-img-hw. • For details about fast import, see fast image file import.
Network	The NIC must be set to DHCP. Otherwise, the ECS startup or network capability will be abnormal. For details, see: Setting the NIC to DHCP The following value-added operations are optional: • Enabling NIC multi-queue NIC multi-queue enables multiple CPUs to process NIC interruptions, thereby improving network PPS and I/O performance. For details, see How Do I Set NIC Multi-Queue for an Image?

Image File Property	Requirement
Tool	You are advised to install Cloudbase-Init. Cloudbase-Init is an open-source cloud initialization tool. When creating ECSs from an image with Cloudbase-Init, you can use user data injection to inject customized initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloudbase-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs. For details, see Installing and Configuring Cloudbase-Init.
Driver	• Installing the PV Driver • Installing UVP VMTools
Other requirements	• Currently, images with data disks cannot be created. The image file must contain only the system disk, and the system disk size must be [40 GB, 1024 GB]. • The initial password in the image file must contain uppercase letters, lowercase letters, digits, and special characters (!@\$%^&_-=+[]{};./?). • The boot partition and system partition must be on the same disk. • For an external image file, you need an administrator account and password combination. • Generally, the boot mode is BIOS in an image. Some OS images support the UEFI boot mode. For details, see "OSs Supporting UEFI Boot Mode" in <i>Image Service Management User Guide</i>. • The image file cannot be encrypted, or ECSs created from the registered image may not work properly.

2.4.3 Uploading an External Image File

You are advised to use OBS Browser to upload external image files to OBS buckets. For details, see *Object Storage Service User Guide*.

NOTE

- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to the OBS bucket.
- The storage class of the OBS bucket must be Standard.
- Do not upload image files using the OBS server-side encryption function.
- If you want to create a data disk image along with the system disk image, you need also to upload an image file containing data disks to the OBS bucket. You can create one system disk image and no more than three data disk images.

2.4.4 Registering an External Image File as a Private Image

Scenarios

Register an image file uploaded to the OBS bucket as a private image.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. Set image parameters.

[Table 2-6](#) and [Table 2-7](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-6 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select Image File for Source . Select the bucket storing the image file from the list and then select the image file.
Enable Fast Create	This parameter is available only when you select a ZVHD2 or RAW image file. This function enables fast image creation and supports import of large files (up to 1 TB) as long as the files to be uploaded are converted to ZVHD2 or RAW format and optimized. If you have a file that meets the requirements, select Enable Fast Create and select the confirmation information following Image File Preparation. NOTE To learn how to convert image file formats and generate bitmap files, see Quickly Importing an Image File.

Table 2-7 Image information

Parameter	Description
Enable automatic configuration	If you select this option, the system will automatically check and optimize the image file. For details, see What Changes Will Be Made to an Image File Used for Registering a Private Image?

Parameter	Description
Function	Specifies whether the image is used to create ECSs or BMSs. The value can be ECS system disk image or BMS system disk image . This section uses ECS system disk image as an example.
Boot Mode	This parameter is optional. The value can be BIOS or UEFI. For details about the differences between the two, see How Is BIOS Different from UEFI? For details about which OSs support UEFI boot, see OSs Supporting UEFI Boot Mode. The boot mode must be the same as that of the image file. You need to confirm which boot mode is used in the image file. After you select the correct boot mode, the boot mode of the image file will be configured at the background. Select the right boot mode, or ECSs created using the image will not be able to boot up.
OS	To ensure that the image can be created and used properly, select an OS consistent with that in the image file. If you do not select an OS, the system attempts to automatically identify the OS in the image file. NOTE • If the system detects that the image file OS is different from the one you selected, the OS detected by the system will be used. • If the system cannot detect the OS in the image file, the OS you selected will be used. • If the OS you selected or identified by the system is incorrect, ECSs created from the image file may be affected.
System Disk (GB)	The system disk capacity. Ensure that this value is at least equal to the system disk size in the image file. NOTE If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk size based on Why Does the Error Message Displayed on Task Center Indicates That the System Disk Size of the External Image File Exceeds the Maximum System Disk Size When a VHD Image File Failed to Be Uploaded?
Data Disk (GB)	You can also add data disks to the image. You need to obtain an image file containing data disks in advance. This function is used to migrate VMs and data disks from other platforms to the current platform. To add data disks, click , set the data disk size, and click Select Image File. In the displayed dialog box, select the target bucket and then the target image file containing the data disk. A maximum of three data disks can be added.

Parameter	Description
Name	Set a name for the image.
Encryption	(Optional) If you want to encrypt the image, select KMS encryption and select the key to be used from the key list. After you select KMS encryption , the system will create a default master private key ims/default for you. You can also select a private key from the private key list. For how to encrypt an image, see Creating Encrypted Images .
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

- Click **Create Now**, confirm the configurations, and click **Submit**.
- Go back to the **Private Images** page. The image is successfully registered when its status becomes **Normal**.

 NOTE

The time required for image registration is determined by the image file size. You may need to wait a long period of time for the image file to be successfully registered as a private image.

If you add data disks during image creation, a system disk image and data disk images will be generated. The number of data disk images depends on the number of data disks you add (a maximum of 3).

2.4.5 Creating a Windows ECS from an Image

Scenarios

After registering an external image file as a private image on the cloud platform, you can use the image to create ECSs or reinstall or change the OSs of existing ECSs. This section describes how to create an ECS from an image.

Procedure

You can create an ECS by referring to [Creating an ECS from an Image](#).

Note the following when setting the parameters:

- **Region:** Select the region where the private image is located.
- **Image:** Select **Private image** and then the created image from the drop-down list.
- (Optional) **Data Disk:** Add data disks. These data disks are created from a data disk image generated together with a system disk image. In this way, you can migrate the data of data disks together with system disk data from the VM on the original platform to the current cloud platform.

2.5 Creating a Linux System Disk Image from an External Image File

2.5.1 Overview

You can import a local image or a system disk image from another cloud platform to the current cloud. After an image is imported, you can use it to create ECSs or reinstall the OSs of existing ECSs.

Creation Process

Figure 2-2 shows the process of creating a private image.

Figure 2-2 Creating a Linux system disk image



The procedure is as follows:

1. Prepare an external image file that meets platform requirements. For details, see [Preparing an Image File](#).
2. Upload the external image file to your OBS bucket. For details, see [Uploading an External Image File](#).
3. On the management console, select the uploaded image file and register it as a private image. For details, see [Registering an External Image File as a Private Image](#).
4. After the private image is registered, you can use it to create ECSs. For details, see [Creating a Linux ECS from an Image](#).

2.5.2 Preparing an Image File

You need to prepare an image file that meets the platform requirements.

 NOTE

- You are advised to complete the file system, network, and driver configurations in [Table 2-8](#) on the VM and then export the image file. You can also complete the configurations on the created ECSs. For details, see [What Do I Do If the Initial Configurations of a Linux External Image File Are Not Completed Before the File Is Exported?](#)
- Currently, only RAW and ZVHD2 files can be imported (maximum file size: 1 TB). In addition to the requirements described in [Table 2-8](#), a bitmap file needs to be generated for each RAW image file. The bitmap file is uploaded together with the image file. For details, see [Quickly Importing an Image File](#).

Table 2-8 Linux image file requirements

Image File Property	Requirement
OS	• SUSE, Oracle Linux, Red Hat, Ubuntu, openSUSE, CentOS, Debian, Fedora, EulerOS, and NeoKylin • 32-bit or 64-bit • The OS cannot be bound to specific hardware. • The OS must support full virtualization. For details about the supported OS versions, see Formats and OSs Supported for External Image Files. These OSs support automatic configuration. For details, see What Changes Will Be Made to an Image File Used for Registering a Private Image? For other OSs, check and install the VirtIO driver. On the image registration page, select Other Linux. After the image is imported, whether the system is started depends on the driver integrity.
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	Maximum file size: 128 GB If the image size is between 128 GB and 1 TB, convert the image file into the RAW or ZVHD2 format and import the image using fast import. • For details about how to convert the image file format, see Converting the Image Format Using qemu-img-hw. • For details about fast import, see fast image file import.

Image File Property	Requirement
Network	The NIC must be set to DHCP and files must be deleted from the network role directory. Otherwise, the ECS startup or network capability will be abnormal. For details, see: • Deleting files in the network rule directory • Setting the NIC to DHCP The following value-added operations are optional: • Enabling NIC multi-queue NIC multi-queue enables multiple CPUs to process NIC interruptions, thereby improving network PPS and I/O performance. For details, see How Do I Set NIC Multi-Queue for an Image?
Tool	You are advised to install Cloud-Init. Cloud-Init is an open-source cloud initialization tool. When creating ECSs from an image with Cloud-Init, you can use user data injection to inject customized initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloud-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs. For details, see Installing Cloud-Init.
Driver	Installing native KVM drivers
File system	• Changing the disk identifier in the GRUB configuration file to UUID • Changing the disk identifier in the fstab file to UUID

Image File Property	Requirement
Other requirements	- Currently, images with data disks cannot be created. The image file must contain only the system disk, and the system disk size must be [40 GB, 1024 GB]. - The initial password in the image file must contain uppercase letters, lowercase letters, digits, and special characters (!@\$%^&_+=+[{ }],./?). - The boot partition and system partition must be on the same disk. - Generally, the boot mode is BIOS in an image. Some OS images support the UEFI boot mode. For details, see "OSs Supporting UEFI Boot Mode" in <i>Image Service Management User Guide</i>. - The image file cannot be encrypted, or ECSs created from the registered image may not work properly. - The /etc/fstab file cannot contain automatic mounting information of non-system disks. Otherwise, the login to the created ECS may fail. - If the external image file uses LVM as the system disk, ECSs created from the private image do not support file injection. - If the VM where the external image file is located has been shut down, it must be a graceful shutdown. Otherwise, a blue screen may occur when the ECS created from the private image is started.

2.5.3 Uploading an External Image File

You are advised to use OBS Browser to upload external image files to OBS buckets. For details, see *Object Storage Service User Guide*.

NOTE

- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to the OBS bucket.
- The storage class of the OBS bucket must be Standard.
- Do not upload image files using the OBS server-side encryption function.
- If you want to create a data disk image along with the system disk image, you need also to upload an image file containing data disks to the OBS bucket. You can create one system disk image and no more than three data disk images.

2.5.4 Registering an External Image File as a Private Image

Scenarios

Register an image file uploaded to the OBS bucket as a private image.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. Set image parameters.

[Table 2-9](#) and [Table 2-10](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-9 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select Image File for Source . Select the bucket storing the image file from the list and then select the image file.
Enable Fast Create	This parameter is available only when you select a ZVHD2 or RAW image file. This function enables fast image creation and supports import of large files (up to 1 TB) as long as the files to be uploaded are converted to ZVHD2 or RAW format and optimized. If you have a file that meets the requirements, select Enable Fast Create and select the confirmation information following Image File Preparation. NOTE To learn how to convert image file formats and generate bitmap files, see Quickly Importing an Image File.

Table 2-10 Image information

Parameter	Description
Enable automatic configuration	If you select this option, the system will automatically check and optimize the image file. For details, see What Changes Will Be Made to an Image File Used for Registering a Private Image?
Function	Specifies whether the image is used to create ECSs or BMSs. The value can be ECS system disk image or BMS system disk image . This section uses ECS system disk image as an example.

Parameter	Description
Boot Mode	This parameter is optional. The value can be BIOS or UEFI. For details about the differences between the two, see How Is BIOS Different from UEFI? For details about which OSs support UEFI boot, see OSs Supporting UEFI Boot Mode. The boot mode must be the same as that of the image file. You need to confirm which boot mode is used in the image file. After you select the correct boot mode, the boot mode of the image file will be configured at the background. Select the right boot mode, or ECSs created using the image will not be able to boot up.
OS	To ensure that the image can be created and used properly, select an OS consistent with that in the image file. If you do not select an OS, the system attempts to automatically identify the OS in the image file. NOTE • If the system detects that the image file OS is different from the one you selected, the OS detected by the system will be used. • If the system cannot detect the OS in the image file, the OS you selected will be used. • If the OS you selected or identified by the system is incorrect, ECSs created from the image file may be affected.
System Disk (GB)	The system disk capacity. Ensure that this value is at least equal to the system disk size in the image file. NOTE If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk size based on Why Does the Error Message Displayed on Task Center Indicates That the System Disk Size of the External Image File Exceeds the Maximum System Disk Size When a VHD Image File Failed to Be Uploaded?
Data Disk (GB)	You can also add data disks to the image. You need to obtain an image file containing data disks in advance. This function is used to migrate VMs and data disks from other platforms to the current platform. To add data disks, click , set the data disk size, and click Select Image File. In the displayed dialog box, select the target bucket and then the target image file containing the data disk. A maximum of three data disks can be added.
Name	Set a name for the image.

Parameter	Description
Encryption	(Optional) If you want to encrypt the image, select KMS encryption and select the key to be used from the key list. After you select KMS encryption , the system will create a default master private key ims/default for you. You can also select a private key from the private key list. For how to encrypt an image, see Creating Encrypted Images .
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

- Click **Create Now**, confirm the configurations, and click **Submit**.
- Go back to the **Private Images** page. The image is successfully registered when its status becomes **Normal**.

NOTE

The time required for image registration is determined by the image file size. You may need to wait a long period of time for the image file to be successfully registered as a private image.

If you add data disks during image creation, a system disk image and data disk images will be generated. The number of data disk images depends on the number of data disks you add (a maximum of 3).

2.5.5 Creating a Linux ECS from an Image

Scenarios

After registering an external image file as a private image on the cloud platform, you can use the image to create ECSs or reinstall or change the OSs of existing ECSs. This section describes how to create an ECS from an image.

Procedure

You can create an ECS by referring to [Creating an ECS from an Image](#).

Note the following when setting the parameters:

- Region:** Select the region where the private image is located.
- Image:** Select **Private image** and then the created image from the drop-down list.
- (Optional) **Data Disk:** Add data disks. These data disks are created from a data disk image generated together with a system disk image. In this way, you can migrate the data of data disks together with system disk data from the VM on the original platform to the current cloud platform.

2.6 Creating a BMS System Disk Image

For how to create a private BMS image, see *Bare Metal Server User Guide (Paris Region)*.

2.7 Creating a Full-ECS Image from an ECS

Scenarios

You can create an image of an entire ECS, including not just the OS, but also the software and all the service data. You can then use this image to migrate data by quickly provisioning exact clones of the original ECS.

Constraints

- When creating a full-ECS image from an ECS, ensure that the ECS has been properly configured. Otherwise, creating ECSs using the full-ECS image may fail.
- A Windows ECS used to create a full-ECS image cannot have a spanned volume. Otherwise, data may be lost when the full-ECS image is used to create ECSs.
- A Linux ECS used to create a full-ECS image cannot have a disk group or logical disk that contains multiple physical disks. Otherwise, data may be lost when the full-ECS image is used to create ECSs.
- An ECS used to create a full-ECS image cannot contain a Dedicated Storage Service disk.
- A full-ECS image cannot be exported, replicated, or shared.
- When creating a full-ECS image from a Windows ECS, you need to change the SAN policy of the ECS to OnlineAll. Otherwise, EVS disks attached to the ECSs created from the image may be offline.

Windows has three types of SAN policies: **OnlineAll**, **OfflineShared**, and **OfflineInternal**.

Table 2-11 SAN policies in Windows

Type	Description
OnlineAll	All newly detected disks are automatically brought online.
OfflineShared	All disks on sharable buses, such as iSCSI and FC, are left offline by default, while disks on non-sharable buses are kept online.
OfflineInternal	All newly detected disks are left offline.

- Execute **cmd.exe** and run the following command to query the current SAN policy of the ECS:

diskpart

- b. Run the following command to view the SAN policy of the ECS:
san
 - If the SAN policy is **OnlineAll**, run the **exit** command to exit DiskPart.
 - If the SAN policy is not **OnlineAll**, go to **c**.
- c. Run the following command to change the SAN policy of the ECS to **OnlineAll**:
san policy=onlineall

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. In the **Image Type and Source** area, select **Full-ECS image** for **Type**.
5. Select **ECS** for **Source** and then select an ECS from the list.

Figure 2-3 Creating a full-ECS image using an ECS



6. In the **Image Information** area, configure basic image information, such as the image name and description.
7. Click **Create Now**.
8. Confirm the parameters and click **Submit**.
9. Switch back to the **Image Management Service** page to view the image status.

When the image status changes to **Normal**, the creation is successful.

Follow-up Operations

If you want to use the full-ECS image to create ECSs, click **Apply for Server** in the **Operation** column. On the displayed page, create ECSs by following the instructions in *Elastic Cloud Server User Guide*.

NOTE

If a full-ECS image contains one or more data disks, the system automatically configures data disk parameters when you use the image to create ECSs.

2.8 Creating a Full-ECS Image from a CSBS Backup

Scenarios

You can use a CSBS backup to create a full-ECS image, which can be used to create ECSs.

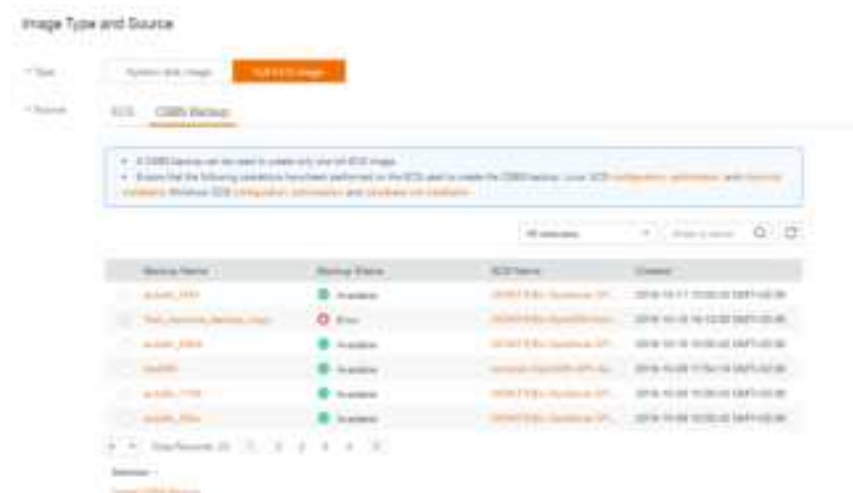
Constraints

- When creating a full-ECS image from a CSBS backup, ensure that the source ECS of the CSBS backup has been properly configured. Otherwise, creating ECSs using the full-ECS image may fail.
- A CSBS backup used to create a full-ECS image cannot have shared disks.
- Only an available CSBS backup can be used to create a full-ECS image. A CSBS backup can be used to create only one full-ECS image.
- A full-ECS image cannot be exported, replicated, or shared.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click **Create Image** in the upper right corner.
4. In the **Image Type and Source** area, select **Full-ECS image** for **Type**.
5. Select **CSBS Backup** for **Source** and then select a backup from the list.

Figure 2-4 Creating a full-ECS image using a CSBS backup



6. In the **Image Information** area, configure basic image information, such as the image name and description.
7. Click **Create Now**.
8. Confirm the parameters and click **Submit**.

9. Switch back to the **Image Management Service** page to view the image status.

When the image status changes to **Normal**, the creation is successful.

Follow-up Operations

If you want to use the full-ECS image to create ECSs, click **Apply for Server** in the **Operation** column. On the displayed page, create ECSs by following the instructions in *Elastic Cloud Server User Guide*.

NOTE

If a full-ECS image contains one or more data disks, the system automatically configures data disk parameters when you use the image to create ECSs.

2.9 Quickly Importing an Image File

2.9.1 Overview

If the size of an external image file is greater than 128 GB, you can import the image file through fast import. Only the RAW and ZVHD2 formats support fast import. The image file size cannot exceed 1 TB.

Import Solution

Select a proper import solution based on the image file format.

- If the file format is ZVHD2, the import solution is as follows: Optimize the image file > Upload the image file to the OBS bucket > Register the image file on the cloud platform.
- If the file format is RAW, the import solution is as follows: Optimize the image file > Generate a bitmap file for the image file > Upload the image file and bitmap file to the OBS bucket > Register the image file on the cloud platform.
- If the file is not in the ZVHD2 or RAW format, import the file in either of the following ways:
 - Optimize the image file > Convert the image file format to ZVHD2 > Upload the image file to the OBS bucket > Register the image file on the cloud platform
 - Optimize the image file > Convert the image file format to RAW and generate a bitmap file for the image file > Upload the image file and bitmap file to the OBS bucket > Register the image file on the cloud platform

NOTE

- Fast import is used to import large files. The import of large files depends on the lazy loading feature. The ZVHD2 format supports this feature but the RAW does not. The import of RAW files depends on a bitmap file. Therefore, you need to upload a bitmap file together with the RAW file.
- For details about how to optimize an image file, see [Optimization Process](#) or [Optimization Process](#) depending on the OS type of the image file.

Import Process

The following describes how to import an external image file in a format other than ZVHD2 or RAW.

You can use **qemu-img-hw** or the open-source tool **qemu-img** to convert the image format. **qemu-img-hw** is used only in Linux. This document provides guidance for importing external image files in both Linux and Windows.

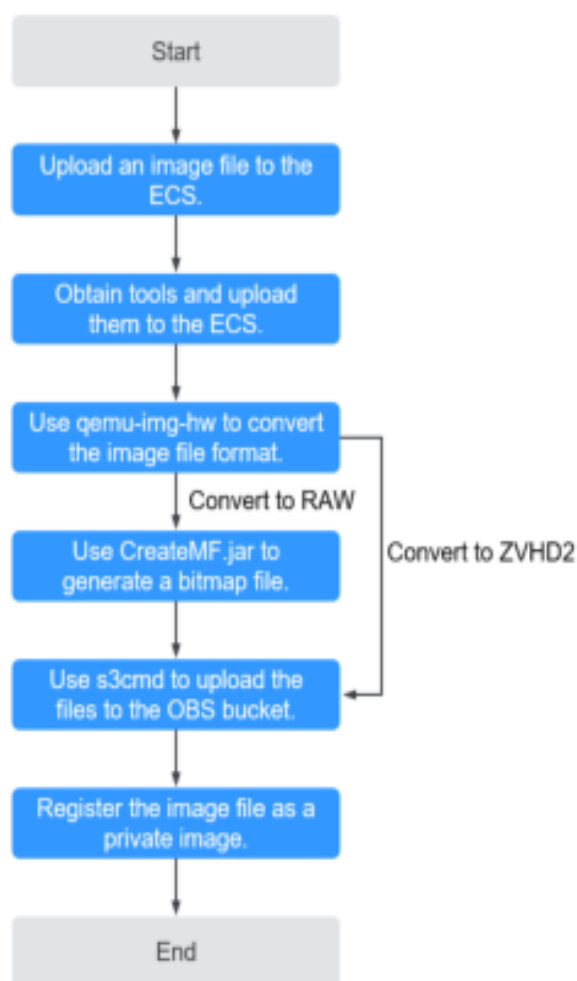
NOTE

The fast import tool consists of **qemu-img-hw** (for converting image formats) and **CreateMF.jar** (for generating bitmap files).

- Linux

You are advised to use an EulerOS ECS on the cloud platform. [Figure 2-5](#) shows the import process.

Figure 2-5 Import process (Linux)



For details, see [Quickly Importing an Image File \(Linux\)](#).

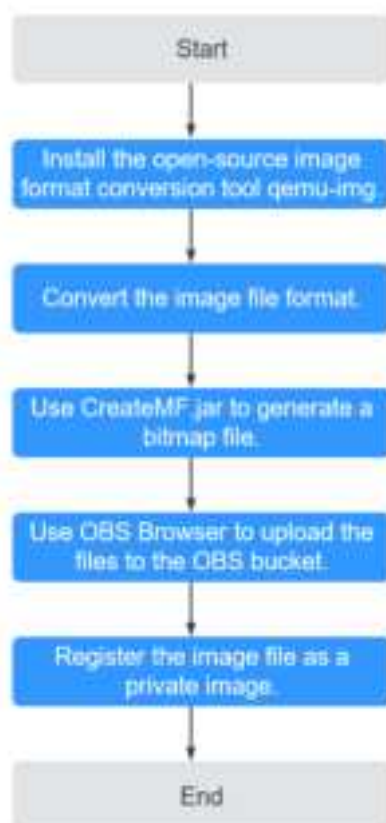
- Windows

You are advised to use a local PC running Windows. [Figure 2-6](#) shows the import process.

 NOTE

qemu-img cannot convert image files to the ZVHD2 format. Therefore, you need to convert the image files to the RAW format and then use **CreateMF.jar** to generate bitmap files.

Figure 2-6 Import process (Windows)



For details, see [Quickly Importing an Image File \(Windows\)](#).

2.9.2 Quickly Importing an Image File (Linux)

Scenarios

This section describes how to quickly import an image file in Linux. You are advised to use an EulerOS ECS on the cloud platform for converting image formats and generating bitmap files.

In Linux, you are advised to use **qemu-img-hw** to convert image formats.

Prerequisites

- The image file has been optimized. For details, see [Optimization Process](#) or [Optimization Process](#). In addition, ensure that the image file meets the requirements in [Table 2-5](#) or [Table 2-8](#).

NOTE

Select the reference content based on the OS type in the image file.

- You have created an ECS running EulerOS on the management console and bound an EIP to the ECS.
- An OBS bucket has been created on the management console.

Procedure

Step 1 Upload the image file to the ECS.

- If the local host runs a Linux OS, run the **scp** command.
For example, to upload **image01.qcow2** to the **/usr/** directory on the ECS, run the following command:
scp /var/image01.qcow2 root@xxx.xxx.xx.xxx:/usr/
xxx.xxx.xx.xxx indicates the EIP bound to the ECS.
- If the local host runs a Windows OS, use a file transfer tool, such as WinSCP, to upload the image file to the ECS.

Step 2 Obtain the fast import tool package, upload it to the ECS, and then decompress the package.

Obtain the fast import tool from the following link in the table.

Table 2-12 Fast import tool package

Tool Package	How to Obtain
quick-import-tools	https://oss.prod-cloud-ocb.orange-business.com/download/ims/OCB-image-convert-tools.zip

Step 3 Use **qemu-img-hw** to convert the image format.

- Go to the directory where **qemu-img-hw** is stored, for example, **/usr/quick-import-tools/qemu-img-hw**.

cd /usr/quick-import-tools/qemu-img-hw

- Run the following command to change file permissions:

chmod +x qemu-img-hw

- Run the **qemu-img-hw** command to convert the image file to the ZVHD2 (recommended) or RAW format.

The command format of **qemu-img-hw** is as follows:

./qemu-img-hw convert -p -O Target_image_format Source_image_file Target_image_file

For example, run the following command to convert an **image01.qcow2** file to an **image01.zvhd2** file:

```
./qemu-img-hw convert -p -O zvhd2 image01.qcow2 image01.zvhd2
```

- If the image file is converted to the ZVHD2 format, go to [Step 5](#).
- If the image file is converted to the RAW format, go to [Step 4](#).

Step 4 Use **CreateMF.jar** to generate a bitmap file.

1. Ensure that JDK has been installed on the ECS.

Run the following command to check whether JDK is installed:

```
source /etc/profile
```

```
java -version
```

If the Java version is displayed, JDK has been installed.

2. Run the following command to enter the directory where **CreateMF.jar** is stored:

```
cd /usr/quick-import-tools/createMF
```

3. Run the following command to generate a bitmap file:

```
java -jar CreateMF.jar /Original RAW file path /Generated .mf file path
```

Example:

```
java -jar CreateMF.jar image01.raw image01.mf
```



CAUTION

The generated bitmap file must have the same name as the RAW image file. For example, if the image file name is **image01.raw**, the generated bitmap name is **image01.mf**.

Step 5 Use **s3cmd** to upload files to the OBS bucket.

1. Install **s3cmd**.

If **s3cmd** has been installed, skip this step.

- a. Run the following command to install **setuptools**:

```
yum install python-setuptools
```

- b. Run the following command to install **wget**:

```
yum install wget
```

- c. Run the following commands to obtain the **s75pxd** software package:

```
wget https://github.com/s3tools/s3cmd/archive/master.zip
```

```
mv master.zip s3cmd-master.zip
```

- d. Run the following commands to install **s3cmd**:

```
unzip s3cmd-master.zip
```

```
cd s3cmd-master
```

```
python setup.py install
```

2. Configure **s3cmd**.

Run the following command to configure **s3cmd**:

```
s3cmd --configure
```

Access Key: Enter the AK.

Secret Key: Enter the SK.

Default Region: Enter the region where the bucket is located.
S3 Endpoint: Refer to the OBS endpoint.
DNS-style bucket+hostname:port template for accessing a bucket: Enter a server address with a bucket name, for example, **mybucket.obs.myclouds.com**.
Encryption password: Press **Enter**.
Path to GPG program: Press **Enter**.
Use HTTPS protocol: Specifies whether to use HTTPS. The value can be **Yes** or **No**.
HTTP Proxy server name: Specifies the proxy address used to connect the cloud from an external network. (If you do not need connect to the cloud, press **Enter**.)
HTTP Proxy server port: Specifies the proxy port used to connect to the cloud from an external network (If you do not need connect to the cloud, press **Enter**.)
Test access with supplied credentials? **y**
(If **Success. Your access key and secret key worked fine :-)** is displayed, the connection is successful.)
Save settings? **y** (Specifies whether to save the configuration. If **y** is entered, the configuration will be saved.)

 NOTE

The configuration is stored in the **/root/.s3cfg** directory. If you want to modify the configuration, run the **s3cmd --configure** command again or directly edit the **.s3cfg** file by running the **vi .s3cfg** command.

3. Run the **s3cmd** command to upload the ZVHD2 image file to the OBS bucket, or upload the RAW image file and its bitmap file to the OBS bucket.

```
s3cmd put image01.zvhd2 s3://mybucket/
```

 CAUTION

The **.mf** file must be in the same OBS bucket as the RAW image file.

Step 6 Register a private image.

You can register a private image using the converted ZVHD2 or RAW file on the console or using an API.

Method 1: Create a private image on the console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. In the upper right corner, click **Create Image**.
4. In the **Image Type and Source** area, select **System disk image** or **Data disk image** for **Type**.
5. Select **Image File** for **Source**. Select the bucket storing the ZVHD2 or RAW image file and then select the image file. If the image file is in the RAW format, you also need to select its bitmap file.
6. Select **Enable Fast Create**, ensure that the image file has been optimized, and select the sentence following **Image File Preparation**.
7. Set parameters as prompted.

For details about the parameters, see [Registering an External Image File as a Private Image](#) and [Registering an External Image File as a Private Image](#).

 CAUTION

- The OS must be the same as that in the image file.
- The size of the system disk must be greater than the size in the image file.

You can use the **qemu-img-hw** tool to query for the image file size.

qemu-img-hw info test.zvhd2

Method 2: Create a private image using an API.

You can use the POST /v2/cloudimages/quickimport/action API to quickly import an image file.

For details about how to call this API, see "Importing an Image File Quickly" in *Image Management Service API Reference*.

----End

2.9.3 Quickly Importing an Image File (Windows)

Scenarios

This section describes how to quickly import an image file in Windows. You are advised to use a local PC running Windows for converting image formats and generating bitmap files.

In Windows, use the open-source tool **qemu-img** to convert image formats. **qemu-img** supports conversion between image files of the VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED formats. Therefore, convert an image to the RAW format and then use the **CreateMF.jar** tool to generate a bitmap file.

Prerequisites

- The image file has been optimized. For details, see [Optimization Process](#) or [Optimization Process](#). In addition, ensure that the image file meets the requirements in [Table 2-5](#) or [Table 2-8](#).

 NOTE

Select the reference content based on the OS type in the image file.

- An OBS bucket has been created on the management console, and OBS Browser has been downloaded.

Procedure

Step 1 Install the open-source image conversion tool **qemu-img**.

Step 2 Run the **cmd** command to go to the **qemu-img** installation directory and run the **qemu-img** command to convert the image file to the RAW format.

For example, run the following command to convert an **image.qcow2** file to an **image.raw** file:

qemu-img convert -p -O raw image.qcow2 image.raw

Step 3 Use **CreateMF.jar** to generate a bitmap file.

1. Obtain the **CreateMF.jar** package and decompress it.

Table 2-13 CreateMF.jar package

Tool Package	How to Obtain
quick-import-tools	https://oss.prod-cloud-ocb.orange-business.com/download/ims/OCB-image-convert-tools.zip

2. Ensure that JDK has been installed in the current environment.
You can verify the installation by performing the following operation:
Run **cmd.exe** and then **java -version**. If Java version information is displayed, JDK has been installed.
3. Go to the directory where **CreateMF.jar** is stored.
For example, if you have downloaded **CreateMF.jar** to **D:/test**, run the following commands to access the directory:
D:
cd test
4. Run the following command to generate the bitmap file corresponding to the image file in the RAW format:
java -jar CreateMF.jar D:/image01.raw D:/image01.mf

Step 4 Use OBS Browser to upload files to the OBS bucket.

You must upload the RAW image file and its bitmap file to the same OBS bucket.

Step 5 Register a private image.

You can register a private image using the converted ZVHD2 or RAW file on the console or using an API.

Method 1: Create a private image on the console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. In the upper right corner, click **Create Image**.
4. In the **Image Type and Source** area, select **System disk image** or **Data disk image** for **Type**.
5. Select **Image File** for **Source**. Select the bucket storing the ZVHD2 or RAW image file and then select the image file. If the image file is in the RAW format, you also need to select its bitmap file.
6. Select **Enable Fast Create**, ensure that the image file has been optimized, and select the sentence following **Image File Preparation**.
7. Set parameters as prompted.
For details about the parameters, see [Registering an External Image File as a Private Image](#) and [Registering an External Image File as a Private Image](#).

 CAUTION

- The OS must be the same as that in the image file.
- The size of the system disk must be greater than the size in the image file.

You can use the **qemu-img-hw** tool to query for the image file size.

qemu-img-hw info test.zvhd2

Method 2: Create a private image using an API.

You can use the POST `/v2/cloudimages/quickimport/action` API to quickly import an image file.

For details about how to call this API, see "Importing an Image File Quickly" in *Image Management Service API Reference*.

----**End**

3 Managing Private Images

3.1 Modifying Image Information

Scenarios

You can modify the following information of a private image.

- Name
- Description
- Minimum Memory
- Maximum Memory
- NIC Multi-Queue

NIC multi-queue enables multiple CPUs to process NIC interruptions for load balancing. For details, see [How Do I Set NIC Multi-Queue for an Image?](#)

Constraints

- You can only modify the attributes of a private image in the **Normal** state.
- For a data disk image, you can only change its name and description.

Procedure

Use any of the following methods to modify image attributes.

Method 1:

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. Locate the row that contains the image and click **Modify** in the **Operation** column.
5. In the **Modify Image** dialog box, modify image attributes.

Method 2:

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. On the image list, click the name of the target image.
5. On the image details page, click **Modify** in the upper right corner. In the **Modify Image** dialog box, modify image attributes.

3.2 Checking the Disk Capacity of an Image

Scenarios

After a private image is created, you can check the disk capacity of the image.

- To check the disk capacity of a system disk image, data disk image, or ISO image, see [Check the Disk Capacity of a System Disk Image, Data Disk Image, or ISO Image](#).
- To check the disk capacity of a full-ECS image, see [Check the Disk Capacity of a Full-ECS Image](#).

Check the Disk Capacity of a System Disk Image, Data Disk Image, or ISO Image

You can view the disk capacity of a system disk image, data disk image, or ISO image in the **Disk Capacity** column in the private image list.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. In the **Disk Capacity** column of the row that contains the image, view the disk capacity value. The unit is **GB**.

Check the Disk Capacity of a Full-ECS Image

The disk capacity of a full-ECS image is the sum of the system disk capacity and data disk capacity in the backup from which the full-ECS image is created.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
In the **Disk Capacity** column of the row that contains the full-ECS image, the value is --.
4. Click the full-ECS image name.
5. Click the **Backups** tab and view the capacities of the system disk and data disks in the backup.

Disk capacity of a full-ECS image = Capacity of the system disk in the backup
+ Capacity of data disks in the backup

For example:

- If the system disk capacity is 40 GB and no data disk is attached, the capacity of the full-ECS image disk is 40 GB.
- If the system disk capacity is 40 GB and the data disk capacity is 50 GB, the full-ECS image disk capacity is 90 GB.

3.3 Creating an ECS from an Image

Scenarios

You can use a public, private, or shared image to create an ECS. The differences are as follows:

- If you use a public image, the created ECS contains an OS and pre-installed public applications. You need to install applications as needed.
- If you use a private or shared image, the created ECS contains an OS, pre-installed public applications, and a user's personal applications.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Public Images**, **Private Images**, or **Images Shared with Me** tab to display the image list.
4. Locate the row that contains the private image and click **Apply for Server** in the **Operation** column.
5. For details about how to create an ECS, see *Elastic Cloud Server User Guide*.
When you use a private system disk image to create an ECS, you can modify the ECS specifications or change the system disk type, but the system disk can only be larger than that in the image.

3.4 Deleting Images

Scenarios

You can delete private images that are no longer needed.

Constraints

Private images that have been published in Marketplace cannot be deleted.

Procedure

1. Log in to the management console.

2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. Locate the row that contains the image, choose **More > Delete** in the **Operation** column.

 NOTE

To delete multiple images, perform the following operations:

1. Select the images you want to delete in the image list.
 2. Click **Delete** above the image list.
5. (Optional) Select **Delete CSBS backups of the full-ECS images**.
This parameter is available only when the private images to be deleted include full-ECS images.
If you select this option, the system deletes CSBS backups of the full-ECS images.

 NOTE

After you perform the operation to delete the images, a case may occur where full-ECS images are successfully deleted, but their CSBS backups are not deleted. This may be because the CSBS backups are being created and cannot be deleted. In this case, delete the CSBS backups as prompted.

6. Click **Yes**.

3.5 Sharing Images

3.5.1 Overview

You can share your private images with other tenants. The tenants who accept the shared images can use the images to create ECSs of the same specifications.

Constraints

- You can share images only within the region where they reside.
- Each image can be shared with a maximum of 128 tenants.
- Encrypted images cannot be shared.
- Full-ECS images cannot be shared.

Procedure

If you want to share a private image with another tenant, the procedure is as follows:

1. You obtain the project ID from the tenant.
2. You share an image with the tenant.
3. The tenant accepts the shared image.

After accepting the image, the tenant can use it to create ECSs.

Related FAQs

If you have any questions, see [Image Sharing](#).

3.5.2 Obtaining the Project ID

Scenarios

Before a tenant shares an image with you, you need to provide your project ID.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the username in the upper right corner and select **My Credentials** from the drop-down list.

On the **My Credentials** page, view the project ID in the project list.

3.5.3 Sharing Specified Images

Scenarios

After obtaining the project ID from a tenant, you can share specified private images with the tenant. You can share a single image or multiple images as needed.

Prerequisites

You have obtained the project ID from the target tenant.

Procedure

- Share multiple images.
 - a. Log in to the management console.
 - b. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
 - c. Click the **Private Images** tab.
 - d. Select the private images to share and click **Share** above the image list.
 - e. In the **Share Image** dialog box, enter the project ID of the target tenant and click **OK**.

To share images with more than one tenant, enter their project IDs and separate them with commas. Then, click **OK**.

NOTE

You can enter a maximum of 100 project IDs at a time.

- Share a single image.
 - a. Log in to the management console.

- b. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
- c. Click the **Private Images** tab.
- d. Locate the row that contains the private image you are to share, click **More** in the **Operation** column, and select **Share** from the drop-down list.
- e. In the **Share Image** dialog box, enter the project ID of the target tenant and click **OK**.
To share images with more than one tenant, enter their project IDs and separate them with commas. Then, click **OK**.

 NOTE

You can enter a maximum of 100 project IDs at a time.

Related Operations

After you share images with a tenant, the tenant can accept the shared images on the **Images Shared with Me** page on the IMS console. For detailed operations, see [Accepting or Rejecting Shared Images](#).

3.5.4 Accepting or Rejecting Shared Images

Scenarios

After another tenant shares images with you, you will receive a message. You can choose to accept or reject all or some of the shared images.

 NOTE

You can receive the message only when you are in the same region where the tenant shares the images with you.

Prerequisites

Another tenant has shared images with you.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Images Shared with Me** tab.
A message is displayed above the image list asking you whether to accept the shared images.
 - To accept all the shared images, click **Accept All** in the upper right corner.
 - To accept some images, select the images and click **Accept**.
 - To reject some images, select the images and click **Reject**.

Result

- **Pending:** If you do not accept or reject a shared image in time, the image is in the **Pending** state.
A pending shared image is not displayed in the shared image list.
- **Accepted:** After an image is accepted, it is displayed in the shared image list. You can use the image to create ECSs.
- **Rejected:** After an image is rejected, it is not displayed in the shared image list. You can click **Rejected Images** to view the images you have rejected and accept them.

3.5.5 Rejecting Accepted Images

Scenarios

You can reject accepted images if you no longer need them.

After an image is rejected, it will not be displayed on the **Images Shared with Me** page.

Prerequisites

You have accepted images shared by other users.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Images Shared with Me** tab.
4. Determine the next step based on how many images you are to reject.
 - To reject multiple images: select the images to be rejected and click **Reject** above the image list. In the displayed dialog box, click **Yes**.
 - To reject a specific image: locate the image to be rejected and click **Reject** in the **Operation** column. In the displayed dialog box, click **Yes**.

3.5.6 Accepting Rejected Images

Scenarios

If you want to use the shared images you have rejected, you can accept them from the list of rejected images.

Prerequisites

- You have rejected the images shared by others.
- The image owners have not stopped sharing the images.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Images Shared with Me** tab.
4. Click **Rejected Images**. All the rejected images are displayed.
5. Select the images you want to accept and click **Accept**.
6. Check the accepted images in the shared image list.

3.5.7 Stopping Sharing Images

Scenarios

You can stop sharing images. After you stop the sharing of an image:

- The image will be invisible to the recipient on the management console and no data will be returned when the recipient query the image through an API.
- The recipient cannot use the image to create an ECS or EVS disk, or change the OS of an ECS.
- The recipient cannot reinstall the OS of the ECSs created from the shared image or create instances identical with these ECSs.

Prerequisites

You have shared private images with others.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab.
4. Locate the row that contains the private image that you no longer want to share, and choose **More > Share** in the **Operation** column.
5. In the **Share Image** dialog box, click the **Stop Sharing** tab.
6. Select the domain name that you want to stop image sharing and click **OK**.

3.5.8 Adding Tenants Who Can Use Shared Images

Scenarios

You can add tenants who can use the images you have shared.

Prerequisites

- You have shared private images.
- You have obtained the project IDs of the tenants to be added.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab.
4. Click the image name to view image details.
5. Click **Add Tenant**.
6. In the **Add Tenant** dialog box, enter the project ID of the tenant to be added and click **OK**.
To add multiple tenants, enter their project IDs and separate them with commas. Click **OK**.

3.5.9 Deleting Image Recipients Who Can Use Shared Images

Scenarios

This section describes how to delete image recipients who can use shared images.

Prerequisites

- You have shared private images.
- You have obtained project IDs of the image recipients.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab.
4. Click the image name to view image details.
5. View the tenants who can use shared image.
6. Determine the operation to perform based on how many tenants you want to delete:
 - To delete an image recipient: locate the target recipient and click **Delete**.
 - To delete all image recipients, click **Delete All** above the image recipient list.
7. Click **Yes**.

3.6 Exporting Images

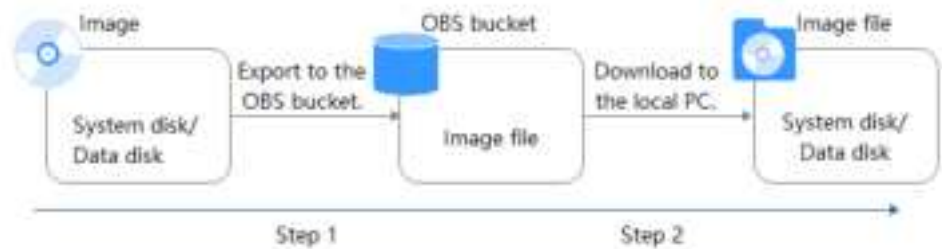
Scenarios

After creating a private image, you can export it to a standard OBS bucket and then download it to your local PC. This section describes how to export an image.

Background

- You can reproduce cloud servers and their running environments in on-premises clusters or private clouds by exporting their images from the cloud platform. The following figure shows the process of exporting an image.

Figure 3-1 Exporting an image



- The time required for exporting an image depends on the image size and the number of concurrent export tasks.
- You can export images in QCOW2, VMDK, VHD, and ZVHD formats. Images exported in different formats may vary in size.

Constraints

- The following private images cannot be exported:
 - Full-ECS image
 - Private image created from a Windows or SUSE public image
- The image size must be less than 128 GB.

Prerequisites

An OBS bucket is available in the region where the private image is located.

If no OBS bucket is available, create one by referring to *Object Storage Service User Guide*. Select **Standard** for **Storage Class**.

Procedure

- Log in to the management console.
- Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
- Locate the row that contains the image to be exported, click **More** in the **Operation** column and select **Export**.
- In the displayed **Export Image** dialog box, set the following parameters:
 - Format**: Select one from **qcow2**, **vmdk**, **vhd**, and **zvhd** as you need.
 - Name**: Enter a name that is easy to identify.
 - Storage Path**: Click  to expand the bucket list and select an OBS bucket for storing the exported image.
- Click **OK**.
You can view the image export progress above the private image list.

Follow-up Procedure

After the image is exported successfully, you can download it from the OBS bucket through the management console or OBS Browser+.

3.7 Optimizing a Windows Private Image

3.7.1 Optimization Process

ECSs require Xen Guest OS driver (PV driver) and KVM Guest OS driver (UVP VMTools) for proper running. To ensure that ECSs support both Xen and KVM and to improve network performance, the PV driver and UVP VMTools must be installed for the image.

1. Create an ECS using the Windows private image to be optimized and log in to the ECS.
2. Install the latest version of PV driver on the ECS.
For details, see [Installing the PV Driver](#).
3. Install the UVP VMTools required for creating ECSs in the KVM virtual resource pool.
For details, see [Installing UVP VMTools](#).
4. On the ECS, choose **Control Panel > Power Options**. Click **Choose when to turn off the display**, select **Never** for **Turn off the display**, and save the changes.
5. Clear system logs and then stop the ECS.
For details, see [Clearing System Logs](#).
6. Create a Windows private image using the ECS.

3.7.2 Viewing the Virtualization Type of a Windows ECS

Open the cmd window and run the following command to query the virtualization type of the ECS:

systeminfo

If the values of **System Manufacturer** and **BIOS Version** are **Xen**, the ECS uses Xen. If KVM is required, perform the operations in this section to optimize a Windows private image.

NOTE

If the ECS uses KVM, you are also advised to optimize the private image to prevent any exceptions with the ECSs created from the image.

Figure 3-2 Viewing the virtualization type of a Windows ECS

```
systeminfo
Host Name: ECS-E5AF
OS Name: Microsoft Windows Server 2012 R2 Datacenter
OS Version: 6.3.9600 x-64 Build 9600
Manufacturer: Microsoft Corporation
Configuration: Standalone Server
Build Type: Multiprocessor Free
Registered Owner: Windows User
Registered Organization:
Product ID: 00253-50800-00000-00442
Original Install Date: 11/2/2015, 21:05:21
System Boot Time: 8/2/2018, 18:31:04
System Manufacturer: Ken
System Model: x64-based PC
System Type: 1 Processor(s) Installed,
Processor(s): 1811: Intel(R) Family 6 Model 62 Stepping 4 GenuineInt
1 - 1000 MHz
BIOS Version: Ken 4.1.2.115-908.762., 3/21/2018
Windows Directory: C:\Windows
System Directory: C:\Windows\System32
Event Service: \Device\Harddisk0\Volume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC+08:00) Beijing, Chongqing, Hong Kong, Urumqi
Total Physical Memory: 1.816 MB
Available Physical Memory: 256 MB
Virtual Memory: Max Size: 1.236 MB
Virtual Memory: Available: 476 MB
Virtual Memory: In Use: 868 MB
```

3.7.3 Obtaining Required Software Packages

PV Driver

Table 3-1 lists the PV driver software packages required for optimizing Windows private images.

Table 3-1 PV driver software packages

Software Package	OS	How to Obtain
pvdriver-win2008R2-64bit.zip	Windows Server 2008 R2 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2008R2-64bit.zip
pvdriver-win2012-64bit.zip	Windows Server 2012 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2012-64bit.zip
pvdriver-win2012R2-64bit.zip	Windows Server 2012 R2 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2012R2-64bit.zip
pvdriver-win2016-64bit.zip	Windows Server 2016 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2016-64bit.zip

UVP VMTools

Table 3-2 lists the UVP VMTools software packages required for optimizing Windows private images.

Table 3-2 UVP VMTools software packages

Software Package	OS	How to Obtain
vmtools-WIN2008-x86.zip	Windows Server 2008-x86	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2008-x86.zip
vmtools-WIN2008-x64.zip	Windows Server 2008-x64	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2008-x64.zip
vmtools-WIN2008R2-x64.zip	Windows Server 2008 R2-x64	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2008R2-x64.zip
vmtools-WIN2012-x64.zip	Windows Server 2012-x64	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2012-x64.zip
vmtools-WIN2012R2-x64.zip	Windows Server 2012 R2-x64	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2012R2-x64.zip
vmtools-WIN2016-x64.zip	- Windows Server 2016-x64 - Windows Server 2019-x64	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/vmtools-WIN2016-x64.zip

3.7.4 Installing the PV Driver

Scenarios

When using an ECS or external image file to create a private image, ensure that the PV driver has been installed in the OS to enable Xen virtualization for subsequently created ECSs, improve the I/O processing performance of the ECSs, and implement advanced functions such as monitoring hardware of the ECSs.

CAUTION

If you do not install the PV driver, the ECS network performance will be poor, and the security group and firewall configured for the ECS will not take effect.

The PV driver has been installed by default when you use a public image to create ECSs. You can perform the following operations to verify the installation:

Open the **version** configuration file to check whether the PV driver is the latest:

C:\Program Files (x86)\Xen PV Drivers\bin\version

Prerequisites

- An OS has been installed for the ECS, and an EIP has been bound to the ECS.
- The remaining capacity of the ECS system disk must be greater than 32 MB.
- If the ECS uses Windows 2008, you must install the PV driver using the administrator account.
- The PV driver software package has been downloaded on the ECS. For how to obtain the software package, see [Obtaining Required Software Packages](#).
- To avoid an installation failure, perform the following operations before starting the installation:
 - Uninstall third-party virtualization platform tools, such as Citrix Xen Tools and VMware Tools. For how to uninstall the tools, see the corresponding official documents of the tools.
 - Disable your antivirus and intrusion detection software. You can enable the software after the PV driver is installed.

Installing the PV Driver

1. Log in to the Windows ECS using VNC.
For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

NOTE

You must log in to the ECS using VNC. Remote desktop connection is not allowed because the NIC driver needs to be updated during the installation but the NIC is in use for the remote desktop connection. As a result, the installation will fail.

2. On the ECS, choose **Start > Control Panel**.
3. Click **Uninstall a program**.
4. Uninstall **GPL PV drivers for Windows x.x.x.xx** as prompted.
5. Download the required PV driver based on the ECS OS and [Obtaining Required Software Packages](#).
6. Decompress the PV driver software package.
7. Right-click **GPL PV Drivers for Windows x.x.x.xx**, select **Run as administrator**, and complete the installation as prompted.
8. Restart the ECS as prompted to make the PV driver take effect.
ECSs running Windows Server 2008 must be restarted twice.

NOTE

After the PV driver is installed, the ECS NIC configuration will be lost. If you have configured NICs before, you need to configure them again.

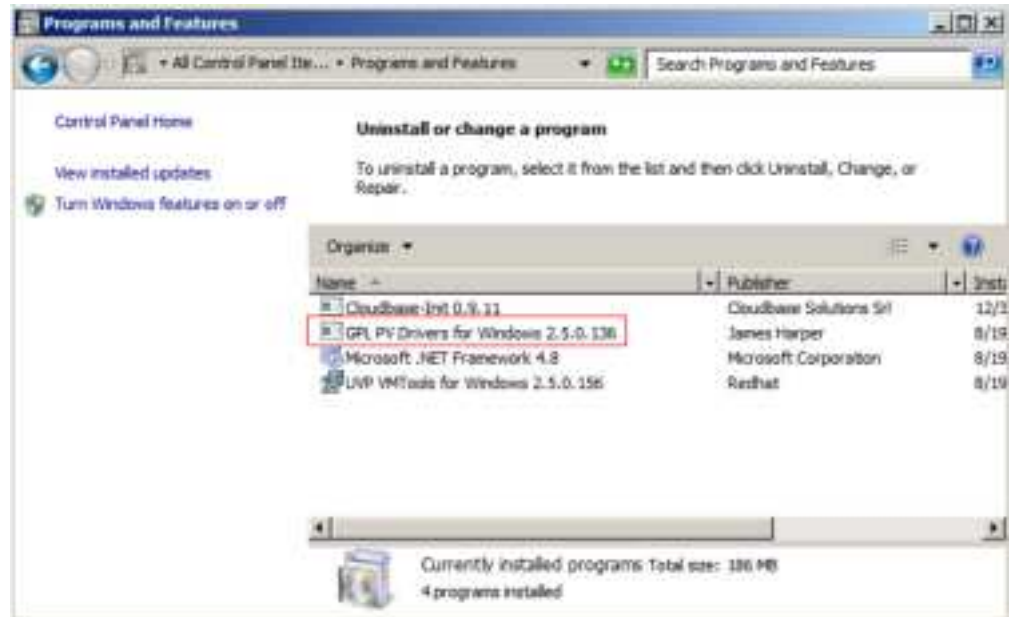
Verifying the Installation

Perform the following steps to verify the installation of the PV driver:

1. Click **Start**. Choose **Control Panel > Programs and Features**.
2. Locate the PV driver for Windows.

If UVP VMTools for Windows exists, the installation is successful, as shown in [Figure 3-3](#).

Figure 3-3 Verifying the installation



3.7.5 Installing UVP VMTools

Scenarios

Before using an ECS or external image file to create a private image, ensure that UVP VMTools has been installed in the OS to enable subsequently created ECSs to support KVM virtualization and improve network performance.

CAUTION

If you do not install UVP VMTools, NICs of the ECS may not be detected and the ECS cannot communicate with other resources.

UVP VMTools has been installed by default when you use a public image to create ECSs. You can perform the following operations to verify the installation:

Open the **version** configuration file to check whether UVP VMTools is the latest:

C:\Program Files (x86)\virtio\bin\version

If the version is 2.5.0 or later, the current UVP VMTools can be used. Otherwise, perform operations in [Installing UVP VMTools](#) to install UVP VMTools.

Prerequisites

- An EIP has been bound to the ECS.
- The UVP VMTools installation package has been downloaded on the ECS. For how to obtain the installation package, see [Obtaining Required Software Packages](#).
- Ensure that the ECS has at least 50 MB disk space.
- To avoid an installation failure, perform the following operations before starting the installation:

- Uninstall third-party virtualization platform tools, such as Citrix Xen Tools and VMware Tools. For how to uninstall the tools, see the corresponding official documents of the tools.
- Disable your antivirus and intrusion detection software. You can enable the software after UVP VMTools is installed.

Installing UVP VMTools

The following operations describe how to install UVP VMTools. **vmtools-WIN2008R2-x64.exe** extracted from **vmtools-WIN2008R2-x64.zip** is used as an example.

1. Log in to the Windows ECS using VNC.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

NOTE

You must log in to the ECS using VNC. Remote desktop connection is not allowed because the NIC driver needs to be updated during the installation but the NIC is in use for the remote desktop connection. As a result, the installation will fail.

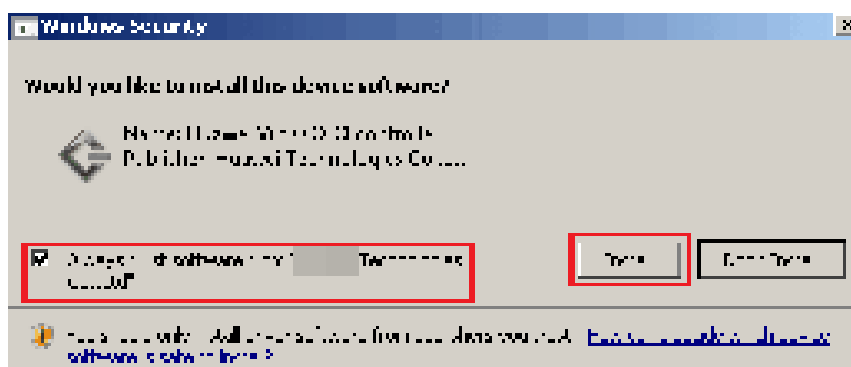
2. Download the required UVP VMTools based on the ECS OS and [Obtaining Required Software Packages](#).
3. Decompress the UVP Tools software package. This section uses **vmtools-WIN2008R2-x64.exe** extracted from **vmtools-WIN2008R2-x64.zip** as an example to describe how to decompress the UVP Tools software package.
4. Right-click **vmtools-WIN2008R2-x64.exe**, select **Run as administrator** from the shortcut menu, and complete the installation as prompted.
5. In the displayed dialog box, select **I accept the terms in the License Agreement** and click **Install**.

Figure 3-4 Installing UVP VMTools



6. Install UVP VMTools as prompted.
7. Perform the following operations to install UVP VMTools on an ECS running Windows Server 2008:
 - a. The **Windows Security** dialog box shown in [Figure 3-5](#) may be displayed during installation. In the dialog box, select **Always trust...** and click **Install**. Otherwise, the installation will fail.

Figure 3-5 Windows Security



- b. Click **Finish**.
8. Perform the operations in [Verifying the Installation](#) to check whether UVP VMTools is successfully installed.

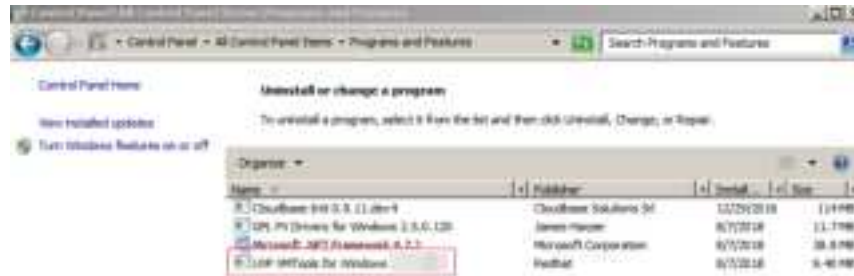
Verifying the Installation

Perform the following steps to verify the installation of UVP VMTools:

1. Click **Start**. Choose **Control Panel > Programs and Features**.
2. Locate UVP VMTools for Windows.

If UVP VMTools for Windows exists, the installation is successful, as shown in [Figure 3-6](#).

Figure 3-6 Verifying the installation



3.7.6 Clearing System Logs

After installing the PV driver and UVP VMTools, perform the following operations to clear system logs:

1. For Windows Server 2008 and Windows Server 2012, right-click **Computer** and select **Manage**.
2. In the displayed dialog box, choose **System Tools > Event Viewer > Windows Logs** and delete logs of five items.
3. Stop the ECS.

3.8 Optimizing a Linux Private Image

3.8.1 Optimization Process

A Linux ECS can be switched from Xen to KVM if xen-pv and VirtIO drivers run on the ECS. Before changing a Xen-based ECS to a KVM-based ECS, ensure that the required drivers have been installed and the UUID has been configured for the Linux private image. In addition, optimizing the private image can improve network performance of the ECS.

1. Use the Linux image to be optimized to create an ECS, and start and log in to the ECS.
2. Uninstall the PV Driver installed on the ECS.
For details, see [Uninstalling the PV Driver from a Linux ECS](#).
3. Change the disk ID in the GRUB configuration file to UUID.
For details, see [Changing the Disk Identifier in the GRUB Configuration File to UUID](#).
4. Change the disk ID in the fstab file to UUID.
For details, see [Changing the Disk Identifier in the fstab File to UUID](#).
5. Install native KVM drivers.
For details, see [Installing Native KVM Drivers](#).

6. Delete log files and historical records, and stop the ECS.
For details, see [Clearing System Logs](#).
7. Create a Linux private image using the ECS.

3.8.2 Viewing the Virtualization Type of a Linux ECS

You can run the following command to query the virtualization type of an ECS:

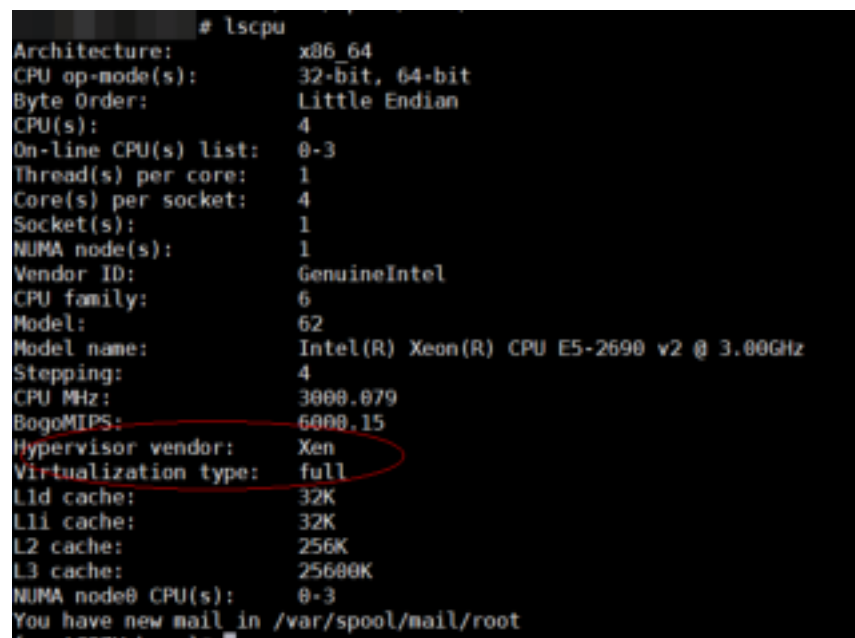
lscpu

If the value of **Hypervisor vendor** is **Xen**, the ECS uses Xen. If KVM is required, perform the operations in this section to optimize the Linux private image.

NOTE

If the ECS uses KVM, you are also advised to optimize the private image to prevent any exceptions with the ECSs created from the image.

Figure 3-7 Viewing the virtualization type of a Linux ECS



```
# lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 62
Model name: Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz
Stepping: 4
CPU MHz: 3000.079
BogoMIPS: 6000.15
Hypervisor vendor: Xen
Virtualization type: full
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 25600K
NUMA node0 CPU(s): 0-3
You have new mail in /var/spool/mail/root
```

3.8.3 Uninstalling the PV Driver from a Linux ECS

Scenarios

When optimizing a Linux private image, you need to change the UUID in the fstab and GRUB configuration files, and install native Xen and KVM drivers on the ECS. To ensure that you can successfully install native Xen and KVM drivers, you must uninstall the PV driver from the ECS.

Procedure

1. Log in to the ECS as user **root** using VNC.
2. Run the following command to check whether the PV driver is installed in the OS:

ps -ef | grep uvp-monitor

The PV driver is installed in the OS if the following information is displayed:

```
root 4561 1 0 Jun29 ? 00:00:00 /usr/bin/uvp-monitor
root 4567 4561 0 Jun29 ? 00:00:00 /usr/bin/uvp-monitor
root 6185 6085 0 03:04 pts/2 00:00:00 grep uvp-monitor
```

- If the PV driver is installed, go to **3**.
 - If the PV driver is not installed, perform the operations in [Changing the Disk Identifier in the fstab File to UUID](#), [Installing Native KVM Drivers](#), and [Changing the Disk Identifier in the GRUB Configuration File to UUID](#).
3. In the VNC login window, open the CLI.
For how to open the CLI, see the OS manual.
 4. Run the following command to uninstall the PV driver:
/etc/.uvp-monitor/uninstall
 - The PV driver is uninstalled successfully if the following command output is displayed:
The PV driver is uninstalled successfully. Reboot the system for the uninstallation to take effect.
 - If **.uvp-monitor** is not contained in the command output, go to **5**.
-bash: /etc/.uvp-monitor/uninstall: No such file or directory
 5. Perform the following operations to delete uvp-monitor that failed to take effect, preventing log overflow:
 - a. Run the following command to check whether UVP user-mode programs are installed in the OS:
rpm -qa | grep uvp
Information similar to the following is displayed:

```
libxenstore_uvp3_0-3.00-36.1.x86_64
uvp-monitor-2.2.0.315-3.1.x86_64
kmod-uvpmod-2.2.0.315-3.1.x86_64
```
 - b. Run the following commands to delete the installation packages:
rpm -e kmod-uvpmod
rpm -e uvp-monitor
rpm -e libxenstore_uvp

3.8.4 Changing the Disk Identifier in the GRUB Configuration File to UUID

Scenarios

When optimizing a Linux private image, you need to change the disk identifier to UUID in the GRUB configuration file of the ECS.

Modify the **menu.lst** or **grub.cfg** configuration file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, **/boot/grub/grub.conf**, or **/boot/efi/EFI/euleros/grub.cfg**), and configure the boot partition using the UUID.

NOTE

The root partition identified in the configuration file varies depending on the OS. It may be **root=/dev/xvda** or **root=/dev/disk**.

Procedure

- Ubuntu 14.04: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub/grub.cfg** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required. The procedure is as follows:
 - a. Log in to the ECS as user **root**.
 - b. Run the following command to query all types of mounted file systems and device UUIDs:
blkid
The following information is displayed:

```
/dev/xvda1: UUID="ec51d860-34bf-4374-ad46-a0c3e337fd34" TYPE="ext3"  
/dev/xvda5: UUID="7a44a9ce-9281-4740-b95f-c8de33ae5c11" TYPE="swap"
```
 - c. Run the following command to query the **grub.cfg** file:
cat /boot/grub/grub.cfg
The following information is displayed:

```
.....menuentry 'Ubuntu Linux, with Linux 3.13.0-24-generic' --class ubuntu --class gnu-linux --  
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.13.0-24-generic-advanced-  
ec51d860-34bf-4374-ad46-a0c3e337fd34' {  
    recordfail  
    load_video  
    gfxmode $linux_gfx_mode  
    insmod gzio  
    insmod part_msdos  
    insmod ext2  
    if [ x$feature_platform_search_hint = xy ]; then  
        search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34  
    else  
        search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34  
    fi  
    echo 'Loading Linux 3.13.0-24-generic ...'  
    linux /boot/vmlinuz-3.13.0-24-generic root=/dev/xvda1 ro  
    echo 'Loading initial ramdisk ...'  
    initrd /boot/initrd.img-3.13.0-24-generic  
}
```
 - d. Check whether the root partition in the **/boot/grub/grub.cfg** configuration file contains **root=/dev/xvda1** or **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34**.
 - If **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34** is contained, the root partition is in the UUID format and requires no change.
 - If **root=/dev/xvda1** is contained, the root partition is in the device name format. Go to [5](#).
 - e. Identify the UUID of the root partition device based on **root=/dev/xvda1** (device name of the root partition) and the partition information obtained by running the **blkid** command.
 - f. Run the following command to open the **grub.cfg** file:
vi /boot/grub/grub.cfg
 - g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda1** to **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34**.

- h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.
- i. Run the following command to verify the change:

cat /boot/grub/grub.cfg

The change is successful if information similar to the following is displayed:

```
.....menuentry 'Ubuntu Linux, with Linux 3.13.0-24-generic' --class ubuntu --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.13.0-24-generic-advanced-
ec51d860-34bf-4374-ad46-a0c3e337fd34' {
  recordfail
  load_video
  gfxmode $linux_gfx_mode
  insmod gzio
  insmod part_msdos
  insmod ext2
  if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
  else
  search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
  fi
  echo 'Loading Linux 3.13.0-24-generic ...'
  linux /boot/vmlinuz-3.13.0-24-generic root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34 ro
  echo 'Loading initial ramdisk ...'
  initrd /boot/initrd.img-3.13.0-24-generic
}
```

- CentOS 6.5: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub/grub.conf** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required. The procedure is as follows:

- a. Log in to the ECS as user **root**.
- b. Run the following command to query all types of mounted file systems and device UUIDs:

blkid

The following information is displayed:

```
/dev/xvda1: UUID="749d6c0c-990a-4661-bed1-46769388365a" TYPE="swap"
/dev/xvda2: UUID="f382872b-eda6-43df-9516-5a687fecdc6" TYPE="ext4"
```

- c. Run the following command to query the **grub.conf** file:

cat /boot/grub/grub.conf

The following information is displayed:

```
default=0
timeout=5
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-573.8.1.el6.x86_64)
root (hd0,1)
kernel /boot/vmlinuz-2.6.32-573.8.1.el6.x86_64 ro root=/dev/xvda2 rd_NO_LUKS rd_NO_LVM
LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=autoKEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img
```

- d. Check whether the root partition in the **/boot/grub/grub.conf** configuration file contains **root=/dev/xvda2** or **root=UUID=f382872b-eda6-43df-9516-5a687fecdc6**.
 - If **root=UUID=f382872b-eda6-43df-9516-5a687fecdc6** is contained, the root partition is in the UUID format and requires no change.

- If **root=/dev/xvda2** is contained, the root partition is in the device name format. Go to 5.
- e. Identify the UUID of the root partition device based on **root=/dev/xvda2** (device name of the root partition) and the partition information obtained by running the **blkid** command.
- f. Run the following command to open the **grub.conf** file:
vi /boot/grub/grub.conf
- g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda2** to **root=UUID=f382872b-eda6-43df-9516-5a687fecdc66**.
- h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.
- i. Run the following command to verify the change:
cat /boot/grub/grub.cfg
The change is successful if information similar to the following is displayed:

```
default=0
timeout=5
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-573.8.1.el6.x86_64)
root (hd0,1)
kernel /boot/vmlinuz-2.6.32-573.8.1.el6.x86_64 ro root=UUID=f382872b-eda6-43df-9516-5a687fecdc66 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD
SYSFONT=latarcyrheb-sun16 crashkernel=autoKEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM
rhgb quiet
initrd /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img
```
- CentOS 7.0: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub2/grub.cfg** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required.
 - a. Log in to the ECS as user **root**.
 - b. Run the following command to query all types of mounted file systems and device UUIDs:
blkid
The following information is displayed:

```
/dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"
/dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
```
 - c. Run the following command to query the **grub.cfg** file:
cat /boot/grub2/grub.cfg
The following information is displayed:

```
.....
menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core)' --class fedora --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-
advanced-4eb40294-4c6f-4384-bbb6-b8795bbb1130' {
  load_video
  set gfxpayload=keep
  insmod gzio
  insmod part_msdos
  insmod xfs
  set root='hd0,msdos2'
  if [ x$feature_platform_search_hint = xy ]; then
```

```
search --no-floppy --fs-uuid --set=root --hint='hd0,msdos2'4eb40294-4c6f-4384-bbb6-  
b8795bbb1130  
else  
search --no-floppy --fs-uuid --set=root 4eb40294-4c6f-4384-bbb6-b8795bbb1130  
fi  
linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=/dev/xvda2 ro crashkernel=auto rhgb quiet  
LANG=en_US.UTF-8  
initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img  
}
```

- d. Check whether the root partition in the **/boot/grub/grub.cfg** configuration file contains **root=/dev/xvda2** or **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130**.
 - If **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130** is contained, the root partition is in the UUID format and requires no change.
 - If **root=/dev/xvda2** is contained, the root partition is in the device name format. Go to [5](#).
- e. Identify the UUID of the root partition device based on **root=/dev/xvda2** (device name of the root partition) and the partition information obtained by running the **blkid** command.
- f. Run the following command to open the **grub.cfg** file:
- g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda2** to **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130**.
- h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.
- i. Run the following command to verify the change:

```
cat /boot/grub2/grub.cfg
```

The change is successful if information similar to the following is displayed:

```
.....  
menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core)' --class fedora --class gnu-linux --  
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-  
advanced-4eb40294-4c6f-4384-bbb6-b8795bbb1130' {  
load_video  
set gfxpayload=keep  
insmod gzio  
insmod part_msdos  
insmod xfs  
set root='hd0,msdos2'  
if [ x$feature_platform_search_hint = xy ]; then  
search --no-floppy --fs-uuid --set=root --hint='hd0,msdos2'4eb40294-4c6f-4384-bbb6-  
b8795bbb1130  
else  
search --no-floppy --fs-uuid --set=root 4eb40294-4c6f-4384-bbb6-b8795bbb1130  
fi  
linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=UUID=4eb40294-4c6f-4384-bbb6-  
b8795bbb1130 ro crashkernel=auto rhgb quiet LANG=en_US.UTF-8  
initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img  
}
```

3.8.5 Changing the Disk Identifier in the fstab File to UUID

Scenarios

When optimizing a Linux private image, you need to change the disk identifier to UUID in the fstab configuration file of the ECS.

Procedure

- Take CentOS 7.0 as an example. Run **blkid** to obtain the UUIDs of all partitions. Modify the **/etc/fstab** file and use the partition UUIDs to configure automatic partition mounting.

- Log in to the ECS as user **root**.
- Run the following command to query all types of mounted file systems and device UUIDs:

blkid

The following information is displayed:

```
/dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"  
/dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
```

- Run the following command to query the **fstab** file:

cat /etc/fstab

The following information is displayed:

```
[root@CTU1000028010 ~]# cat /etc/fstab  
/dev/xvda2 / xfs defaults 0 0  
/dev/xvda1 swap swap defaults 0 0
```

- Check whether the disk identifier in the **fstab** file is the device name.
 - If the disk is represented by UUID, no further operation is required.
 - If the disk is represented by the device name, go to [5](#).
 - Run the following command to open the **fstab** file:
- Take CentOS 7.1 as an example. Run **blkid** to obtain the UUIDs of all partitions. Modify the **/etc/fstab** file and use the partition UUIDs to configure automatic partition mounting.
- Log in to the ECS as user **root**.
 - Run the following command to query all types of mounted file systems and device UUIDs:

blkid

```
/dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"  
/dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
```

Before the change:

```
[root@CTU1000028010 ~]# cat /etc/fstab  
/dev/xvda2 / xfs defaults 0 0  
/dev/xvda1 swap swap defaults 0 0
```

After the change:

```
[root@CTU1000028010 ~]# cat /etc/fstab
UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130 / xfs defaults 0 0
UUID=2de37c6b-2648-43b4-a4f5-40162154e135 swap swap defaults 0 0
```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.
4. Run the following command to verify the change:

```
cat /etc/fstab
```

The change is successful if information similar to the following is displayed:

```
[root@CTU1000028010 ~]# cat /etc/fstab
UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130 / xfs defaults 0 0
UUID=2de37c6b-2648-43b4-a4f5-40162154e135 swap swap defaults 0 0
```

3.8.6 Installing Native KVM Drivers

Scenarios

When optimizing a Linux private image, you need to install native KVM drivers on the ECS.

CAUTION

If you do not install KVM drivers, NICs of the ECS may not be detected and the ECS cannot communicate with other resources. Therefore, you must install KVM drivers.

This section describes how to install native KVM drivers.

Prerequisites

- ECSs that use native Linux KVM drivers must have a kernel later than the 2.6.24 version.
- Disable your antivirus and intrusion detection software. You can enable the software after KVM drivers are installed.

Procedure

Edit the configuration file based on the OS version.

- CentOS, EulerOS

Take CentOS 7.0 as an example. In the **/etc/dracut.conf** file, add VirtIO drivers to **add_drivers**, including **virtio_blk**, **virtio_scsi**, **virtio_net**, **virtio_pci**, **virtio_ring**, and **virtio**. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate **initrd**.

For details, see [CentOS and EulerOS](#).

- Ubuntu and Debian

In the **/etc/initramfs-tools/modules** file, add VirtIO drivers, including **virtio_blk**, **virtio_scsi**, **virtio_net**, **virtio_pci**, **virtio_ring**, and **virtio**. Separate driver names with spaces. Save and exit the **/etc/initramfs-tools/modules** file. Run the **update-initramfs -u** command to regenerate **initrd**.

For details, see [Ubuntu and Debian](#).

- SUSE and openSUSE
 - If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, add VirtIO drivers to **INITRD_MODULES=""** in the **/etc/sysconfig/kernel** file. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **mkinitrd** command to regenerate initrd.
 - If the OS version is SUSE 12 SP1, add VirtIO drivers to **add_drivers** in the **/etc/dracut.conf** file. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **dracut -f** command to regenerate initrd.
 - If the OS version is later than SUSE 12 SP1 or openSUSE 13, add VirtIO drivers to **add_drivers** in the **/etc/dracut.conf** file. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

For details, see [SUSE and openSUSE](#).

CentOS and EulerOS

1. Run the following command to open the **/etc/dracut.conf** file:
vi /etc/dracut.conf
2. Press **i** to enter the editing mode and add VirtIO drivers to **add_drivers** (the format depends on the OS requirements).

```
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
....
```
3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
4. Run the following command to regenerate initrd:

dracut -f /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img

If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

5. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is initrd, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initrd-`uname -r` | grep virtio

Assume that the virtual file system is initramfs. The following command output will be displayed:

```
[root@CTU10000xxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
-rwxr--r-- 1 root root 23448 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
block/virtio_blk.ko
-rwxr--r-- 1 root root 50704 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/
drivers/net/virtio_net.ko
-rwxr--r-- 1 root root 28424 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
scsi/virtio_scsi.ko
```

```
drwxr-xr-x  2 root   root       0 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio
-rwxr--r--  1 root   root    14544 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio.ko
-rwxr--r--  1 root   root    21040 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_pci.ko
-rwxr--r--  1 root   root    18016 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_ring.ko
```

 NOTE

If you add built-in drivers to the `initrd` or `initramfs` file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the `lsinitrd` command. You can run the following command to check whether the drivers are built-in ones in the kernel:

```
cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
```

Ubuntu and Debian

1. Run the following command to open the **modules** file:
vi /etc/initramfs-tools/modules
2. Press **i** to enter the editing mode and add VirtIO drivers to the **/etc/initramfs-tools/modules** file (the format depends on the OS requirements).

```
[root@CTU10000xxxxx ~]#vi /etc/initramfs-tools/modules
...
# Examples:
#
# raid1
# sd_mOd
virtio_blk
virtio_scsi
virtio_net
virtio_pci
virtio_ring
virtio
```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/initramfs-tools/modules** file.
4. Run the following command to regenerate `initrd`:
update-initramfs -u
5. Run the following command to check whether native KVM drivers have been installed:

```
lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
```

```
[root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
lib/modules/3.5.0-23-generic/kernel/drivers/scsi/virtio_scsi.ko
```

 NOTE

If you add built-in drivers to the `initrd` or `initramfs` file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the `lsinitrd` command. You can run the following command to check whether the drivers are built-in ones in the kernel:

```
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
CONFIG_VIRTIO_BLK=y
CONFIG_VIRTIO_NET=y
CONFIG_VIRTIO=y
CONFIG_VIRTIO_RING=y
CONFIG_VIRTIO_PCI=y
CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
```

SUSE and openSUSE

If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the `/etc/sysconfig/kernel` file. For details, see [scenario 1](#).

If the OS version is SUSE 12 SP1, modify the `/etc/dracut.conf` file and add VirtIO drivers. For details, see [scenario 2](#).

If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the `/etc/dracut.conf` file and add VirtIO drivers to `add_drivers`. For details, see [scenario 3](#).

- If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, perform the following steps:
 - a. Run the following command to modify the `/etc/sysconfig/kernel` file:
vi etc/sysconfig/kernel
 - b. Add VirtIO drivers to `INITRD_MODULES=""` (the format of drivers depends on the OS).

```
SIA10000xxxxx:~ # vi /etc/sysconfig/kernel
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci
virtio_ring virtio"
```
 - c. Run the **mkinitrd** command to regenerate initrd:

NOTE

If the virtual file system is not the default `initramfs` or `initrd`, run the **dracut -f *Name of the initramfs or initrd file actually used*** command. The actual `initramfs` or `initrd` file name can be obtained from the `menu.lst` or `grub.cfg` file (`/boot/grub/menu.lst`, `/boot/grub/grub.cfg`, or `/boot/grub2/grub.cfg`).

The following is an example `initrd` file of SUSE 11 SP4:

```
default 0
timeout 10
gfxmenu (hd0,0)/boot/message
title sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent console=ttyS0,115200n8 console=tty0
net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
initrd /boot/initrd.vmx
title Failsafe_sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent ide=nodma apm=off noresume edd=off
powersaved=off nohz=off highres=off processor.max+cstate=1 nomodeset x11failsafe
console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1
showopts
initrd /boot/initrd.vmx
```

`/boot/initrd.vmx` in the `initrd` line is the `initrd` file actually used. Run the **dracut -f /boot/initrd.vmx** command. If the `initrd` file does not contain the `/boot` directory, such as `/initramfs-xxx`, run the **dracut -f /boot/initramfs-xxx** command.

- d. Run the following command to check whether the VirtIO module for KVM is loaded:

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
```

```
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

- e. Restart the ECS.
- f. Run the following command to check whether KVM drivers exist in initrd:

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

- If the OS version is SUSE 12 SP1, perform the following steps:
 - a. Run the following command to open the **/etc/dracut.conf** file:

```
vi /etc/dracut.conf
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring
virtio"
```

- c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
- d. Run the following command to regenerate initrd:

dracut -f /boot/initramfs-File name

If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

- e. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is initrd, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initrd-`uname -r` | grep virtio

- If the OS version is later than SUSE 12 SP1 or openSUSE 13, perform the following steps:

Take SUSE Linux Enterprise Server 12 SP2 (x86_64) as an example.

- a. Run the following command to open the **/etc/dracut.conf** file:

vi /etc/dracut.conf

- b. Press **i** to enter the editing mode and add VirtIO drivers to **add_drivers** (the format depends on the OS requirements).

```
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring
virtio"
```

- c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
- d. Run the following command to regenerate initrd:

dracut -f /boot/initramfs-File name

If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

- e. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is initrd, run the following command to check whether native KVM drivers have been loaded:

lsinitrd /boot/initrd-`uname -r` | grep virtio

Assume that the virtual file system is initrd. The following command output will be displayed:

```
sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rw-r--r-- 1 root root 29335 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/
virtio_blk.ko
-rw-r--r-- 1 root root 57007 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/
virtio_net.ko
-rw-r--r-- 1 root root 32415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/scsi/
virtio_scsi.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/kernel/drivers/virtio
-rw-r--r-- 1 root root 19623 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio.ko
-rw-r--r-- 1 root root 38943 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_pci.ko
-rw-r--r-- 1 root root 24431 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_ring.ko
```

 NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

3.8.7 Clearing System Logs

Delete log files and historical records, and stop the ECS.

1. Run the following commands to delete redundant key files:

```
echo > /$path/$to/$root/.ssh/authorized_keys
```

An example command is **echo > /root/.ssh/authorized_keys**.

```
echo > /$path/$to/$none-root/.ssh/authorized_keys
```

An example command is **echo > /home/linux/.ssh/authorized_keys**.

2. Run the following command to clear log files in the **/var/log** directory:

```
rm -rf /var/log/*
```

3. Run the following commands to delete historical records:

```
history -w
```

```
echo > /root/.bash_history
```

```
history -c
```

3.9 Encrypting Images

3.9.1 Overview

IMS allows you to create encrypted images to ensure data security.

NOTE

To use the image encryption function, you must apply for KMS Administrator permissions.

Constraints

- KMS must be enabled.
- Encrypted images cannot be shared with others.
- The system disk of an ECS created from an encrypted image is also encrypted, and its key is the same as the image key.
- If an ECS has an encrypted system disk, private images created using the ECS are also encrypted.
- The key used for encrypting an image cannot be changed.
- If the key used for encrypting an image is disabled or deleted, the image is unavailable.
- Images of multi-project users cannot be encrypted.

3.9.2 Creating Encrypted Images

You can create an encrypted image using an encrypted ECS or an external image file.

- Create an encrypted image using an encrypted ECS.

When you use an ECS to create a private image, if the system disk of the ECS is encrypted, the private image created using the ECS is also encrypted. The

key used for encrypting the image must be the same as that used for encrypting the system disk. For details, see [Creating a System Disk Image from a Windows ECS](#) and [Creating a System Disk Image from a Linux ECS](#).

- Create an encrypted image using an external image file.

When you use an external image file that has been uploaded to an OBS bucket to create a private image, you can select KMS encryption and the key when registering the image to encrypt the image. For details, see [Creating a Windows System Disk Image from an External Image File](#) and [Creating a Linux System Disk Image from an External Image File](#).

3.10 Replicating Images

Scenarios

You can convert encrypted and unencrypted images into each other or enable some advanced features (such as fast ECS creation from an image) using the image replication function. You may need to replicate an image in the following scenarios:

- Replicate an encrypted image to an unencrypted one.
Encrypted images cannot be shared. If you want to share an encrypted image, you can replicate it to an unencrypted one.
- Replicate an encrypted image to an encrypted one.
Keys for encrypting the images cannot be changed. If you want to change the key of an encrypted image, you can replicate this image to a new one and encrypt the new image using an encryption key.
- Replicate an unencrypted image to an encrypted one.
If you want to store an unencrypted image in an encrypted way, you can replicate this image as a new one and encrypt the new image using a key.
- Optimize a system disk image so that it can be used to quickly create ECSs.
If a system disk image supports fast ECS creation, the time required for creating ECSs from it can be greatly reduced. Existing system disk images may not support this function. You can optimize the images using the image replication function. If image A cannot be used to quickly create ECSs, you can replicate it to generate image copy_A, which can be used to quickly create ECSs.

Constraints

Full-ECS images cannot be replicated.

Prerequisites

The images to be replicated are in the **Normal** state.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

The IMS console is displayed.

3. Locate the row that contains the image to be replicated, click **More** in the **Operation** column, and select **Replicate**.
4. In the displayed **Replicate Image** dialog box, set the following parameters:
 - **Name**: Enter a name that is easy to identify.
 - **Description**: This parameter is optional. Enter description of the replication.
 - **Encryption**: If you want to encrypt the image or change a key, select **KMS encryption** and select the key you want to use from the drop-down list.
5. Click **OK**.

On the **Private Images** page, view the replication progress. If the status of the new image becomes **Normal**, the image replication is successful.

3.11 Tagging an Image

Scenarios

You can use tags to classify images. You can add, modify, or delete image tags, or search for required images by tag in the image list.

NOTE

When adding predefined tags to an image or searching for an image using predefined tags, you must have permission to access the Tag Management Service (TMS).

Constraints

An image can have a maximum of 10 tags.

Add, Delete, and Modify Image Tags

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

The IMS console is displayed.
3. Click the **Private Images** tab and click the image name to display the image details.
 - To modify an image tag, go to [4](#).
 - To delete an image tag, go to [5](#).
 - To add an image tag, go to [6](#).
4. Click the **Tags** tab, locate the target tag, and click **Edit** in the **Operation** column. In the displayed dialog box, modify the tag.
5. Click the **Tags** tab, locate the target tag, and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.
6. Click the **Tags** tab and then **Add Tag**. In the displayed dialog box, add a tag.

Search for Private Images by Tag

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab and then **Search by Tag**.
4. Enter the tag key and value.
Neither the tag key nor tag value can be empty. When the tag key and tag value are matched, the system automatically shows your desired private images.
5. Click  to add a tag.
You can add multiple tags to search for private images. The system will display private images that match all tags.
6. Click **Search**.
The system searches for private images based on tag keys or tag values.

3.12 Exporting Image Information

Scenarios

You can export information about public images or your private images in the form of a CSV file. This file contains detailed information about each image, such as the image name, OS, image type, image creation time, and disk capacity.

Exporting Private Image Information

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab and click .
The system will automatically export all private images in the current region under your account to a local directory.

NOTE

The file name is in the format of **private-images-Region ID-Export time**.

Exporting Public Image Information

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Public Images** tab and click .
The system will automatically export all public images in the current region to a local directory.

 NOTE

The file name is in the format of **public-images-Region ID-Export time**.

3.13 Auditing Key Operations

3.13.1 IMS Operations Recorded by CTS

Scenarios

Cloud Trace Service (CTS) is a log audit service provided by the public cloud and intended for cloud security. It allows you to collect, store, and query cloud resource operation records and use these records for security analysis, compliance auditing, resource tracking, and fault locating.

You can use CTS to record IMS operations for later querying, auditing, and backtracking.

Prerequisites

You need to enable CTS before using it. If it is not enabled, IMS operations cannot be recorded. After being enabled, CTS automatically creates a tracker to record all your operations. The tracker stores only the operations of the last seven days. To store the operations for a longer time, store trace files in OBS buckets.

IMS Operations Recorded by CTS

Table 3-3 IMS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating an Image	ims	createImage
Modifying an image	ims	updateImage
Deleting images in a batch	ims	deleteImage
Replicating an image	ims	copyImage
Exporting an image	ims	exportImage
Adding a tenant that can use a shared image	ims	addMember
Modifying tenants that can use a shared image	ims	updateMember
Deleting tenants from the group where the members can use a shared image	ims	deleteMemeber

Table 3-4 Relationship between IMS operations and native OpenStack APIs

Operation	Trace Name	Service Type	Resource Type	OpenStack Component
Creating an Image	createImage	IMS	image	glance
Modifying/ Uploading an image	updateImage	IMS	image	glance
Deleting an image	deleteImage	IMS	image	glance
Tagging an image	addTag	IMS	image	glance
Deleting an image tag	deleteTag	IMS	image	glance
Adding a tenant that can use a shared image	addMember	IMS	image	glance
Modifying information about a tenant that can use a shared image	updateMember	IMS	image	glance
Deleting a tenant from the group where the members can use a shared image	deleteMember	IMS	image	glance

3.13.2 Viewing Traces

Scenarios

Once CTS is enabled, it starts recording IMS operations. You can view operations recorded in the last seven days on the CTS management console.

This section describes how to view the records.

Procedure

1. Log in to the management console.

2. Click **Service List**. Under **Management & Deployment**, select **Cloud Trace Service**.
3. In the navigation pane on the left, choose **Trace List**.
4. Set the filter criteria and click **Query**.

The following filters are available:

- **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**.

Select **Management** for **Trace Type** and **IMS** for **Trace Source**.

Note that:

- If you select **Resource ID** for **Search By**, you need to enter a resource ID. Only whole word match is supported.
 - If you select **Resource name** for **Search By**, you need to select or enter a specific resource name.
 - **Operator**: Select a specific operator from the drop-down list.
 - **Trace Status**: Available values are **All trace statuses**, **Normal**, **Warning**, and **Incident**.
 - Time range: You can select **Last 1 hour**, **Last 1 day**, **Last 1 week**, or **Customize**.
5. Locate the target trace and click  to expand the trace details.
 6. Click **View Trace** in the upper right corner of the trace details area.

3.14 Converting the Image Format Using qemu-img

Scenarios

You can import an image file in VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format to the cloud platform. Image files in other formats need to be converted before being imported. The open-source tool **qemu-img** is provided for you to convert image file formats.

Background

- **qemu-img** supports the mutual conversion of image formats VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED.
- ZVHD and ZVHD2 are self-developed image file formats and cannot be identified by **qemu-img**. To convert image files to any of the two formats, use the **qemu-img-hw** tool. For details, see [Converting the Image Format Using qemu-img-hw](#)
- When you run the command to convert the format of VHD image files, use VPC to replace VHD. Otherwise, qemu-img cannot identify the image format.
For example, to convert a CentOS 6.9 VHD image file into a QCOW2 image file, run the following command:

```
qemu-img convert -p -f vpc -O qcow2 centos6.9.vhd centos6.9.qcow2
```

Windows

1. Install qemu-img.
 - a. Download the qemu-img installation package from <https://qemu.weilnetz.de/w64/>.
 - b. Double-click the setup file to install qemu-img in **D:\Program Files\qemu** (an example installation path).
2. Configure environment variables.
 - a. Choose **Start > Computer** and right-click **Properties**.
 - b. Click **Advanced system settings**.
 - c. In the **System Properties** dialog box, click **Advanced > Environment Variables**.
 - d. In the **Environment Variables** dialog box, search for **Path** in the **System Variable** area and click **Edit**. Add **D:\Program Files\qemu** to **Variable Value**. Use semicolons (;) to separate variable values.

NOTE

If **Path** does not exist, add it and set its value to **D:\Program Files\qemu**.

- e. Click **OK**.
3. Verify the installation.

Choose **Start > Run**, enter **cmd**, and press **Enter**. In the **cmd** window, enter **qemu-img --help**. If the qemu-img version information is contained in the command output, the installation is successful.
4. Convert the image format.
 - a. In the **cmd** window, run the following commands to switch to **D:\Program Files\qemu**:
d:
cd D:\Program Files\qemu
 - b. Run the following command to convert the image file format from VMDK to QCOW2:
qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2
The parameters are described as follows:
 - **-p** indicates the image conversion progress.
 - **-f** indicates the source image format.
 - The part following **-O** (which must be in upper case) consists of the required format, source image file, and target image file.After the conversion is complete, the target image file is displayed in the directory where the source image file is located.
The following information is displayed:

```
# qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2
(100.00/100%)
```
 - c. Run the following command to query details about the converted image file in QCOW2 format:

qemu-img info centos6.9.qcow2

The following information is displayed:

```
# qemu-img info centos6.9.qcow2
image: centos6.9.qcow2
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 200K
cluster_size: 65536
Format specific information:
  compat: 1.1
  lazy refcounts: false
```

Linux

1. Install qemu-img.
 - For Ubuntu or Debian, run the following command:
apt install qemu-img
 - For CentOS, Red Hat, or Oracle, run the following command:
yum install qemu-img
 - For SUSE or openSUSE, run the following command:
zypper install qemu-img
2. Run the following command to check whether the installation is successful:
qemu-img -v

If the version information and help manual of the qemu-img tool are contained in the command output, the installation is successful. If CentOS 7 is used, the command output is as follows:

```
[root@CentOS7 ~]# qemu-img -v
qemu-img version 1.5.3, Copyright (c) 2004-2008 Fabrice Bellard
usage: qemu-img command [command options]
QEMU disk image utility

Command syntax:
check [-q] [-f fmt] [--output=ofmt] [-r [leaks | all]] [-T src_cache] filename
create [-q] [-f fmt] [-o options] filename [size]
commit [-q] [-f fmt] [-t cache] filename
compare [-f fmt] [-F fmt] [-T src_cach]
```

3. Convert the image format. For example, perform the following steps to convert a VMDK image file running CentOS 7 to a QCOW2 image file:
 - a. Run the following command to convert the image file format to QCOW2:
qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2

The parameters are described as follows:

- **-p**: indicates the conversion progress.
- **-f** indicates the source image format.
- The pat following **-O** (which must be in upper case) is the converted image format + source image file name + target image file name.

After the conversion is complete, the target image file is displayed in the directory where the source image file is located.

The following information is displayed:

```
[root@CentOS7 home]# qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk
centos6.9.qcow2
(100.00/100%)
```

- b. Run the following command to query details about the converted image file in QCOW2 format:

qemu-img info centos6.9.qcow2

The following information is displayed:

```
[root@CentOS7 home]# qemu-img info centos6.9.qcow2
image: centos6.9.qcow2
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 200K
cluster_size: 65536
Format specific information:
  compat: 1.1
  lazy refcounts: false
```

Examples

A pre-allocated image depends on two files: **xxxx.vmdk** (configuration file) and **xxxx-flat.vmdk** (data file) and cannot be directly imported to the cloud platform. When you export a pre-allocated image file in VMDK monolithic Flat format from the VMware platform, you must convert its format to common VMDK or QCOW2 before it can be imported to the cloud platform.

The following uses the image files **centos6.9-64bit-flat.vmdk** and **centos6.9-64bit.vmdk** as an example to describe how to use **qemu-img** to convert image formats.

1. Run the following commands to query the image file details:

ls -lh centos6.9-64bit*

qemu-img info centos6.9-64bit.vmdk

qemu-img info centos6.9-64bit-flat.vmdk

The following information is displayed:

```
[root@CentOS7 tmp]# ls -lh centos6.9-64bit*
-rw-r--r--. 1 root root 10G Jun 13 05:30 centos6.9-64bit-flat.vmdk
-rw-r--r--. 1 root root 327 Jun 13 05:30 centos6.9-64bit.vmdk
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit.vmdk
image: centos6.9-64bit.vmdk
file format: vmdk
virtual size: 10G (10737418240 bytes)
disk size: 4.0K
Format specific information:
  cid: 3302005459
  parent cid: 4294967295
  create type: monolithicFlat
  extents:
    [0]:
      virtual size: 10737418240
      filename: centos6.9-64bit-flat.vmdk
      format: FLAT
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit-flat.vmdk
image: centos6.9-64bit-flat.vmdk
file format: raw
virtual size: 10G (10737418240 bytes)
disk size: 0
```

 NOTE

The command output shows that the format of **centos6.9-64bit.vmdk** is VMDK and that of **centos6.9-64bit-flat.vmdk** is RAW. You can convert the format of only **centos6.9-64bit.vmdk**. For details about how to convert it, see [3](#).

2. Run the following command to query the configuration of the pre-allocated image file:

cat centos6.9-64bit.vmdk

The following information is displayed:

```
[root@CentOS7 tmp]# cat centos6.9-64bit.vmdk
# Disk DescriptorFile
version=1
CID=c4d09ad3
parentCID=ffffffff
createType="monolithicFlat"

# Extent description
RW 20971520 FLAT "centos6.9-64bit-flat.vmdk" 0

# The Disk Data Base
#DDB

ddb.virtualHWVersion = "4"
ddb.geometry.cylinders = "20805"
ddb.geometry.heads = "16"
ddb.geometry.sectors = "63"
ddb.adapterType = "ide"
```

3. Place **centos6.9-64bit-flat.vmdk** and **centos6.9-64bit.vmdk** in the same directory. Run the following command to convert the format of **centos6.9-64bit.vmdk** to QCOW2 using **qemu-img**:

```
[root@CentOS7 tmp]# qemu-img convert -p -f vmdk -O qcow2 centos6.9-64bit.vmdk
centos6.9-64bit.qcow2
(100.00/100%)
```

4. Run the following command to query details about the converted image file in QCOW2 format:

qemu-img info centos6.9-64bit.qcow2

The following information is displayed:

```
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit.qcow2
image: centos6.9-64bit.qcow2
file format: qcow2
virtual size: 10G (10737418240 bytes)
disk size: 200K
cluster_size: 65536
Format specific information:
  compat: 1.1
  lazy refcounts: false
```

3.15 Converting the Image Format Using **qemu-img-hw**

Scenarios

You can import an image file in VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format to the cloud platform. Image files in other formats need to be converted into any of these formats using the open-source tool **qemu-img** before being imported. However, the **qemu-img** tool cannot convert image files to the ZVHD or ZVHD2 format. To convert image files to any of the two

formats, use the self-developed tool **qemu-img-hw**. This section describes how to use **qemu-img-hw** to convert an image file to ZVHD2.

Background

qemu-img-hw can be used only in Linux. You can run it on a local Linux server or a Linux ECS on the cloud platform. The following procedure uses an EulerOS ECS as an example.

Procedure

1. Upload the image file to be converted to the ECS.
 - If the local host runs a Linux OS, run the **scp** command.
For example, to upload **image01.qcow2** to the **/usr/** directory on the ECS, run the following command:
scp /var/image01.qcow2 root@xxx.xxx.xx.xxx:/usr/
xxx.xxx.xx.xxx indicates the EIP bound to the ECS.
 - If the local host runs a Windows OS, use a file transfer tool, such as WinSCP, to upload the image file to the ECS.
2. Obtain the **qemu-img-hw** software package, upload it to the ECS, and then decompress the package.

Table 3-5 qemu-img-hw package

Tool Package	How to Obtain
quick-import-tools	https://oss.prod-cloud-ocb.orange-business.com/download/ims/OCB-image-convert-tools.zip

NOTE

This tool can be used only on x86 servers.

3. Convert the image format.
 - a. Go to the directory where **qemu-img-hw** is stored, for example, **/usr/quick-import-tools/qemu-img-hw**.
cd /usr/quick-import-tools/qemu-img-hw
 - b. Run the following command to change file permissions:
chmod +x qemu-img-hw
 - c. Run the **qemu-img-hw** command to convert the image file to the ZVHD2 format.
The command format of **qemu-img-hw** is as follows:
./qemu-img-hw convert -p -O Target_image_format Source_image_file Target_image_file
For example, run the following command to convert an **image01.qcow2** file to an **image01.zvhd2** file:
./qemu-img-hw convert -p -O zvhd2 image01.qcow2 image01.zvhd2

Appendix 1: Common qemu-img-hw Commands

- Converting image file formats: **qemu-img-hw convert -p -O *Target_image_format* *Source_image_file* *Target_image_file***

The parameters are described as follows:

-p: indicates the conversion progress.

The part following **-O** (which must be in upper case) consists of the target image format, source image file, and target image file.

For example, run the following command to convert a QCOW2 image file to a ZVHD2 file:

qemu-img-hw convert -p -O zvhd2 test.qcow2 test.zvhd2

- Querying image file information: **qemu-img-hw info *Image file***
An example command is **qemu-img-hw info test.zvhd2**.
- Viewing help information: **qemu-img-hw -help**

Appendix 2: Common Errors During qemu-img-hw Running

- Symptom:

The following information is displayed when you run the **qemu-img-hw** command:

```
./qemu-img-hw: /lib64/libc.so.6: version `GLIBC_2.14' not found (required by ./qemu-img-hw)
```

Solution:

Run the **strings /lib64/libc.so.6 | grep glibc** command to check the glibc version. If the version is too early, install the latest version. Run the following commands in sequence:

```
wget http://ftp.gnu.org/gnu/glibc/glibc-2.15.tar.gz
```

```
wget http://ftp.gnu.org/gnu/glibc/glibc-ports-2.15.tar.gz
```

```
tar -xvf glibc-2.15.tar.gz
```

```
tar -xvf glibc-ports-2.15.tar.gz
```

```
mv glibc-ports-2.15 glibc-2.15/ports
```

```
mkdir glibc-build-2.15
```

```
cd glibc-build-2.15
```

```
../glibc-2.15/configure --prefix=/usr --disable-profile --enable-add-ons --with-headers=/usr/include --with-binutils=/usr/bin
```

NOTE

If **configure: error: no acceptable C compiler found in \$PATH** is displayed, run the **yum -y install gcc** command.

```
make
```

```
make install
```

- Symptom:

The following information is displayed when you run the **qemu-img-hw** command:

```
./qemu-img-hw: error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory
```

Solution: Run the **yum install libaio** command.

4 Windows Operations

4.1 Setting the NIC to DHCP

Scenarios

If a private image is created from an ECS or external image file and the VM where the ECS or external image file is located is configured with a static IP address, you need to change the NIC attribute to DHCP so that the new ECSs created from the private image can dynamically obtain an IP address.

This section uses Windows Server 2008 R2 as an example to describe how to configure DHCP. For details about how to configure DHCP on ECSs running other OSs, see the relevant OS documentation.

NOTE

When registering an external image file as a private image, configure DHCP on the VM where the external image file is located. You are advised to configure DHCP on the VM and then export the image file.

Prerequisites

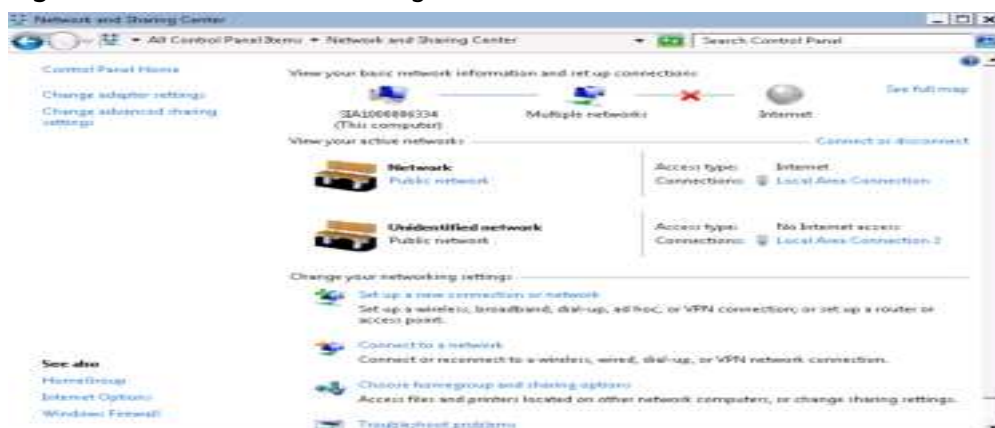
You have logged in to the ECS used to create a Windows private image.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Procedure

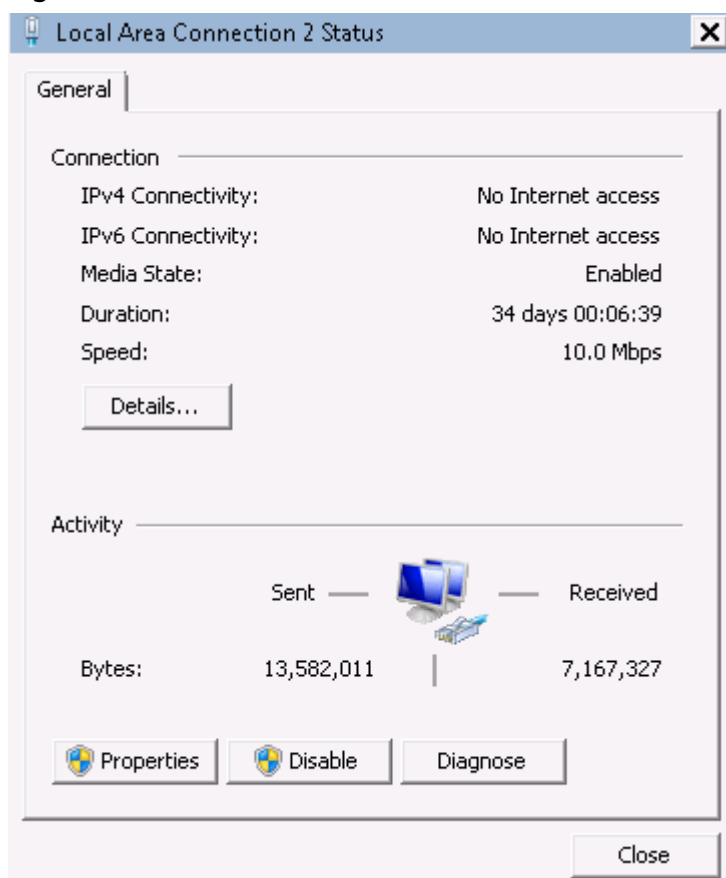
1. On the ECS, choose **Start > Control Panel**.
2. Click **Network and Internet Connections**.
3. Click **Network and Sharing Center**.

Figure 4-1 Network and Sharing Center



4. Select the connection configured with the static IP address. For example, click **Local Area Connection 2**.

Figure 4-2 Local Area Connection 2 Status

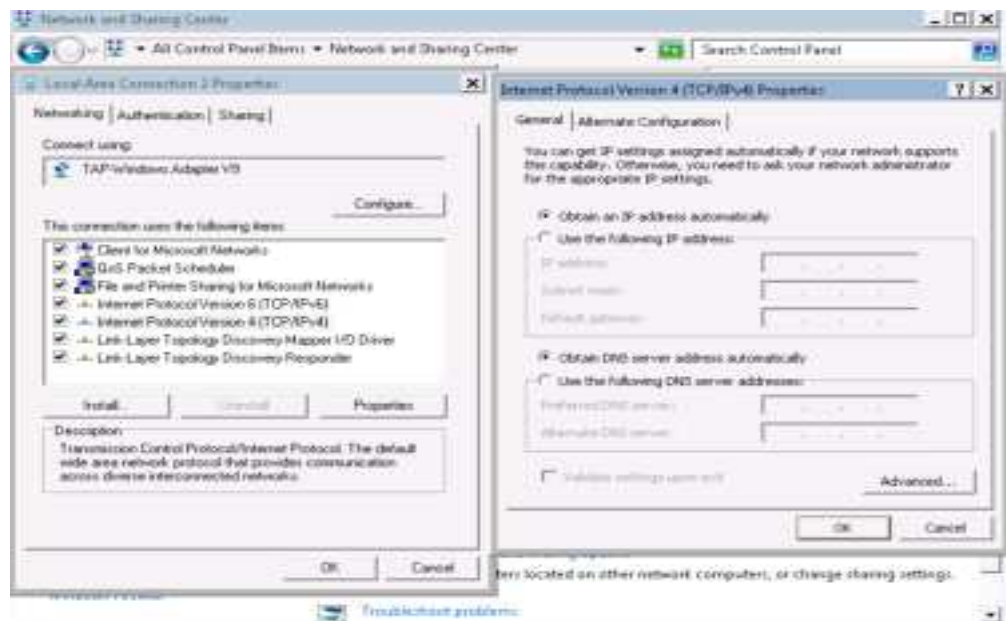


5. Click **Properties** and select the configured Internet protocol version.
6. On the **General** tab, select **Obtain an IP address automatically** and **Obtain DNS server address automatically** and click **OK**. **Figure 4-3** shows the dialog box for configuring the IP address obtaining mode.

NOTE

You are advised to record the original network information so that you can restore the network if necessary.

Figure 4-3 Configuring the IP address obtaining mode



The system will automatically obtain an IP address.

4.2 Enabling Remote Desktop Connection

Scenarios

If you want to remotely access an ECS, enable remote desktop connection for the source ECS when creating a private image. This function must be enabled for GPU-accelerated ECSs.

NOTE

When registering an external image file as a private image, enable remote desktop connection on the VM where the external image file is located. You are advised to enable this function on the VM and then export the image file.

Prerequisites

You have logged in to the ECS used to create a Windows private image.

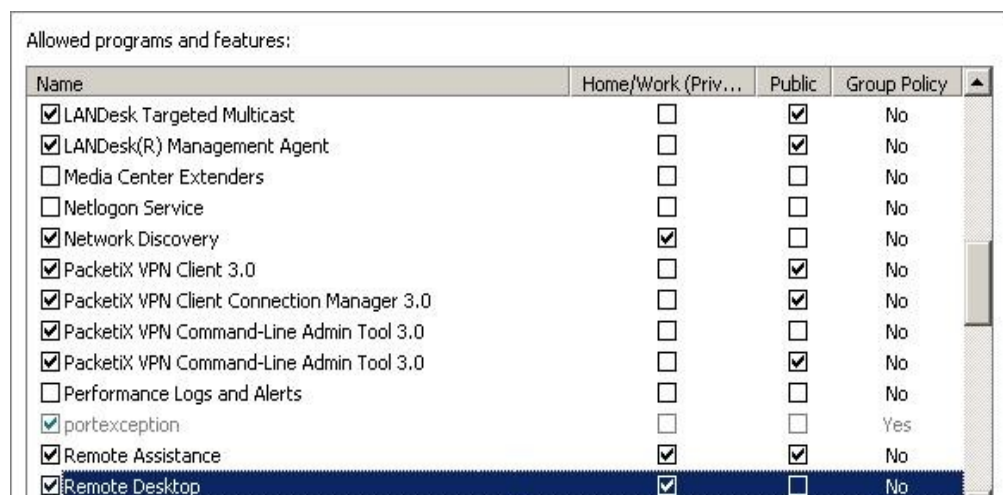
For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Procedure

- Before enabling this function, you are advised to set the resolution of the ECS to 1920×1080.
On the ECS, choose **Start** > **Control Panel**. Under **Appearance and Personalization**, click **Adjust screen resolution**. Then select a proper value from the **Resolution** drop-down list box.
- Choose **Start**, right-click **Computer**, and choose **Properties** from the shortcut menu.

3. Click **Remote settings**.
4. In the **Remote** tab, select **Allow connections from computers running any version of Remote Desktop (less secure)**.
5. Click **OK**.
6. Choose **Start > Control Panel** and navigate to **Windows Firewall**.
7. Choose **Allow a program or feature through Windows Firewall** in the left pane.
8. Select programs and features that are allowed by the Windows firewall for **Remote Desktop** based on your network requirements and click **OK** in the lower part.

Figure 4-4 Configuring remote desktop



4.3 Installing and Configuring Cloudbase-Init

Scenarios

To ensure that you can use the user data injection function to inject initial custom information into ECSs created from a private image (such as setting the ECS login password), install Cloudbase-Init on the ECS used to create the image.

- If Cloudbase-Init is not installed, you cannot configure an ECS. As a result, you can only use the password in the image file to log in to the ECS.
- By default, ECSs created from a public image have Cloudbase-Init installed. You do not need to install or configure Cloudbase-Init on such ECSs.
- For ECSs created from external image files, install and configure Cloudbase-Init by performing the operations in this section.

Prerequisites

- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Install Cloudbase-Init

1. On the Windows **Start** menu, choose **Control Panel > Programs > Programs and Features** and check whether Cloudbase-Init is installed.
 - If yes, go to [Configure Cloudbase-Init](#).
 - If no, go to the next step.
2. Check whether the version of the OS is Windows desktop.
 - If yes, go to [3](#).
 - If the OS is Windows Server, go to [4](#).
3. Enable the administrator account (Windows 7 is used as an example).
 - a. Click **Start** and choose **Control Panel > System and Security > Administrative Tools**.
 - b. Double-click **Computer Management**.
 - c. Choose **System Tools > Local Users and Groups > Users**.
 - d. Right-click **Administrator** and select **Properties**.
 - e. Deselect **Account is disabled**.
4. Download the Cloudbase-Init installation package.

Download the Cloudbase-Init installation package of the appropriate version based on the OS architecture from the Cloudbase-Init official website (<http://www.cloudbase.it/cloud-init-for-windows-instances/>).

Cloudbase-Init has two versions: stable and beta.

To obtain the stable version, visit the following paths:

 - 64-bit: https://www.cloudbase.it/downloads/CloudbaseInitSetup_Stable_x64.msi
 - 32-bit: https://www.cloudbase.it/downloads/CloudbaseInitSetup_Stable_x86.msi

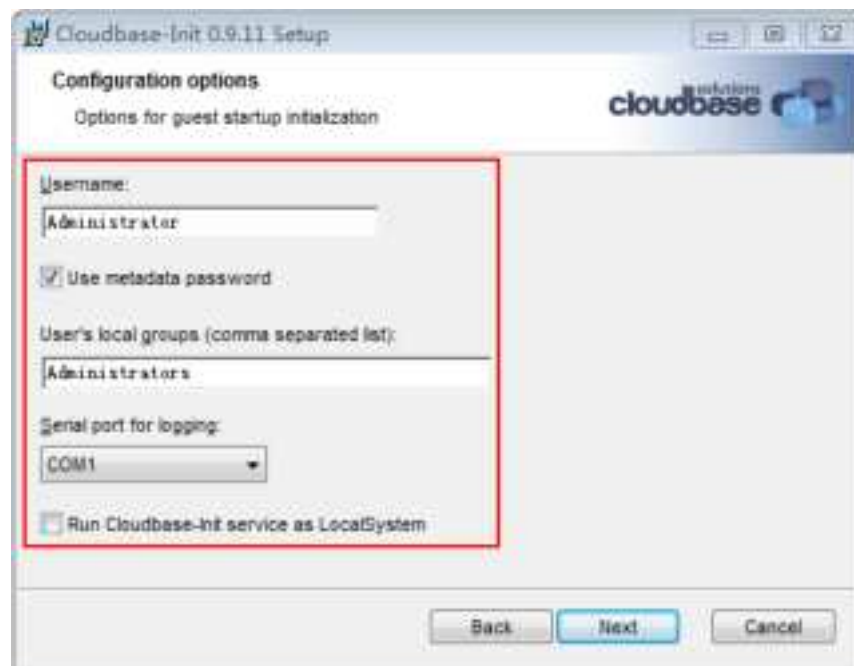
To obtain the beta version, visit the following paths:

 - 64-bit: https://www.cloudbase.it/downloads/CloudbaseInitSetup_x64.msi
 - 32-bit: https://www.cloudbase.it/downloads/CloudbaseInitSetup_x86.msi
5. Double-click the Cloudbase-Init installation package.
6. Click **Next**.
7. Select **I accept the terms in the License Agreement** and click **Next**.
8. Retain the default path and click **Next**.
9. In the **Configuration options** window, enter **Administrator** for **Username**, select **COM1** for **Serial port for logging**, and ensure that **Run Cloudbase-Init service as LocalSystem** is not selected.

NOTE

The version number shown in the figure is for reference only.

Figure 4-5 Configuring parameters



10. Click **Next**.
11. Click **Install**.
12. In the **Files in Use** dialog box, select **Close the application and attempt to restart them** and click **OK**.
13. Check whether the version of the OS is Windows desktop.
 - If yes, go to [15](#).
 - If no, go to [14](#).
14. In the **Completed the Cloudbase-Init Setup Wizard** window, ensure that neither option is selected.

Figure 4-6 Completing the Cloudbase-Init installation



NOTE

The version number shown in the figure is for reference only.

15. Click **Finish**.

Configure Cloudbase-Init

1. Edit configuration file **C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf** in the Cloudbase-Init installation path.
 - a. Add **netbios_host_name_compatibility=false** to the last line of the file so that the hostname supports a maximum of 63 characters.

NOTE

NetBIOS contains no more than 15 characters due to Windows system restrictions.

- b. Add **metadata_services=cloudbaseinit.metadata.services.httpservice.HttpService** to enable the agent to access the IaaS OpenStack data source.
 - c. (Optional) Add the following configuration items to configure the number of retry times and interval for obtaining metadata:

```
retry_count=40
retry_count_interval=5
```
 - d. (Optional) Add the following configuration item to prevent metadata network disconnections caused by the default route added by Windows:

```
[openstack]
add_metadata_private_ip_route=False
```
 - e. **(Optional)** When the Cloudbase-Init version is 0.9.12 or later, you can customize the length of the password.

Change the value of **user_password_length** to customize the password length.

2. Release the current DHCP address so that the created ECS can obtain the correct addresses.

In the Windows command line, run the following command to release the current DHCP address:

ipconfig /release

 NOTE

This operation will interrupt network connection and adversely affect ECS use. The network will automatically recover after the ECS is started again.

3. When creating an image using a Windows ECS, you need to change the SAN policy of the ECS to **OnlineAll**. Otherwise, EVS disks attached to the ECSs created from the image may be offline.

Windows has three types of SAN policies: **OnlineAll**, **OfflineShared**, and **OfflineInternal**.

Table 4-1 SAN policies

Type	Description
OnlineAll	All newly detected disks are automatically brought online.
OfflineShared	All disks on sharable buses, such as iSCSI and FC, are left offline by default, while disks on non-sharable buses are kept online.
OfflineInternal	All newly detected disks are left offline.

- a. Execute **cmd.exe** and run the following command to query the current SAN policy of the ECS using DiskPart:

diskpart

- b. Run the following command to view the SAN policy of the ECS:

san

- If the SAN policy is **OnlineAll**, run the **exit** command to exit DiskPart.
- If the SAN policy is not **OnlineAll**, go to [3.c](#).

- c. Run the following command to change the SAN policy of the ECS to **OnlineAll**:

san policy=onlineall

4.4 Running Sysprep

Scenarios

Running Sysprep ensures that an ECS has a unique SID after it is added to a domain.

After installing Cloudbase-Init on an ECS, you need to decide whether the ECS needs to be added to a domain or whether it must have a unique SID. If yes, run Sysprep as instructed in this section.

Prerequisites

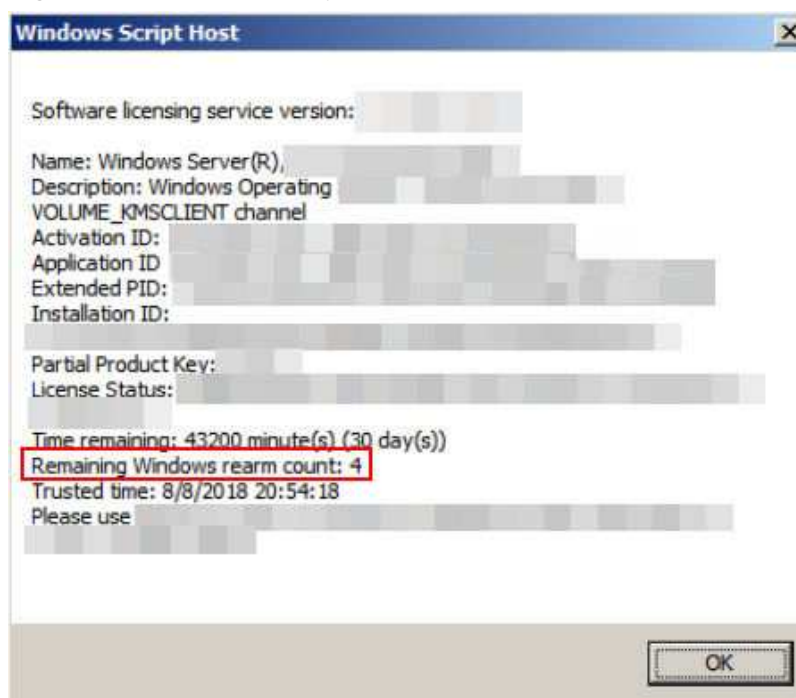
- Run Sysprep as the administrator.
- For a newly activated Windows ECS, you can run Sysprep only once at a time.
- If an ECS is created from an image file, only Sysprep provided by the image file can be used. In addition, Sysprep must always reside in the **%WINDIR%\system32\sysprep** directory.
- Windows must be in the activated state, and the remaining Windows rearm count must be greater than or equal to 1. Otherwise, the Sysprep encapsulation cannot be executed.

Run the following command in the Windows command line and check how many times you can run Sysprep in the displayed **Windows Script Host** dialog box:

slmgr.vbs /dlv

If the value of **Remaining Windows rearm count** is **0**, you cannot run Sysprep.

Figure 4-7 Windows Script Host

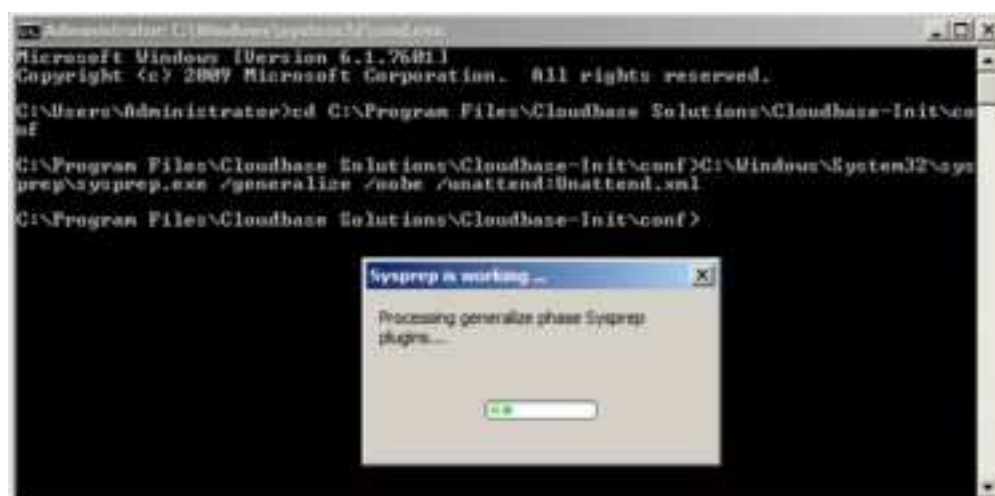


Procedure

1. Enter the Cloudbase-Init installation directory.
C:\Program Files\Cloudbase Solutions is used as an example of the Cloudbase-Init installation directory. Switch to the root directory of drive C and run the following command to enter the installation directory:
cd C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf
2. Run the following command to encapsulate Windows:
C:\Windows\System32\sysprep\sysprep.exe /generalize /oobe /unattend:Unattend.xml

CAUTION

- Ensure that **/unattend:Unattend.xml** is contained in the preceding command. Otherwise, the username, password, and other important configuration information of the ECS will be reset, and you must configure the OS manually when you use ECSs created from the Windows private image.
 - After this command is executed, the ECS will be automatically stopped. After the ECS is stopped, use the ECS to create an image. ECSs created using the image have unique SIDs. If you restart a Windows ECS on which Sysprep has been executed, Sysprep takes effect only for the current ECS. Before creating an image using the ECS, you must run Sysprep again.
 - For Windows Server 2012 and Windows Server 2012 R2, the administrator password of the ECS will be deleted after Sysprep is executed on the ECS. You need to log in to the ECS and reset the administrator password. In this case, the administrator password set on the management console will be invalid. Keep the password you set secure.
 - If a domain account is required for logins, run Sysprep on the ECS before using it to create a private image. For details about the impact of running Sysprep, see [Why Is Sysprep Required for Creating a Private Image from a Windows ECS?](#)
 - The Cloudbase-Init account of a Windows ECS is an internal account of the Cloudbase-Init agent. This account is used for obtaining metadata and completing relevant configuration when the Windows ECS starts. If you modify or delete this account, or uninstall the Cloudbase-Init agent, you will be unable to inject initial custom information into an ECS created from a Windows private image. Therefore, you are not advised to modify or delete the Cloudbase-Init account.
-

Figure 4-8 Running Sysprep

Follow-up Operations

1. Create a private image from the ECS on which Sysprep is executed. For details, see [Creating a System Disk Image from a Windows ECS](#).
2. You can use the image to create ECSs. Each ECS has a unique SID.

Run the following command to query the ECS SID:

whoami /user

Figure 4-9 ECS SID before Sysprep is executed**Figure 4-10** ECS SID after Sysprep is executed

4.5 Installing Special Windows Drivers

Scenarios

Before using some types of ECSs to create private images, you need to install special drivers on the ECSs.

GPU Driver

If you want to use the created private image to create GPU-accelerated ECSs, install a proper GPU driver for the image to enable GPU acceleration. There are two types of NVIDIA Tesla GPU drivers for GPU-accelerated ECSs, Tesla and GRID/vGPU drivers.

- To use graphics acceleration, such as OpenGL, DirectX, or Vulkan, install the GRID/vGPU driver and separately configure a GRID license. The GRID/vGPU driver with a vDWS license also supports CUDA for both computing and graphics acceleration.
- To use NVIDIA CUDA computing acceleration, install the Tesla driver.

Table 4-2 Installing the GRID driver

ECS Type	How to Install the Driver
G1	For details, see Instances > Installing a Driver and Toolkit > Installing a GRID Driver on a GPU-accelerated ECS > Downloading GRID Driver and Software License Packages in <i>Elastic Cloud Server User Guide (Paris Region)</i> .
G3	For details, see Instances > Installing a Driver and Toolkit > Installing a GRID Driver on a GPU-accelerated ECS > Downloading GRID Driver and Software License Packages in <i>Elastic Cloud Server User Guide (Paris Region)</i> .
G2	Log in at http://www.nvidia.com/Download/index.aspx?lang=en-us. You are advised to select the latest CUDA Toolkit version. NOTICE After the GPU driver is installed, run the following command to switch the GPU working mode and restart the ECS (for example, the GPU driver is installed in C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi.exe): "C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi.exe" -dm 0

SR-IOV NIC Driver

If you want to use the created private image to create G2 ECSs, install the SR-IOV NIC driver for the image to improve performance and scalability.

To download the SR-IOV driver, log in at <https://downloadcenter.intel.com/search?keyword=Intel++Ethernet+Connections+CD>. You are advised to select version 20.4.1 or later.

If error "No Intel adapter found" occurs during the driver installation, refer to [What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?](#) for troubleshooting.

5 Linux Operations

5.1 Setting the NIC to DHCP

Scenarios

If a private image is created from an ECS or external image file and the VM where the ECS or external image file is located is configured with a static IP address, you need to change the NIC attribute to DHCP so that the new ECSs created from the private image can dynamically obtain an IP address.

The configuration method varies depending on OSs.

NOTE

When registering an external image file as a private image, configure DHCP on the VM where the external image file is located. You are advised to configure DHCP on the VM and then export the image file.

Prerequisites

You have logged in to the ECS used to create a Windows private image.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Procedure

This section uses Ubuntu 14.04 as an example to describe how to query and configure NIC attributes of an ECS.

1. Run the following command on the ECS to open the **/etc/network/interfaces** file using the vi editor and query the IP address obtaining mode:
vi /etc/network/interfaces
 - If DHCP has been configured on all NICs, enter **:q** to exit the vi editor.

Figure 5-1 DHCP IP address obtaining mode

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet dhcp
```

- If static IP addresses are set on the NICs, go to [2](#).

Figure 5-2 Static IP address obtaining mode

```
auto lo
iface lo inet loopback
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.1.109
netmask 255.255.255.0
gateway 192.168.1.1
```

2. Press **i** to enter editing mode.
3. Delete the static IP address configuration and configure DHCP for the NICs.
You can insert a number sign (#) in front of each line of static IP address configuration to comment it out.

Figure 5-3 Configuring DHCP on a NIC

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
```

If the ECS has multiple NICs, you must configure DHCP for all the NICs.

Figure 5-4 Configuring DHCP on multiple NICs

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet dhcp
```

4. Press **Esc**, enter **:wq**, and press **Enter**.
The system saves the configuration and exits the vi editor.

Related Operations

Configure DHCP to enable the ECS to obtain IP addresses continuously.

- For CentOS and EulerOS, use the vi editor to add **PERSISTENT_DHCLIENT="y"** to configuration file **/etc/sysconfig/network-scripts/ifcfg-ethX**.
- For SUSE Linux Enterprise, use the vi editor to set **DHCLIENT_USE_LAST_LEASE** to **no** in the configuration file **/etc/sysconfig/network/dhcp**.
- For Ubuntu 12.04, upgrade dhclient to ISC dhclient 4.2.4 so that the NIC can consistently obtain IP addresses from the DHCP server. To perform the upgrade, you need to install isc-dhcp-server first.

5.2 Deleting Files in the Network Rule Directory

Scenarios

To prevent NIC name drift when you use a private image to create ECSs, you need to delete files in the network rule directory of the VM where the ECS or image file is located during the private image creation.

NOTE

When registering an external image file as a private image, delete files in the network rule directory on the VM where the external image file is located. You are advised to delete the files on the VM and then export the image file.

Prerequisites

An OS and VirtIO drivers have been installed on the ECS.

Procedure

1. Run the following command to query the files in the network rule directory:
ls -l /etc/udev/rules.d
2. Run the following commands to delete the files whose names contain **persistent** and **net** from the network rule directory:

Example:

rm /etc/udev/rules.d/*30-net_persistent-names.rules*

rm /etc/udev/rules.d/*70-persistent-net.rules*

The italic content in the commands varies depending on your environment.

NOTE

For CentOS 6 images, to prevent NIC name drift, you need to create an empty rules configuration file.

Example:

touch /etc/udev/rules.d/75-persistent-net-generator.rules //Replace 75 with the actual value in the environment.

3. Delete network rules.

- If the OS uses the initrd system image, perform the following operations:
 - i. Run the following command to check whether the initrd image file whose name starts with **initrd** and ends with **default** contains the **persistent** and **net** network device rule files (replace the italic content in the following command with the actual OS version):
lsinitrd /boot/initrd-2.6.32.12-0.7-default |grep persistent|grep net
 - o If no, no further action is required.
 - o If yes, go to [3.ii](#).
 - ii. Run the following command to back up the initrd image files (replace the italic part in the following command with the actual OS version):
cp /boot/initrd-2.6.32.12-0.7-default /boot/initrd-2.6.32.12-0.7-default_bak
 - iii. Run the following command to generate the initrd file again:
mkinitrd
- If the OS uses the initramfs system image (such as Ubuntu), perform the following operations:
 - i. Run the following command to check whether the initramfs image file whose name starts with **initrd** and ends with **generic** contains persistent and net rule files.
lsinitramfs /boot/initrd.img-3.19.0-25-generic|grep persistent|grep net
 - o If no, no further action is required.
 - o If yes, go to [3.ii](#).
 - ii. Run the following command to back up the initrd image files:
cp /boot/initrd.img-3.19.0-25-generic /boot/initrd.img-3.19.0-25-generic_bak
 - iii. Run the following command to generate the initramfs image files again:
update-initramfs -u

5.3 Installing Cloud-Init

Scenarios

To ensure that you can use the user data injection function to inject initial custom information into ECSs created from a private image (such as setting the ECS login password), install Cloud-Init on the ECS used to create the image.

- You need to download Cloud-Init from its official website. Therefore, you must bind an EIP to the ECS.
- If Cloud-Init is not installed, you cannot configure an ECS. As a result, you can only use the password in the image file to log in to the created ECSs.
- By default, ECSs created from a public image have Cloud-Init installed. You do not need to install or configure Cloud-Init on such ECSs.
- For ECSs created using an external image file, install and configure Cloud-Init by performing the operations in this section. For how to configure Cloud-Init, see [Configuring Cloud-Init](#).

Prerequisites

- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Procedure

1. Check whether Cloud-Init has been installed.
For details, see [Check Whether Cloud-Init Has Been Installed](#).
2. Install Cloud-Init.
You can install Cloud-Init using either of the following methods:
[\(Recommended\) Install Cloud-Init Using the Official Installation Package](#)
and [Install Cloud-Init Using the Official Source Code Package and pip](#).

Check Whether Cloud-Init Has Been Installed

Perform the operations provided here to check whether Cloud-Init has been installed.

The methods of checking whether Cloud-Init is installed vary depending on the OSs. Take CentOS 6 as an example. Run the following command to check whether Cloud-Init is installed:

`rpm -qa |grep cloud-init`

If information similar to the following is displayed, Cloud-Init has been installed:

```
cloud-init-0.7.5-10.el6.centos.2.x86_64
```

If Cloud-Init has been installed, perform the following operations:

- Check whether to use the certificate in the ECS OS. If the certificate is no longer used, delete it.
 - If the certificate is stored in a directory of user **root**, for example, `/$path/$to/$root/.ssh/authorized_keys`, run the following commands:
`cd /root/.ssh`
`rm authorized_keys`
 - If the certificate is not stored in a directory of user **root**, for example, `/$path/$to/$none-root/.ssh/authorized_keys`, run the following commands:
`cd /home/centos/.ssh`

rm authorized_keys

- Run the following command to delete the cache generated by Cloud-Init and ensure that the ECS created from the private image can be logged in by using the certificate:

sudo rm -rf /var/lib/cloud/*

NOTE

Do not restart the ECS after performing the configuration. Otherwise, you need to configure it again.

(Recommended) Install Cloud-Init Using the Official Installation Package

The method of installing Cloud-Init on an ECS varies depending on the OS. Perform the installation operations as user **root**.

The following describes how to install Cloud-Init on an ECS running SUSE Linux, CentOS, Fedora, Debian, and Ubuntu. For other OS types, install the required type of Cloud-Init. For example, you need to install coreos-cloudinit on ECSs running CoreOS.

- SUSE Linux

Paths for obtaining the Cloud-Init installation package for SUSE Linux

<http://ftp5.gwdg.de/pub/opensuse/repositories/Cloud:/Tools/>

<http://download.opensuse.org/repositories/Cloud:/Tools/>

NOTE

Select the required repo installation package in the provided paths.

Take SUSE Enterprise Linux Server 12 as an example. Perform the following steps to install Cloud-Init:

- a. Log in to the ECS used to create a Linux private image.
- b. Run the following command to install the network installation source for SUSE Enterprise Linux Server 12:

zypper ar http://ftp5.gwdg.de/pub/opensuse/repositories/Cloud:/Tools/SLE_12_SP3/Cloud:Tools.repo

- c. Run the following command to update the network installation source:

zypper refresh

- d. Run the following command to install Cloud-Init:

zypper install cloud-init

- e. Run the following commands to enable Cloud-Init to automatically start upon system boot:

- SUSE 11

chkconfig cloud-init-local on; chkconfig cloud-init on; chkconfig cloud-config on; chkconfig cloud-final on

service cloud-init-local status; service cloud-init status; service cloud-config status; service cloud-final status

- SUSE 12 and openSUSE 12/13/42

systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

```
systemctl status cloud-init-local.service cloud-init.service cloud-  
config.service cloud-final.service
```

CAUTION

For SUSE and openSUSE, perform the following steps to disable dynamic change of the ECS name:

1. Run the following command to open the **dhcp** file using the vi editor:
vi etc/sysconfig/network/dhcp
2. Change the value of **DHCLIENT_SET_HOSTNAME** in the **dhcp** file to **no**.

- **CentOS**

Table 5-1 lists the Cloud-Init installation paths for CentOS. Select the required installation package from the following addresses.

Table 5-1 Cloud-Init installation package addresses

OS Type	Version	How to Obtain
CentOS	6 32-bit	https://archives.fedoraproject.org/pub/archive/epel/6/i386/
	6 64-bit	https://archives.fedoraproject.org/pub/archive/epel/6/x86_64/
	7 64-bit	https://archives.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release-7-13.noarch.rpm

Run the following commands to install Cloud-Init on an ECS running CentOS 6.5 64-bit (example):

```
yum install https://archives.fedoraproject.org/pub/archive/epel/6/x86_64/  
epel-release-xx-xx.noarch.rpm
```

```
yum install cloud-init
```

NOTE

xx-xx indicates the version of Extra Packages for Enterprise Linux (EPEL) required by the current OS.

- **Fedora**

Before installing Cloud-Init, ensure that the network installation source address has been configured for the OS by checking whether the **/etc/yum.repo.d/fedora.repo** file contains the installation source address of the software package. If the file does not contain the address, configure the address by following the instructions on the Fedora official website.

Run the following command to install Cloud-Init:

```
yum install cloud-init
```

- Debian and Ubuntu
Before installing Cloud-Init, ensure that the network installation source address has been configured for the OS by checking whether the **/etc/apt/sources.list** file contains the installation source address of the software package. If the file does not contain the address, configure the address by following the instructions on the Debian or Ubuntu official website.
Run the following commands to install Cloud-Init:
apt-get update
apt-get install cloud-init

Install Cloud-Init Using the Official Source Code Package and pip

The following operations use Cloud-Init 0.7.9 as an example to describe how to install Cloud-Init.

1. Download the **cloud-init-0.7.9.tar.gz** source code package (version 0.7.9 is recommended) and upload it to the **/home/** directory of the ECS.
Download **cloud-init-0.7.9.tar.gz** from the following path:
<https://launchpad.net/cloud-init/trunk/0.7.9/+download/cloud-init-0.7.9.tar.gz>
2. Create a **pip.conf** file in the **~/.pip/** directory and edit the following content:

NOTE

If the **~/.pip/** directory does not exist, run the **mkdir ~/.pip** command to create it.

```
[global]
index-url = https://<$mirror>/simple/
trusted-host = <$mirror>
```

3. Run the following command to install the downloaded Cloud-Init source code package (select **--upgrade** as needed during installation):
pip install [--upgrade] /home/cloud-init-0.7.9.tar.gz
4. Run the **cloud-init -v** command. Cloud-Init is installed successfully if the following information is displayed:
cloud-init 0.7.9
5. Enable Cloud-Init to automatically start upon system boot.
 - If the OS uses SysVinit to manage automatic start of services, run the following commands:
chkconfig --add cloud-init-local; chkconfig --add cloud-init; chkconfig --add cloud-config; chkconfig --add cloud-final
chkconfig cloud-init-local on; chkconfig cloud-init on; chkconfig cloud-config on; chkconfig cloud-final on
service cloud-init-local status; service cloud-init status; service cloud-config status; service cloud-final status
 - If the OS uses Systemd to manage automatic start of services, run the following commands:
systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service
systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

 CAUTION

If you install Cloud-Init using the official source code package and pip, pay attention to the following:

1. Add user **syslog** to the **adm** group during the installation. If user **syslog** exists, add it to the **adm** group. For some OSs (such as CentOS and SUSE), user **syslog** may not exist. Run the following commands to create user **syslog** and add it to the **adm** group:

```
useradd syslog
```

```
groupadd adm
```

```
usermod -g adm syslog
```

2. Change the value of **distro** in **system_info** in the **/etc/cloud/cloud.cfg** file based on the OS release version, such as **distro: ubuntu**, **distro: sles**, **distro: debian**, and **distro: fedora**.
-

5.4 Configuring Cloud-Init

Scenarios

You need to configure Cloud-Init after it is installed.

Prerequisites

- Cloud-Init has been installed.
- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Procedure

The following operations are required:

1. Configure Cloud-Init.
For details, see [Configure Cloud-Init](#).
2. Check whether Cloud-Init is successfully configured.
For details, see [Check the Cloud-Init Configuration](#).

Configure Cloud-Init

1. Configure the user permissions for logging in to the ECS. If you use a common account (not user **root**) to log in to the ECS, disable the SSH permissions of user **root** and remote login using a password to improve the ECS security.
 - You can remotely log in to the ECS using SSH and a key pair injected into your account. (It is recommended that you select the key pair login mode when creating an ECS.)
 - You can also use a random password to log in to the ECS through noVNC.Run the following command to open the **sshd_config** file using the vi editor:

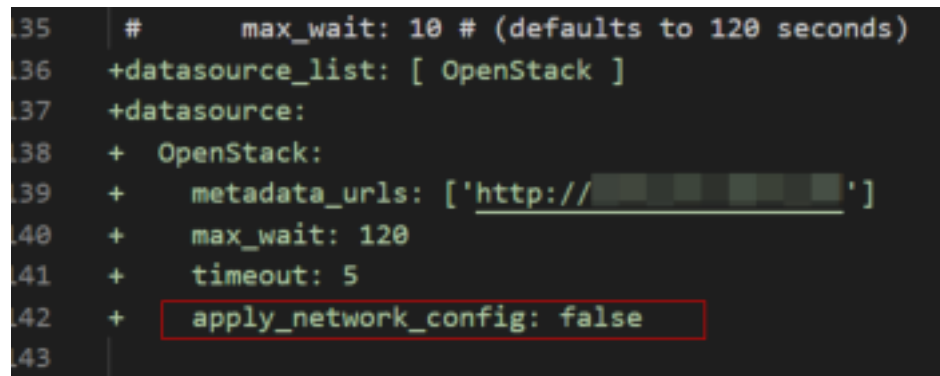
vi /etc/ssh/sshd_config

2. Change the value of **PasswordAuthentication** in the **sshd_config** file to **no**.

NOTE

For SUSE and openSUSE, change the values of the following parameters in the **sshd_config** file to **no**:

- PasswordAuthentication
 - ChallengeResponseAuthentication
3. Run the following command to open the **cloud.cfg** file using the vi editor:
vi /etc/cloud/cloud.cfg
 4. (Optional) In **/etc/cloud/cloud.cfg**, set **apply_network_config** to **false**.
This step is only for Cloud-Init 18.3 or later.

Figure 5-5 Example configuration

```
35 # max_wait: 10 # (defaults to 120 seconds)
36 +datasource_list: [ OpenStack ]
37 +datasource:
38 + OpenStack:
39 + metadata_urls: ['http://[redacted]']
40 + max_wait: 120
41 + timeout: 5
42 + apply_network_config: false
43
```

5. Disable the SSH permissions of user **root** in **/etc/cloud/cloud.cfg**, add a common user (which is used for logging in to the ECS using VNC), and configure a password for the added user and assign sudo permissions to it.

NOTE

For Ubuntu and Debian, set the value of **manage_etc_hosts** in the **/etc/cloud/cloud.cfg** file to **localhost**. Otherwise, switching to user **root** from a user other than **root** may time out.

Take Ubuntu as an example.

- Run the following command to create script **/etc/cloud/set_linux_random_password.sh**, which is executable and can be used to generate random passwords:

cat /etc/cloud/set_linux_random_password.sh

The file content is as follows:

```
#!/bin/bash

password=$(cat /dev/urandom | tr -dc 'A-Za-z0-9!@#%&+= ' | head -c 9)

echo "linux:$password" | chpasswd
sed -i -e '/^Login/d' /etc/issue
sed -i -e '/^Initial/d' /etc/issue
sed -i -c -e '/^$/d' /etc/issue
echo -e "\nInitial login with linux:$password\n" >> /etc/issue
```

 NOTE

You can run the **chmod +x /etc/cloud/set_linux_random_password.sh** command to add execute permissions of **set_linux_random_password.sh**.

- After you log in to the ECS, run the following commands to add a user-friendly prompt "Please change password for user linux after first login."

```
echo -e '\e[1;31m#####\n\e[0m' > /etc/motd
```

```
echo -e '\e[1;31m# Important !!! #\e[0m' >> /etc/motd
```

```
echo -e '\e[1;31m# Please change password for user linux after first login. #\e[0m' >> /etc/motd
```

```
echo -e '\e[1;31m#####\n\e[0m' >> /etc/motd
```

```
echo -e " " >> /etc/motd
```

6. Add a common login user, set its password, assign sudo permissions to it, and use bootcmd to create a script used for generating a random password for each created ECS.

 CAUTION

Ensure that the configuration file format (such as alignment and spaces) is consistent with the provided example.

```
system_info:
# This will affect which distro class gets used
distro: rhel
# Default user name + that default users groups (if added/used)
default_user:
  name: linux //Username for login
  lock_passwd: False //Login using a password is enabled. Note that some OSs use value 0 to
enable the password login.
gecos: Cloud User
groups: users //Optional. Add users to other groups that have been configured in /etc/group.
passwd: $6$I63DBVKK
$Zh4lchiJR7NuZvtJHsYBQJlg5RoQCRLS1X2Hsgj2s5JwXI7KUO1we8WYcwbzeaS2VNpRmNo28vmxx
CyU6LwoD0
sudo: ["ALL=(ALL) NOPASSWD:ALL"] // Assign the root rights to the user.
shell: /bin/bash //Execute shell in bash mode.
# Other config here will be given to the distro class and/or path classes
paths:
  cloud_dir: /var/lib/cloud/
  templates_dir: /etc/cloud/templates/
ssh_svcname: sshd

bootcmd:
- [cloud-init-per, instance, password, bash,
/etc/cloud/set_linux_random_password.sh]
```

 NOTE

The value of **passwd** is encrypted using SHA512 (which is used as an example). For more details, see <https://cloudinit.readthedocs.io/en/latest/topics/examples.html>.

For details about how to encrypt a password and generate ciphertext, see the following (encrypting password **cloud.1234** is used as an example):

```
[root@** ~]# python -c "import crypt, getpass, pwd; print crypt.mksalt()"
$6$I63DBVKK
[root@** ~]# python -c "import crypt, getpass, pwd; print crypt.crypt('cloud.1234', '$6$I63DBVKK')"
$6$I63DBVKK
$Zh4lchiJR7NuZvtJHsYBQJlg5RoQCRLS1X2Hsgj2s5JwXI7KUO1we8WYcwbzeaS2VNpRmNo28vmxx
CyU6LwoD0
```

7. Enable the agent to access the IaaS OpenStack data source.

Add the following information to the last line of **/etc/cloud/cloud.cfg**:

```
datasource_list: [ OpenStack ]
datasource:
  OpenStack:
    metadata_urls: ['http://169.254.169.254']
    max_wait: 120
    timeout: 5
```

 NOTE

- You can decide whether to set **max_wait** and **timeout**. The values of **max_wait** and **timeout** in the preceding example are only for reference.
- If the OS version is earlier than Debian 8 or CentOS 5, you cannot enable the agent to access the IaaS OpenStack data source.
- The default zeroconf route must be disabled for CentOS and EulerOS ECSs for accurate access to the IaaS OpenStack data source.

```
echo "NOZEROCONF=yes" >> /etc/sysconfig/network
```

8. Prevent Cloud-Init from taking over the network in **/etc/cloud/cloud.cfg**.

If the Cloud-Init version is 0.7.9 or later, add the following content to **/etc/cloud/cloud.cfg**:

```
network:
  config: disabled
```

 NOTE

The added content must be in the YAML format.

Figure 5-6 Preventing Cloud-Init from taking over the network

```
users:
  - default

disable_root: 1
ssh_pwauth: 0

datasource_list: [ OpenStack ]
datasource:
  OpenStack:
    metadata_urls: ['http://[REDACTED]']
    max_wait: 120
    timeout: 50

network:
  config: disabled
```

9. Modify the **cloud_init_modules** configuration file.
Move **ssh** from the bottom to the top to speed up the SSH login.

Figure 5-7 Speeding up the SSH login to the ECS

```
cloud_init_modules:
- ssh
- migrator
- bootcmd
- write-files
- growpart
- resizefs
- set_hostname
- update_hostname
- update_etc_hosts
- rsyslog
- users-groups
```

10. Modify the configuration so that the hostname of the ECS created from the image does not contain the **.novalocal** suffix and can contain a dot (.).
- a. Run the following command to modify the **__init__.py** file:

vi /usr/lib/python2.7/site-packages/cloudinit/sources/__init__.py

Press **i** to enter editing mode. Search for **toks**. The following information is displayed:

```
if toks:
    toks = str(toks).split('.')
else:
    toks = ["ip-%s" % lhost.replace(".", "-")]
else:
    toks = lhost.split(".novalocal")

if len(toks) > 1:
    hostname = toks[0]
    #domain = ''.join(toks[1:])
else:
    hostname = toks[0]
```

```
if fqdn and domain != defdomain:
    return "%s.%s" % (hostname, domain)
else:
    return hostname
```

After the modification is complete, press **Esc** to exit the editing mode and enter **:wq!** to save the configuration and exit.

Figure 5-8 Modifying the `__init__.py` file

```
192      # if there is an ipv4 address in 'local-hostname', then
193      # make up a hostname (LP: #475354) in format ip->xx.xx.xx.xx
194      lhost = self.metadata['local-hostname']
195      if util.is_ipof(lhost):
196          toks = []
197          if resolve_ip:
198              toks = util.gethostbyaddr(lhost)
199
200          if toks:
201              toks = str(toks).split('.')
202          else:
203              toks = ["ip->xx" % lhost.replace(".", "-")]
204      else:
205          toks = lhost.split("nowlocal")
206
207      if len(toks) > 1:
208          hostname = toks[0]
209          domain = '.'.join(toks[1:])
210      else:
211          hostname = toks[0]
212
213      if fqdn and domain != defdomain:
214          return "%s.%s" % (hostname, domain)
215      else:
216          return hostname
```

- b. Run the following command to switch to the **cloudinit/sources** folder:
cd /usr/lib/python2.7/site-packages/cloudinit/sources/
 - c. Run the following commands to delete the **__init__.pyc** file and the optimized **__init__.pyo** file:
rm -rf __init__.pyc
rm -rf __init__.pyo
 - d. Run the following commands to clear the logs:
rm -rf /var/lib/cloud/*
rm -rf /var/log/cloud-init*
11. Run the following command to edit the **/etc/cloud/cloud.cfg.d/05_logging.cfg** file to use **cloudLogHandler** to process logs:
vim /etc/cloud/cloud.cfg.d/05_logging.cfg

Figure 5-9 Setting the parameter value to **cloudLogHandler**

```
[logger_cloudinit]
level=DEBUG
qualname=cloudinit
handlers=cloudLogHandler
propagate=1
```

Check the Cloud-Init Configuration

Run the following command to check whether Cloud-Init has been properly configured:

cloud-init init --local

If Cloud-Init has been properly installed, the version information is displayed and no error occurs. For example, messages indicating lack of files will not be displayed.

NOTE

(Optional) Run the following command to set the password validity period to the maximum:

chage -M 99999 \$user_name

user_name is a system user, such as user **root**.

You are advised to set the password validity period to **99999**.

5.5 Detaching Data Disks from an ECS

Scenarios

If multiple data disks are attached to the ECS used to create a private image, ECSs created from the image may be unavailable. Therefore, you need to detach all data disks from the ECS before using it to create a private image.

This section describes how to detach all data disks from an ECS.

Prerequisites

You have logged in to the ECS used to create a Linux private image.

Procedure

1. Check whether the ECS has data disks.

Run the following command to check the number of disks attached to the ECS:

fdisk -l

- If the number is greater than 1, the ECS has data disks. Go to [2](#).
- If the number is equal to 1, no data disk is attached to the ECS. Go to [3](#).

2. Run the following command to check the data disks attached to the ECS:

mount

- If the command output does not contain any EVS disk information, no EVS data disks need to be detached.

```
/dev/vda1 on / type ext4 (rw,relatime,data=ordered)
```

- If information similar to the following is displayed, go to [3](#):

```
/dev/vda1 on / type ext4 (rw,relatime,data=ordered)
```

```
/dev/vdb1 on /mnt/test type ext4 (rw,relatime,data=ordered)
```

3. Delete the configuration information in the **fstab** file.

- a. Run the following command to edit the **fstab** file:

vi /etc/fstab

- b. Delete the disk configuration from the **fstab** file.

The **/etc/fstab** file contains information about the file systems and storage devices automatically attached to the ECS when the ECS starts. The configuration about data disks automatically attached to the ECS needs to be deleted, for example, the last line shown in the following figure.

Figure 5-10 EVS disk configuration in the **fstab** file



```
[root@ecs-bf78 ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Wed Feb 27 08:58:18 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=4c2c090d-4228-49fc-8c0e-3928b1b7287c / ext4 defaults 1 1
UUID=9c29104b-3108-4421-a2d7-102f80ec7a85 /mnt/test ext4 defaults 1 1
```

4. Run the following command to detach data disks from the ECS:

Run the following command to detach the disks:

umount /dev/vdb1

5. Run the following command to check the data disks attached to the ECS:

mount

If the command output contains no information about the data disks, they have been detached from the ECS.

6 FAQs

6.1 Image Consulting

6.1.1 How Do I Select an Image?

When creating an ECS or BMS, you can select an image based on the following factors:

- [Region and AZ](#)
- [Image Type](#)
- [OS](#)

Region and AZ

An image is a regional resource. You cannot use an image to create an instance across regions. For example, when creating an instance in region A, you can select an image only from region A. For more regions, see [Region and AZ](#).

Image Type

Images are classified into public images, private images, and shared images. A private image can be a system disk image, data disk image, or full-ECS image. For details, see [What Is Image Management Service?](#)

OS

When selecting an OS, consider the following factors:

- Architecture types

System Architecture	Applicable Memory	Constraints
32-bit	Smaller than 4 GB	• If the instance memory is greater than 4 GB, the 32-bit OS cannot be used. • A 32-bit OS allows addressing only within a 4 GB memory range. An OS with more than 4 GB memory cannot be accessed.
64-bit	4 GB or larger	If your application requires more than 4 GB memory or the memory may need to be expanded to more than 4 GB, use a 64-bit OS.

- OS types

OS Type	Applicable Scenario	Constraints
Windows	• Runs programs developed on Windows (for example, .NET). • Supports databases such as SQL Server. (You need to install the database.)	The system disk must be no less than 1 GB or larger, and the memory must be no less than 1 GB.
Linux	• Runs high-performance server applications (for example, Web) and supports common programming languages such as PHP and Python. • Supports databases such as MySQL. (You need to install the database.)	The system disk must be no less than 1 GB or larger, and the memory must be no less than 512 MB.

6.1.2 How Do I Increase the Image Quota?

What Is Quota?

Quotas are enforced for service resources on the platform to prevent unforeseen spikes in resource usage. Quotas can limit the number or amount of resources available to users, such as the maximum number of ECSs or EVS disks that can be created.

If the existing resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. In the upper right corner of the page, click  .
The **Service Quota** page is displayed.
3. View the used and total quota of each type of resources on the displayed page.
If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment.

If you need to adjust a quota, contact the administrator.

6.1.3 Can I Use Private Images of Other Tenants?

Yes.

Other tenants can share a private image with you. You can use it after accepting it. For details about image sharing, see [Sharing Specified Images](#).

6.2 Image Creation

6.2.1 Image Creation FAQs

How Many Private Images Can I Create Under an Account?

Currently, you can create a maximum of 100 private images under an account in a region.

Do I Have to Stop the ECS Before Using It to Create a Private Image?

No. You can create an image from a running ECS. However, if data is written to the ECS during image creation, the data will not be contained in the created image.

Where Can I View the Image Creation Progress? How Long Does It Take to Create an Image?

Log in to the management console. Choose **Computing > Image Management Service** and click the **Private Images** tab. View the image creation progress in the **Status** column.

The image creation involves the installation of Xen and KVM drivers, OS kernel loading, and GRUB boot configuration, which may take a long time. In addition, the network speed, image file type, and disk size have an impact on the image creation duration.

6.2.2 How Do I Create a Full-ECS Image Using an ECS That Has a Spanned Volume?

An ECS used to create a Windows full-ECS image cannot have a spanned volume. Otherwise, data may be lost when the full-ECS image is used to create ECSs.

If the ECS has a spanned volume, back up the data in the spanned volume and then delete this volume from the ECS. Use the ECS to create a full-ECS image. Use the full-ECS image to create an ECS. Then, use the backup to create a spanned volume for the new ECS if necessary.

NOTE

If a Linux ECS has a volume group or a logical volume consisting of multiple physical volumes, back up the data in the volume group or logical volume and delete the volume group or logical volume before using this ECS to create a full-ECS image. This prevents data loss.

6.2.3 Why Is Sysprep Required for Creating a Private Image from a Windows ECS?

Why Is Sysprep Required?

For a user that needs to be added to a domain and uses the domain account to log in to Windows, Sysprep is required before a private image is created. Otherwise, the image will contain information about the original ECS, especially the SID. ECSs with the same SID cannot be added to a domain. If Windows does not require any user or ECS to be added to a domain, you do not need to run Sysprep.

CAUTION

- Before running Sysprep, ensure that Windows is activated.
 - For details about Sysprep, visit [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc721940\(v=ws.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc721940(v=ws.10)?redirectedfrom=MSDN).
-

Restrictions on Running Sysprep

Sysprep can only be used for configuring a new Windows installation. You can run Sysprep multiple times to install and configure Windows. However, you can reset and activate a Windows OS only three times, and you are not allowed to use Sysprep to re-configure an existing Windows OS.

NOTE

In the Windows command line, enter the following command to check how many times you can run Sysprep in the displayed **Windows Script Host** dialog box:

slmgr /dlv

If the value of **Remaining Windows rearm count** is **0**, you cannot run Sysprep.

6.2.4 What Do I Do If I Cannot Create an Image in ZVHD2 Format Using an API?

Symptom

When you create a ZVHD2 image using an API, the image is created in the ZVHD format.

Solution

Check whether your token contains the **op_gated_ild** role (**op_gated_ild** is the OBT tag, which can be viewed in the body of the response message of the API used to obtain a user token). The ZVHD2 image has the lazy loading feature. If the current environment does not support this feature or this feature is in the OBT phase, the ZVHD2 image will fail to be created.

Contact the customer service to ensure that the current environment supports the lazy loading feature, obtain a new token, and use the new token to create an image.

6.3 Image Sharing

6.3.1 Image Sharing

How Many Tenants Can I Share an Image with at Most?

128

How Many Images Can Be Shared with Me at Most?

There is no limit.

Do Shared Images Affect My Private Image Quota?

No.

I Shared an Image to an Account But the Account Did Not Accept or Reject the Image. Will My Image Sharing Quota Be Consumed?

No.

Can I Use an Image I Have Shared with Others to Create an ECS?

Yes. After sharing an image with other tenants, you can still use the image to create an ECS and use the created ECS to create a private image.

When Can I Do If I Want to Use a Rejected Image?

If you have rejected an image shared by another tenant, but now want to use it, two methods are available:

- Method 1
Ask the image owner to add you to the tenants the image is shared with. For details, see [Adding Tenants Who Can Use Shared Images](#).
- Method 2
Accept the rejected image again. For details, see [Accepting Rejected Images](#).

6.4 OS

6.4.1 How Is BIOS Different from UEFI?

Table 6-1 Differences between the UEFI and BIOS boot modes

Boot Mode	Description	Highlight
BIOS	Basic Input Output System (BIOS) stores important basic input/output programs of ECSs, system settings, self-test programs upon system startup, and automatic startup programs.	Provides basic settings and control for ECSs.
UEFI	UEFI, an acronym for Unified Extensible Firmware Interface, is a specification that defines a software interface between an OS and platform firmware. UEFI can be used to automatically load an OS from a pre-boot operating environment.	Shortens the OS startup time and the time that the OS needs to recover from the sleep state.

6.4.2 How Do I Delete Redundant Network Connections from a Windows ECS?

Method 1

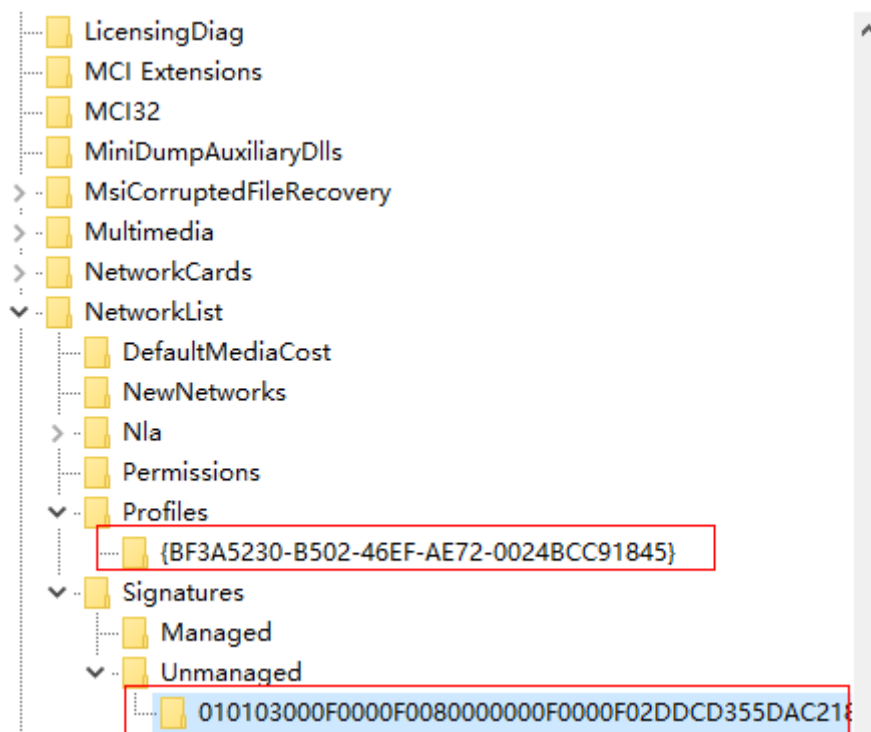
1. Press **Win+R**. In the displayed dialog box, enter **regedit** and press **Enter** to open the registry editor.
2. Open the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT \CurrentVersion\NetworkList\Profiles
Click each item under **Profiles** and query the **Data** column of **ProfileName** in the right pane.
3. Double-click **ProfileName** and set **Value Data** to the name of a new network.

4. Restart the ECS for the change to take effect.

Method 2

1. Press **Win+R**. In the displayed dialog box, enter **regedit** and press **Enter** to open the registry editor.
2. Open the following registry keys:
**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT
\CurrentVersion\NetworkList\Profiles**
**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT
\CurrentVersion\NetworkList\Signatures\Unmanaged**
3. Delete the directories shown in the following figure.

Figure 6-1 Registry directory



6.4.3 What Do I Do If an ECS Starts Slowly?

Symptom

If an ECS starts slowly, you can change the default timeout duration to speed up the startup.

Solution

1. Log in to the ECS.
2. Run the following command to switch to user **root**:
sudo su
3. Run the following command to query the version of the GRUB file:

```
rpm -qa | grep grub
```

Figure 6-2 Querying the GRUB file version

```
[root@iz1mm00t3174cau007 ~]# rpm -qa | grep grub
grub2-2.02-0.44.el7.centos.x86_64
```

4. Set **timeout** in the GRUB file to 0.
 - If the GRUB file version is earlier than 2:
Open **/boot/grub/grub.cfg** or **/boot/grub/menu.lst** and set **timeout** to 0.
 - If the GRUB file version is 2:
Open **/boot/grub2/grub.cfg** and set the value of **timeout** to 0.

Figure 6-3 Modifying the timeout duration

```
# boot --duro/sda
default 0
timeout 0
splashimage=(hd0,1)/boot/grub/splash.spm.gz
hidmenu
title CentOS (2.6.32-096.16.1.el6.x86_64)
    root (hd0,1)
    kernel /boot/vmlinuz-2.6.32-096.16.1.el6.x86_64 ro root=UUID=23c0f5f4-e8
15-4ba5-0cc0-8fe1234efc24 rd_NO_LUKS rd_NO_LDM LANG=en_US.UTF-8 rd_NO_MD SYSFONT
=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=en rd_NO_DM rhgb q
uit
```

6.4.4 What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?

Symptom

When the 20.4.1 driver package downloaded at Intel website <https://downloadcenter.intel.com/search?keyword=Intel++Ethernet+Connections+CD> was installed in a Windows 7 64bit ECS with SR-IOV passthrough enabled, the system displayed the message "No Intel adapter found".

Cause Analysis

The OS identifies an Intel 82599 passthrough NIC without a driver installed as an Ethernet controller. When the 20.4.1 driver package was installed, the OS did not identify the Intel NIC, leading to the error.

Solution

Run **Autorun.exe** in the folder where the 20.4.1 driver package is stored. Install a driver on the NIC before installing the driver package so that the NIC can be identified as an Intel 82599 virtual function (VF) device by the OS. Use either of the following methods to install the driver:

- Method 1: Update the version.
 - a. Download the 18.6 driver package at the Intel website.
 - b. Run **Autorun.exe**.
 - c. Run **Autorun.exe** in the folder where the 20.4.1 driver package is stored to update the driver.

- Method 2: Use the device manager.
 - a. Start the Windows resource manager. Right-click **Computer** and choose **Manage** from the shortcut menu. In the **Device Manager** window, locate the NIC. When the NIC has no driver installed, the NIC locates in **Other devices** and is named **Ethernet Controller**.
 - b. Right-click **Ethernet Controller** and choose **Update Driver Software**.
 - c. Click **Browse**, select the path where the driver package is stored, and click **Next**.
 - d. Locate the NIC in **Network Adapter** of **Device Manager**.
 - e. Run **Autorun.exe** to install the 20.4.1 driver package.

6.5 Image Importing

6.5.1 Can I Use Images in Formats Other Than Those Specified in This Document?

No. Currently, only the VMDK, VHD, RAW, QCOW2, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD formats are supported.

Images of the -flat.vmdk format and image file packages containing snapshot volumes or delta volumes are not supported. You can use qemu-img to convert the format of an image into a supported one before uploading it to the cloud platform.

NOTE

For how to install and use qemu-img on Windows, visit the following website:

<https://cloudbase.it/qemu-img-windows/>

6.5.2 What Are the Impacts If I Do Not Pre-configure an ECS Used to Create a Private Image?

Before using an ECS or external image file to create a private image, you need to pre-configure the VM where the ECS or image file is located. If you do not perform the pre-configuration, there will be the following impacts:

1. If you do not set the IP address obtaining mode to DHCP for the ECS NICs or do not delete residual udev rules, ECSs created from the registered private image retain the configuration of the source image file, or the ECS NICs do not start from eth0. In this case, you need to remotely log in to the ECS to configure it.
2. If you do not configure the image used to create a Linux ECS, the following issues may occur during the ECS creation:
 - Customized passwords cannot be injected.
 - Certificates cannot be injected.
 - Other customized configurations cannot be implemented on the ECS.
3. If you do not delete the automatic attaching detection information of user disks from the **fstab** file, the ECSs created from the private image may fail to start.

6.5.3 What Do I Do If I Configure an Incorrect OS or System Disk Size During Private Image Registration Using an Image File?

If you select an incorrect OS, ECSs may fail to be created from the private image. If the configured system disk size is less than the system disk size in the image file, the image will fail to be created.

In such cases, delete the incorrect image and create a private image again using the correct parameter settings.

6.5.4 Why Does the Error Message Displayed on Task Center Indicates That the System Disk Size of the External Image File Exceeds the Maximum System Disk Size When a VHD Image File Failed to Be Uploaded?

If you fail to register an external image file as a private image and the error message displayed on Task Center indicates that the system disk size of the external image file exceeds the maximum system disk size you have configured, possible causes include:

1. The system disk size you have configured is less than the system disk size of the VM on the original platform. Confirm the system disk size of the image file and register it again.
2. The VHD image file is generated using **qemu-img** or similar tools and the virtual size of the VHD image is inconsistent with that of the original VM. For details, see <https://bugs.launchpad.net/qemu/+bug/1490611>.

In this case, run the **qemu-img info** command.

```
[xxxx@xxxxx test]$ qemu-img info 2g.vhd
image: 2g.vhd
file format: vpc
virtual size: 2.0G (2147991552 bytes)
disk size: 8.0K
cluster_size: 2097152
```

Check whether the virtual size value is an integer in GB. As shown in the preceding command output, **2147991552 bytes (2.0004 G)** is larger than **2 G**. Therefore, you need to configure a value larger than 2 GB for the system disk size.

6.6 Image Optimization

6.6.1 Must I Install Guest OS Drivers on an ECS?

Installing Guest OSs driver on an ECS improves your experience in using the ECS. In addition, it also ensures high reliability and stability of ECSs.

- Windows ECSs: Install the PV driver and UVP VMTools on ECSs.
- Linux ECSs: Install xen-pv and VirtIO drivers and load them in initrd.

6.6.2 Why Do I Need to Install and Update VMTools for Windows?

Why Do I Need to Install VMTools?

VMTools is a VirtIO driver (para-virtualization driver) that provides high-performance disks and NICs for ECSs.

- A standard Windows OS does not have the VirtIO driver.
- Public images provided on the platform have VMTools by default.
- You need to install VMTools for private images. For details, see [Installing UVP VMTools](#).

Why Do I Need to Update VMTools?

The platform periodically synchronizes issue-fixed versions from the virtio community and releases updated versions every month. This ensures that known issues found in the community or R&D tests can be avoided on the latest driver.

When Do I Need to Update VMTools?

- If a major error is fixed, you are advised to update VMTools immediately. (This has not happened by now.) If other issues are fixed, choose whether to update VMTools based on your needs.
- The platform updates the VMTools stored in the OBS bucket on a regular basis to ensure that the VMTools you download when creating private images is the latest version.
- Public images on the platform are updated on a regular basis to ensure that the latest version of VMTools is installed.
- The document is updated on a regular basis in accordance with VMTools in the OBS bucket to ensure that the download link of VMTools provided in the document is the latest.

What Operations Do I Need to Perform?

- Update Windows private images or drivers in running Windows ECSs as prompted.
- If you have any technical issue or question, contact the customer service.

6.6.3 What Changes Will Be Made to an Image File Used for Registering a Private Image?

If you enable automatic configuration when registering a private image using an image file, the system will perform the following operations:

Linux

- Check whether drivers related to the PV driver exist. If yes, delete them.
- Modify the **grub** and **syslinux** configuration files to add the OS kernel boot parameters and change the disk partition name (**UUID=UUID of the disk partition**).

- Change the names of the disk partitions in the **/etc/fstab** file (**UUID=UUID of the disk partition**).
- Check whether the **initrd** file has the Xen and IDE drivers. If no, load the Xen and IDE drivers.
- Modify X Window configuration file **/etc/X11/xorg.conf** to prevent display failures.
- Delete services of VMware tools.
- Record the latest automatic modification made to the image into **/var/log/rainbow_modification_record.log**.

 NOTE

For image files in the following scenarios, this function does not take effect after **Enable automatic configuration** is selected:

- Image files whose **/usr** directory is an independent partition
- Fedora 29 64-bit, Fedora 30 64-bit, and CentOS 8.0 64-bit image files that use the XFS file system
- Image files that use SUSE 12 SP4 64bit and the ext4 file system

Windows

- Restore the IDE driver to enable the system to use the IDE driver for its initial start.
- Delete the registry keys of the mouse and keyboard and generate the registry keys on the new platform to ensure that the mouse and keyboard are available.
- Restore the PV driver registry key to rectify driver installation failures and Xen driver conflicts.
- Inject the VirtIO driver offline to solve the problem that the system cannot start when UVP VMTools is not installed.
- Restore DHCP. The system dynamically obtains information such as the IP address based on the DHCP protocol.

6.6.4 What Initial Configuration Needs to Be Performed on the ECS, BMS, or Image File Before It Is Used to Create an Image?

ECS or Image File Configurations

Table 6-2 ECS configurations

OS	Configuration Item	Reference
Windows	• Setting the NIC to DHCP • Enabling remote desktop connection • (Optional) installing special Windows drivers • (Optional) Installing Cloudbase-Init • Installing the Guest OS drivers, including the PV driver and UVP VMTools • Running Sysprep	Creating a System Disk Image from a Windows ECS
Linux	• Setting the NIC to DHCP • (Optional) Installing Cloud-Init • Deleting files in the network rule directory • Changing the disk identifier in the GRUB configuration file to UUID • Changing the disk identifier in the fstab file to UUID • Installing native Xen and KVM drivers • Detaching data disks from the ECS	Creating a System Disk Image from a Linux ECS

Table 6-3 Image file configurations

OS	Configuration Item	Reference
Windows	• Setting the NIC to DHCP • Enabling remote desktop connection • Installing the Guest OS drivers, including the PV driver and UVP VMTools • (Optional) Installing Cloudbase-Init • (Optional) Enabling NIC multi-queue	Preparing an Image File
Linux	• Deleting files in the network rule directory • Setting the NIC to DHCP • Installing native Xen and KVM drivers • Changing the disk identifier in the GRUB configuration file to UUID • Changing the disk identifier in the fstab file to UUID • Deleting the automatic attachment information of non-system disks from the /etc/fstab file • (Optional) Installing Cloud-Init • (Optional) Enabling NIC multi-queue	Preparing an Image File

 NOTE

- When registering an external image file as a private image, you are advised to perform the preceding operations on the VM where the external image file is located.
- When registering a Windows external image file as a private image, if the Guest OS drivers are installed, the cloud platform will check the image file after you select **Enable automatic configuration**. If the GuestOS drivers are not installed, the cloud platform will try to install them.

BMS or Image File Configurations

Table 6-4 BMS configurations

OS	Configuration Item	Reference
Windows	- Installing the bms-network-config software package - Installing Cloudbase-Init - Deleting residual files in the OS	For details, see "Creating a Private Image from a BMS" in <i>Bare Metal Server User Guide</i> .
Linux	- Installing the bms-network-config software package - Installing Cloud-Init - Deleting residual files in the OS	For details, see "Creating a Private Image from a BMS" in <i>Bare Metal Server User Guide</i> .

Table 6-5 Image file configurations

OS	Configuration Item	Reference
Windows	- Installing the V5 server driver - Installing Cloudbase-Init - Installing the bms-network-config software package (Optional) Installing the SDI iNIC driver - Setting the Windows time zone - Setting the virtual memory (Optional) Configuring automatic Windows update - Configuring the SID	<i>Bare Metal Server Image Creation Guide</i>

OS	Configuration Item	Reference
Linux	• Installing and configuring Cloud-Init. • Modifying the hardware device driver that boots the OS • Installing the bms-network-config software package • (Optional) Installing the SDI iNIC driver • (Optional) Installing the Hi1822 NIC driver • (Optional) Installing the IB driver • (Optional) Installing the V5 server driver • (Optional) Installing the UltraPath software • Performing security configuration • Configuring remote login to the BMS • Configuring automatic root partition expansion	<i>Bare Metal Server Image Creation Guide</i>

6.6.5 What Do I Do If the Initial Configurations of a Windows External Image File Are Not Completed Before the File Is Exported?

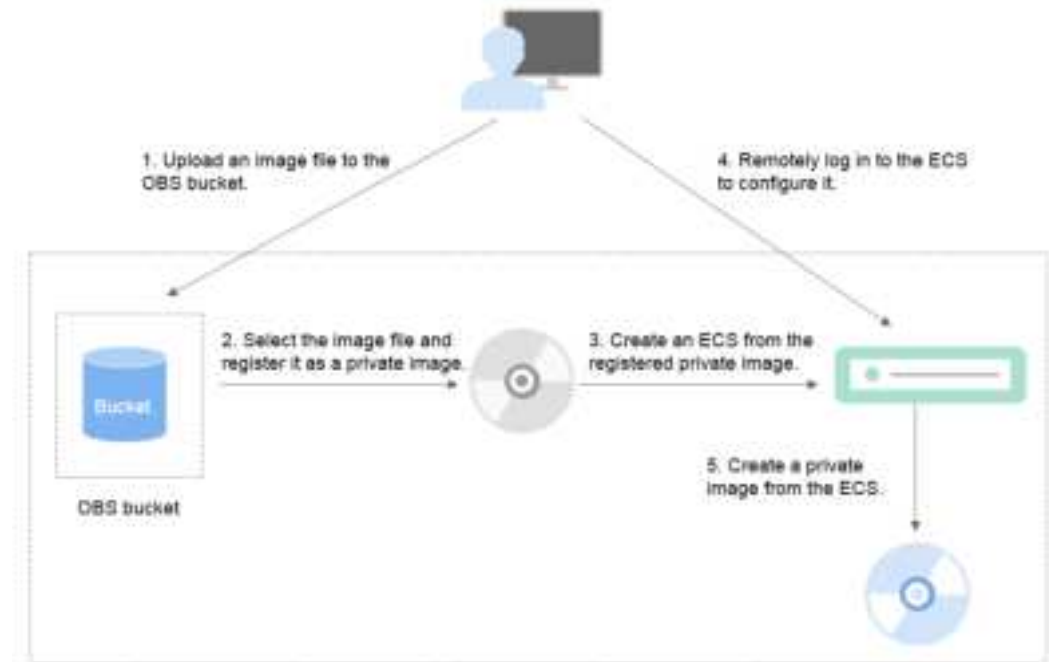
The ECS where the external image file is located is not configured as instructed in [Table 2-5](#) before the image file is exported. You are advised to follow the process in [Figure 6-4](#) to configure the ECS.

CAUTION

The proper running of ECSs depends on the XEN Guest OS driver (PV driver) and KVM Guest OS driver (UVP VMTools). If the drivers are not installed, the performance of ECSs will be affected and some functions will be unavailable. Ensure that the two drivers have been installed for the image file before it is exported from the original platform. Otherwise, the ECSs created from the image will fail to start and cannot be configured.

- Install the PV driver. For details, see [Installing the PV Driver](#).
 - Install UVP VMTools. For details, see [Installing UVP VMTools](#).
-

Figure 6-4 Image creation process



Step 1: Upload the Image File

Upload the external image file to the OBS bucket. For details, see [Uploading an External Image File](#).

Step 2 Register the External Image File as a Private Image

On the management console, select the uploaded image file and register it as an uninitialized private image. For details, see [Registering an External Image File as a Private Image](#).

Step 3: Create an ECS

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. Locate the row that contains the uninitialized private image and click **Apply for Server** in the **Operation** column.
5. Set parameters as promoted to create the ECS. Pay attention to the following:
 - Bind an EIP to the ECS so that you can upload installation packages to the ECS or download installation packages from the ECS.
 - You must add inbound rules for security groups of the ECS to ensure that the ECS can be accessed.
 - If the image file has Cloudbase-Init installed and configured, set a password and log in to the ECS using the password as prompted. If Cloudbase-Init is not installed, use the password or certificate contained in the image file to log in the ECS.

For details, see *Elastic Cloud Server User Guide*.

6. Perform the following steps to check whether the private image is available:
 - a. Check whether the ECS can be successfully started. If the start succeeds, the Guest OS drivers have been installed for the external image file on the original platform or the drivers have been automatically installed for the private image on the cloud platform. If the start failed, install the Guest OS drivers for the image file and register it as a private image again.
 - b. Check whether you can log in to the ECS using your configured password or key. If yes, Cloudbase-Init has been installed. If no, use the password or key contained in the image file to log in to the ECS and install Cloudbase-Init as instructed in [Installing and Configuring Cloudbase-Init](#).
 - c. Check whether the NICs are set to DHCP by referring to **2** in [Step 4: Configure the ECS](#).
 - d. Use MSTSC to log in to the ECS. If the login is successful, remote desktop connection is enabled on the ECS. If the login fails, enable remote desktop connection by referring to **3** in [Step 4: Configure the ECS](#).

If the ECS meets the preceding requirements, the private image is available. You can clear the environment as instructed in [\(Optional\) Clear the Environment](#).

Step 4: Configure the ECS

Remotely log in to the ECS created in [Step 3: Create an ECS](#) to configure the network and install software.

1. Log in to the ECS.
2. Check whether the NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in [Setting the NIC to DHCP](#).
3. Enable remote desktop connection for the ECS as needed. For details about how to enable this function, see [Enabling Remote Desktop Connection](#).
4. (Optional) Configure value-added functions.
 - Install and configure Cloudbase-Init. For details, see [Installing and Configuring Cloudbase-Init](#).
 - Enable NIC multi-queue. For details, see [How Do I Set NIC Multi-Queue for an Image?](#)

Step 5: Create an Image from the ECS

Create a private image from the ECS. For details, see [Creating a System Disk Image from a Windows ECS](#).

(Optional) Clear the Environment

In the preceding steps, the uninitialized image file and created ECS occupy storage and compute resources. Therefore, you are advised to clear the environment after the image is registered.

- Delete the uninitialized image registered in [Step 2 Register the External Image File as a Private Image](#).

- Delete the ECS created in [Step 3: Create an ECS](#).
- Delete the image files stored in the OBS bucket.

6.6.6 What Do I Do If the Initial Configurations of a Linux External Image File Are Not Completed Before the File Is Exported?

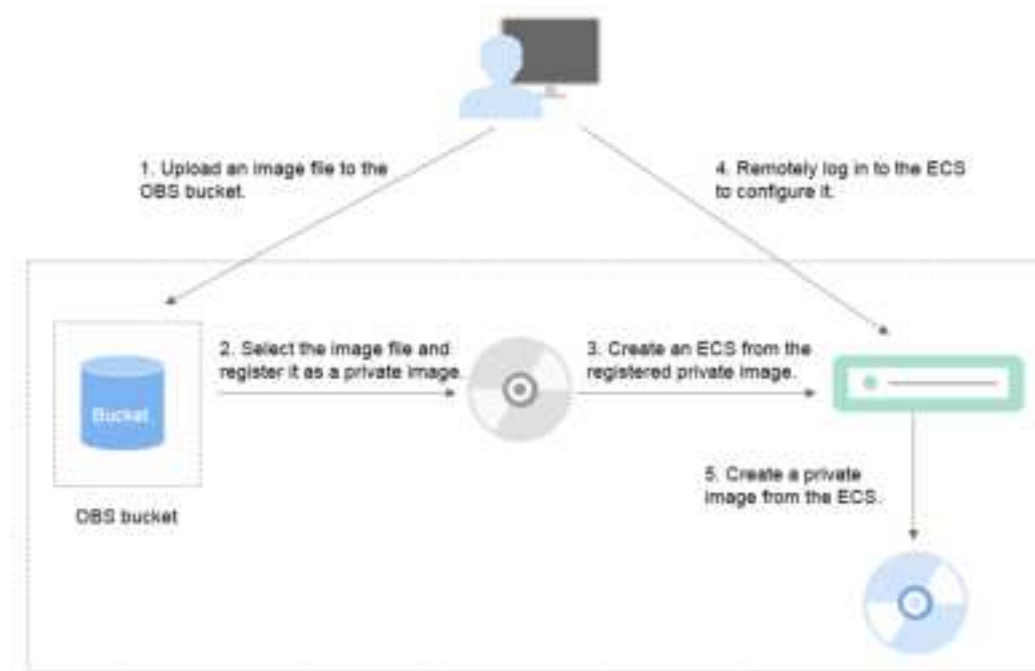
The ECS where the external image file is located is not configured as instructed as in [Table 2-8](#) before the image file is exported. You are advised to follow the process in [Figure 6-5](#) to configure the ECS.

CAUTION

The proper running of ECSs depends on the XEN and KVM drivers. If the drivers are not installed, the performance of ECSs will be affected and some functions will be unavailable. Ensure that the two drivers have been installed for the image file before it is exported from the original platform. Otherwise, the ECSs created from the image will fail to start and cannot be configured.

For details, see [How Do I Install Native Xen and KVM Drivers?](#)

Figure 6-5 Image creation process



Step 1: Upload the Image File

Upload the external image file to the OBS bucket. For details, see [Uploading an External Image File](#).

Step 2 Register the External Image File as a Private Image

On the management console, select the uploaded image file and register it as an uninitialized private image. For details, see [Registering an External Image File as a Private Image](#).

Step 3: Create an ECS

Create an ECS from the uninitialized private image.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. Locate the row that contains the uninitialized private image and click **Apply for Server** in the **Operation** column.
5. Set parameters as promoted to create the ECS. Pay attention to the following:
 - You must add inbound rules for security groups of the ECS to ensure that the ECS can be accessed.
 - If Cloud-Init has been installed in the image file, set a login password as prompted. If Cloud-Init is not installed, use the password or certificate contained in the image file to log in.

For details, see *Elastic Cloud Server User Guide*.

6. Perform the following steps to check whether the private image is available:
 - a. Check whether the ECS can be successfully started. If the start succeeds, the XEN and KVM drivers have been installed for the external image file on the original platform or the drivers have been automatically installed for the private image on the cloud platform. If the start failed, install the XEN and KVM drivers for the image file and register it as a private image again.
 - b. Check whether you can log in to the ECS using your configured password or key. If yes, Cloud-Init has been installed. If no, use the password or key contained in the image file to log in to the ECS and install Cloud-Init as instructed in [Installing Cloud-Init](#).
 - c. Check the network configuration by referring to [Step 4: Configure the ECS](#).

If the ECS meets the preceding requirements, the private image is available. You can clear the environment as instructed in [\(Optional\) Clear the Environment](#).

Step 4: Configure the ECS

Remotely log in to the ECS created in [Step 3: Create an ECS](#) to configure the network and install software.

1. Log in to the ECS.
2. Configuring the network.
 - Run the **ifconfig** command to check whether the private IP address of the ECS is the same as that displayed on the console. If they are

- inconsistent, delete files in the network rule directory as instructed in [Deleting Files in the Network Rule Directory](#).
- Check whether the NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in [Setting the NIC to DHCP](#).
 - Run the **service sshd status** command to check whether SSH is enabled. If it is disabled, run the **service sshd start** command to enable it. Ensure that your firewall (for example, Linux iptables) allows SSH access.
3. Configure a file system.
- Change the disk identifier in the GRUB configuration file to UUID. For details, see [Changing the Disk Identifier in the GRUB Configuration File to UUID](#).
 - Change the disk identifier in the fstab file to UUID. For details, see [Changing the Disk Identifier in the fstab File to UUID](#).
 - Clear the automatic attachment information of non-system disks in the **/etc/fstab** file to prevent impacts on subsequent data disk attachment. For details, see [Detaching Data Disks from an ECS](#).
4. (Optional) Configure value-added functions.
- Install and configure Cloud-Init. For details, see [Installing Cloud-Init](#) and [Configuring Cloud-Init](#).
 - Enable NIC multi-queue. For details, see [How Do I Set NIC Multi-Queue for an Image?](#)

Step 5: Create an Image from the ECS

Create a private image from the ECS. For details, see [Creating a System Disk Image from a Linux ECS](#).

(Optional) Clear the Environment

In the preceding steps, the uninitialized image file and created ECS occupy storage and compute resources. Therefore, you are advised to clear the environment after the image is registered.

- Delete the uninitialized image registered in [Step 2 Register the External Image File as a Private Image](#).
- Delete the ECS created in [Step 3: Create an ECS](#).
- Delete the image files stored in the OBS bucket.

6.6.7 How Do I Set NIC Multi-Queue for an Image?

Scenarios

With the increase of network I/O bandwidth, a single vCPU cannot meet the requirement of processing NIC interruptions. NIC multi-queue enables multiple vCPUs to process NIC interruptions, thereby improving network PPS and I/O performance.

ECSs Supporting NIC Multi-Queue

NIC multi-queue can be enabled on an ECS only when the ECS specifications, virtualization type, and image OS meet the requirements described in this section.

- For details about the ECS flavors that support NIC multi-queue, see section "Instances" in *Elastic Cloud Server User Guide*.

NOTE

If the number of NIC queues is greater than 1, NIC multi-queue is supported.

- Only KVM ECSs support NIC multi-queue.
- The Linux public images listed in [Table 6-6](#) support NIC multi-queue.

NOTE

- Windows public images have not supported NIC multi-queue. If you enable NIC multi-queue in a Windows public image, starting an ECS created using such an image may be slow.
- You are advised to upgrade the kernel version of the Linux ECS to 2.6.35 or later. Otherwise, NIC multi-queue is not supported.

Run the **uname -r** command to obtain the kernel version. If the kernel version is earlier than 2.6.35, contact technical support to upgrade the kernel.

Table 6-6 KVM ECSs that support NIC multi-queue

OS	Image	Status
Windows	Windows Server 2008 WEB R2 64bit	Supported using private images
	Windows Server 2008 R2 Standard/Datacenter/Enterprise 64bit	Supported using private images
	Windows Server 2012 R2 Standard/Datacenter 64bit	Supported using private images
	Windows Server 2016 Standard/Datacenter 64bit	Supported using private images
Linux	Ubuntu 14.04/16.04 Server 64bit	Supported
	openSUSE 42.2 64bit	Supported
	SUSE Enterprise 12 SP1/SP2 64bit	Supported
	CentOS 6.8/6.9/7.0/7.1/7.2/7.3/7.4/7.5/7.6 64bit	Supported
	Debian 8.0.0/8.8.0/8.9.0/9.0.0 64bit	Supported
	Fedora 24/25 64bit	Supported
	EulerOS 2.2 64bit	Supported

Operation Instructions

The ECS described in the following section is assumed to meet the requirements on specifications and virtualization type.

- If the ECS was created using a public image listed in [ECSs Supporting NIC Multi-Queue](#), NIC multi-queue has been enabled on the ECS by default. Therefore, you do not need to perform the operations described in this section.
- If the ECS was created using a private image and the external image file is listed in [ECSs Supporting NIC Multi-Queue](#), perform the following operations to enable NIC multi-queue:
 - a. [Import an External Image File to the IMS Console](#)
 - b. [Set NIC Multi-Queue for the Image](#)
 - c. [Create an ECS from the Private Image](#)
 - d. [Enable NIC Multi-Queue](#)

Import an External Image File to the IMS Console

Import an external image file to the IMS console. For details, see [Registering an External Image File as a Private Image](#).

Set NIC Multi-Queue for the Image

Windows OSs have not commercially supported NIC multi-queue. If you enable NIC multi-queue in a Windows image, starting an ECS created using such an image may be slow.

Use either of the following methods to set the NIC multi-queue attribute.

Method 1:

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. On the displayed **Private Images** page, locate the row that contains the target image and click **Modify** in the **Operation** column.
4. Set the NIC multi-queue attribute of the image.

Method 2:

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. On the displayed **Private Images** page, click the name of the target image.
4. In the upper right corner of the displayed image details page, click **Modify**. In the displayed **Modify Image** dialog box, set parameter **NIC Multi-Queue**.

Method 3: Add `hw_vif_multiqueue_enabled` to an image through the API.

1. For details about how to obtain the token, see **Calling APIs > Authentication** in *Image Management Service API Reference*.

2. For details about how to call an API to update image information, see "Updating Image Information (Native OpenStack API)" in *Image Management Service API Reference*.
3. Add **X-Auth-Token** to the request header.
The value of **X-Auth-Token** is the token obtained in step 1.
4. Add **Content-Type** to the request header.
The value of **Content-Type** is **application/openstack-images-v2.1-json-patch**.

The request URI is in the following format:

PATCH /v2/images/{image_id}

The request body is as follows:

```
[
  {
    "op": "add",
    "path": "/hw_vif_multiqueue_enabled",
    "value": "true"
  }
]
```

Figure 6-6 shows an example request body for setting the NIC multi-queue attribute.

Figure 6-6 Example request body



Create an ECS from the Private Image

Use the registered private image to create an ECS. For details, see *Elastic Cloud Server User Guide*. Note the following when setting the parameters:

- **Region:** Select the region where the private image is located.
- **Image:** Select **Private image** and then the desired image from the drop-down list.

Enable NIC Multi-Queue

KVM ECSs running Windows use private images to support NIC multi-queue.

For Linux ECSs, which run CentOS 7.4 as an example, perform the following operations to enable NIC multi-queue:

Step 1 Enable NIC multi-queue.

1. Log in to the ECS.
2. Run the following command to obtain the number of queues supported by the NIC and the number of queues with NIC multi-queue enabled:
ethtool -l NIC
3. Run the following command to configure the number of queues used by the NIC:
ethtool -L NIC combined *Number of queues*

Example:

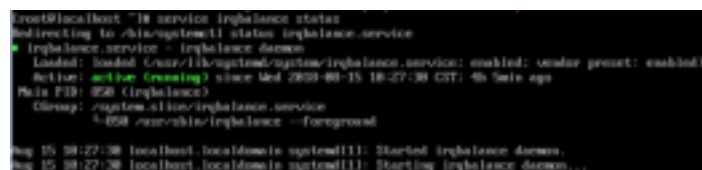
```
[root@localhost ~]# ethtool -l eth0 #View the number of queues used by NIC eth0.
Channel parameters for eth0:
Pre-set maximums:
RX:          0
TX:          0
Other:       0
Combined: 4 #Indicates that a maximum of four queues can be enabled for the NIC.
Current hardware settings:
RX:          0
TX:          0
Other:       0
Combined: 1 #Indicates that one queue has been enabled.

[root@localhost ~]# ethtool -L eth0 combined 4 #Enable four queues on NIC eth0.
```

Step 2 (Optional) Enable irqbalance so that the system automatically allocates NIC interruptions to multiple vCPUs.

1. Run the following command to enable irqbalance:
service irqbalance start
2. Run the following command to view the irqbalance status:
service irqbalance status
If the **Active** value in the command output contains **active (running)**, irqbalance has been enabled.

Figure 6-7 Enabled irqbalance



```
root@localhost ~# service irqbalance status
Redirecting to /sbin/systemctl status irqbalance.service
* irqbalance.service - irqbalance daemon
   loaded: loaded (/usr/lib/systemd/system/irqbalance.service; enabled; vendor preset: enabled)
   active: active (running) since Wed 2020-09-15 18:27:38 CST; 4s left ago
   Main PID: 858 (irqbalance)
   CGroup: /system.slice/irqbalance.service
           └─858 /usr/sbin/irqbalance --foreground

Aug 15 08:27:38 localhost.localdomain systemd[1]: Started irqbalance daemon.
Aug 15 08:27:38 localhost.localdomain systemd[1]: Starting irqbalance daemon...
```

Step 3 (Optional) Enable interrupt binding.

Enabling irqbalance allows the system to automatically allocate NIC interruptions, improving network performance. If the improved network performance fails to meet your expectations, manually configure interrupt affinity on the target ECS.

The detailed operations are as follows:

Run the following script so that each ECS vCPU responds the interrupt requests initialized by one queue. That is, one queue corresponds to one interrupt, and one interrupt binds to one vCPU.

```
#!/bin/bash
service irqbalance stop

eth_dirs=$(ls -d /sys/class/net/eth*)
```

```
if [ $? -ne 0 ];then
    echo "Failed to find eth* , sleep 30" >> $ecs_network_log
    sleep 30
    eth_dirs=$(ls -d /sys/class/net/eth*)
fi

for eth in $eth_dirs
do
    cur_eth=$(basename $eth)
    cpu_count=`cat /proc/cpuinfo| grep "processor"| wc -l`
    virtio_name=$(ls -l /sys/class/net/"$cur_eth"/device/driver/ | grep pci |awk '{print $9}')

    affinity_cpu=0
    virtio_input="$virtio_name"-input
    irqs_in=$(grep "$virtio_input" /proc/interrupts | awk -F ":" '{print $1}')
    for irq in ${irqs_in[*]}
    do
        echo $((affinity_cpu%cpu_count)) > /proc/irq/"$irq"/smp_affinity_list
        affinity_cpu=$((affinity_cpu+2))
    done

    affinity_cpu=1
    virtio_output="$virtio_name"-output
    irqs_out=$(grep "$virtio_output" /proc/interrupts | awk -F ":" '{print $1}')
    for irq in ${irqs_out[*]}
    do
        echo $((affinity_cpu%cpu_count)) > /proc/irq/"$irq"/smp_affinity_list
        affinity_cpu=$((affinity_cpu+2))
    done
done
```

Step 4 (Optional) Enable XPS and RPS.

XPS allows the system with NIC multi-queue enabled to select a queue by vCPU when sending a data packet.

```
#!/bin/bash
# enable XPS feature
cpu_count=$(grep -c processor /proc/cpuinfo)
dec2hex(){
    echo $(printf "%0x" $1)
}
eth_dirs=$(ls -d /sys/class/net/eth*)
if [ $? -ne 0 ];then
    echo "Failed to find eth* , sleep 30" >> $ecs_network_log
    sleep 30
    eth_dirs=$(ls -d /sys/class/net/eth*)
fi
for eth in $eth_dirs
do
    cpu_id=1
    cur_eth=$(basename $eth)
    cur_q_num=$(ethtool -l $cur_eth | grep -iA5 current | grep -i combined | awk '{print $2}')
    for ((i=0;i<cur_q_num;i++))
    do
        if [ $i -eq $cpu_count ];then
            cpu_id=1
        fi
        xps_file="/sys/class/net/${cur_eth}/queues/tx-$i/xps_cpus"
        rps_file="/sys/class/net/${cur_eth}/queues/rx-$i/rps_cpus"
        cpuset=$(dec2hex "$cpu_id")
        echo $cpuset > $xps_file
        echo $cpuset > $rps_file
        let cpu_id=cpu_id*2
    done
done
```

----End

6.6.8 How Do I Optimize a System Disk Image So That It Can Be Used to Create ECSs Quickly?

Scenarios

If a system disk image supports fast ECS creation, the time required for creating ECSs from it can be greatly reduced. Existing system disk images may not support fast ECS creation. You are advised to optimize the images using the image replication function.

If image A cannot be used to quickly create ECSs, you can replicate it to generate image copy_A, which can be used to quickly create ECSs.

Constraints

Full-ECS images cannot be optimized using this method.

Check Whether an Image Supports Fast ECS Creation

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
3. Click the **Private Images** tab to display the image list.
4. Click the name of the target image.
5. On the displayed image details page, check the value of **Fast ECS Creation**.

Optimize an Image

1. Locate the target system disk image, click **More** in the **Operation** column, and select **Replicate** from the drop-down list.
The **Replicate Image** dialog box is displayed.
2. Set parameters based on [Replicating Images](#).
3. After the image is successfully replicated, the generated image can be used to quickly create ECSs.

6.6.9 What Is the Cause of the Failure to Install a Guest OS Driver on a Windows ECS?

Any of the following may cause the Guest OS driver installation to fail:

- Your image file was exported from a VMware VM, and VMware Tools was not uninstalled or not completely uninstalled.
- You have downloaded the Guest OS driver of an incorrect version for your Windows ECS.
- The disk space available for installing the Guest OS driver is insufficient. Therefore, you must ensure that the disk where the Guest OS driver is installed has at least 300 MB available space.

6.6.10 How Do I Install Native Xen and KVM Drivers?

Scenarios

When optimizing a Linux private image, you need to install native Xen and KVM drivers for the image.

CAUTION

If you do not install Xen drivers for the image, the network performance of the ECSs created from this image will be poor, and the security groups and firewall configured for the ECSs will not take effect.

If you do not install KVM drivers for the image, the NICs of the ECSs may not be detected and the ECSs will be unable to communicate with other resources. Therefore, you must install Xen and KVM drivers.

This section describes how to install native Xen and KVM drivers.

Prerequisites

- ECSs that use native Linux Xen and KVM drivers must have a kernel later than the 2.6.24 version.
- Disable your antivirus and intrusion detection software. You can enable the software after Xen and KVM drivers are installed.

Procedure

Edit the configuration file based on the OS version.

- CentOS, EulerOS

Take CentOS 7.0 as an example. Modify the `/etc/dracut.conf` file. Add the xen-pv and VirtIO drivers to **add_drivers**. xen-pv drivers include xen-blkfront and xen-netfront. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the `/etc/dracut.conf` file. Run the **dracut -f** command to regenerate initrd.

For details, see [CentOS and EulerOS](#).

- Ubuntu and Debian

Modify the `/etc/initramfs-tools/modules` file. Add the xen-pv and VirtIO drivers. xen-pv drivers include xen-blkfront and xen-netfront. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the `/etc/initramfs-tools/modules` file. Run the **update-initramfs -u** command to regenerate initrd.

For details, see [Ubuntu and Debian](#).

- SUSE and openSUSE

- If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the `/etc/sysconfig/kernel` file and add xen-pv and VirtIO drivers to **INITRD_MODULES=""**. xen-pv drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net,

virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **mkinitrd** command to regenerate initrd.

- If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file and add xen-pv and VirtIO drivers to **add_drivers**. xen-pv drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **dracut -f** command to regenerate initrd.
- If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file and add xen-pv and VirtIO drivers to **add_drivers**. xen-pv drivers include xen-blkfront and xen-netfront. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

For details, see [SUSE and openSUSE](#).

NOTE

For SUSE, run the following command to check whether xen-kmp (driver package for xen-pv) is installed:

```
rpm -qa |grep xen-kmp
```

The following information is displayed:

```
xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
```

If xen-kmp is not installed, obtain it from the ISO file and install it first.

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected.

CentOS and EulerOS

1. Run the following command to open the **/etc/dracut.conf** file:
vi /etc/dracut.conf
2. Press **i** to enter editing mode and add the xen-pv and VirtIO drivers to **add_drivers** (the format depends on the OS requirements).
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
additional kernel modules to the default
add_drivers+="xen-blkfront xen-netfront virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
.....
3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
4. Run the following command to regenerate initrd:
dracut -f /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img
If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.
5. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been loaded:
lsinitrd /boot/initramfs-`uname -r`.img | grep xen
lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is `initrd`, run the following commands to check whether native Xen and KVM drivers have been loaded:

`lsinitrd /boot/initrd-`uname -r` | grep xen`

`lsinitrd /boot/initrd-`uname -r` | grep virtio`

Assume that the virtual file system is `initramfs`. The following command output will be displayed:

```
[root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep xen
-rwxr--r-- 1 root root 54888 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
block/xen-blkfront.ko
-rwxr--r-- 1 root root 45664 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/
drivers/net/xen-netfront.ko

[root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
-rwxr--r-- 1 root root 23448 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
block/virtio_blk.ko
-rwxr--r-- 1 root root 50704 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/
drivers/net/virtio_net.ko
-rwxr--r-- 1 root root 28424 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
scsi/virtio_scsi.ko
drwxr-xr-x 2 root root 0 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio
-rwxr--r-- 1 root root 14544 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio.ko
-rwxr--r-- 1 root root 21040 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_pci.ko
-rwxr--r-- 1 root root 18016 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_ring.ko
```

NOTE

If you add built-in drivers to the `initrd` or `initramfs` file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the `lsinitrd` command. You can run the following commands to check whether the drivers are built-in ones in the kernel:

`cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y`

`cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y`

Ubuntu and Debian

1. Run the following command to open the **modules** file:
`vi /etc/initramfs-tools/modules`
2. Press **i** to enter editing mode and add the `xen-pv` and `VirtIO` drivers to the `/etc/initramfs-tools/modules` file (the format depends on the OS requirements).

```
[root@CTU10000xxxxx ~]# vi /etc/initramfs-tools/modules
.....
# Examples:
#
# raid1
# sd_mOd
xen-blkfront
xen-netfront
virtio_blk
virtio_scsi
virtio_net
virtio_pci
virtio_ring
virtio
```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the `/etc/initramfs-tools/modules` file.

4. Run the following command to regenerate initrd:

update-initramfs -u

5. Run the following commands to check whether native Xen and KVM drivers have been installed:

lsinitramfs /boot/initrd.img-`uname -r` |grep xen

lsinitramfs /boot/initrd.img-`uname -r` |grep virtio

```
[root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep xen
lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen
lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen/netxen_nic.ko
lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback
lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback/xen-netback.ko
lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback
lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback/xen-blkback.ko
```

```
[root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
lib/modules/3.5.0-23-generic/kernel/drivers/scsi/virtio_scsi.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether the drivers are built-in ones in the kernel:

```
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
CONFIG_VIRTIO_BLK=y
CONFIG_VIRTIO_NET=y
CONFIG_VIRTIO=y
CONFIG_VIRTIO_RING=y
CONFIG_VIRTIO_PCI=y
CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y
CONFIG_XEN_BLKDEV_FRONTEND=y
CONFIG_XEN_NETDEV_FRONTEND=y
```

SUSE and openSUSE

If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the **/etc/sysconfig/kernel** file. For details, see [scenario 1](#).

If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file and add xen-pv and VirtIO drivers. For details, see [scenario 2](#).

If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file and add xen-pv and VirtIO drivers to **add_drivers**. For details, see [scenario 3](#).

- If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, perform the following steps:

NOTE

For SUSE, run the following command to check whether xen-kmp (driver package for xen-pv) is installed:

rpm -qa |grep xen-kmp

The following information is displayed:

```
xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
```

If xen-kmp is not installed, obtain it from the installation ISO and install it first.

- a. Run the following command to modify the **/etc/sysconfig/kernel** file:

vi etc/sysconfig/kernel

- b. Add the xen-pv and VirtIO drivers after **INITRD_MODULES=** (the format of drivers depends on the OS).

```
SIA10000xxxx:~ # vi /etc/sysconfig/kernel
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk
virtio_scsi virtio_net virtio_pci virtio_ring virtio"
```

- c. Run the **mkinitrd** command to regenerate initrd:

 NOTE

If the virtual file system is not the default initramfs or initrd, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **menu.lst** or **grub.cfg** file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, or **/boot/grub2/grub.cfg**).

The following is an example initrd file of SUSE 11 SP4:

```
default 0
timeout 10
gfxmenu (hd0,0)/boot/message
title sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent console=ttyS0,115200n8 console=tty0
net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
initrd /boot/initrd.vmx
title Failsafe_sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent ide=nodma apm=off noresume edd=off
powersaved=off nohz=off highres=off processor.max+cstate=1 nomodeset x11failsafe
console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1
showopts
initrd /boot/initrd.vmx
```

/boot/initrd.vmx in the **initrd** line is the **initrd** file actually used. Run the **dracut -f /boot/initrd.vmx** command. If the **initrd** file does not contain the **/boot** directory, such as **/initramfs-xxx**, run the **dracut -f /boot/initramfs-xxx** command.

- d. Run the following commands to check whether the PVOPS module for Xen or VirtIO module for KVM is loaded:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
xen-netfront.ko
```

```
SIA10000xxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

- e. Restart the ECS.

- f. Modify the **/boot/grub/menu.lst** file. Add **xen_platform_pci.dev_unplug=all** and modify the root configuration.

Before the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314
initrd /boot/initrd-3.0.76-0.11-default
```

After the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314
xen_platform_pci.dev_unplug=all
initrd /boot/initrd-3.0.76-0.11-default
```

NOTE

- Ensure that the root partition is in the UUID format.
 - **xen_platform_pci.dev_unplug=all** is added to shield the QEMU device.
 - For SUSE 11 SP1 64-bit to SUSE 11 SP4 64-bit, add **xen_platform_pci.dev_unplug=all** to the **menu.lst** file. For SUSE 12 or later, this function is enabled by default, and you do not need to configure it.
- g. Run the following commands to check whether the Xen drivers exist in initrd:

```
lsinitrd /boot/initrd-`uname -r` | grep xen
```

```
lsinitrd /boot/initrd-`uname -r` | grep virtio
```

```
SIA10000xxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
xen-netfront.ko
```

```
SIA10000xxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether the drivers are built-in ones in the kernel:

```
cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
```

```
cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y
```

- If the OS version is SUSE 12 SP1, perform the following steps:

- a. Run the following command to open the **/etc/dracut.conf** file:
vi /etc/dracut.conf
- b. Press **i** to enter editing mode and add the xen-pv and VirtIO drivers to **add_drivers** (the format depends on the OS requirements).

```
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk virtio_scsi
virtio_net virtio_pci virtio_ring virtio"
```
- c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
- d. Run the following command to regenerate initrd:
dracut -f /boot/initramfs-File name
If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.
- e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been loaded:
lsinitrd /boot/initramfs-`uname -r`.img | grep xen
lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
If the virtual file system is initrd, run the following commands to check whether native Xen and KVM drivers have been loaded:
lsinitrd /boot/initrd-`uname -r` | grep xen
lsinitrd /boot/initrd-`uname -r` | grep virtio
- If the OS version is later than SUSE 12 SP1 or openSUSE 13, perform the following steps:
Take SUSE Linux Enterprise Server 12 SP2 (x86_64) as an example.
 - a. Run the following command to open the **/etc/dracut.conf** file:
vi /etc/dracut.conf
 - b. Press **i** to enter editing mode and add the xen-pv and VirtIO drivers to **add_drivers** (the format depends on the OS requirements).

```
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic xen-blkfront xen-netfront virtio_blk virtio_scsi
virtio_net virtio_pci virtio_ring virtio"
```
 - c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
 - d. Run the following command to regenerate initrd:
dracut -f /boot/initramfs-File name
If the virtual file system is not the default initramfs, run the **dracut -f Name of the initramfs or initrd file actually used** command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.
 - e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been loaded:
lsinitrd /boot/initramfs-`uname -r`.img | grep xen

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is `initrd`, run the following commands to check whether the native Xen and KVM driver modules are successfully loaded:

lsinitrd /boot/initrd-`uname -r` | grep xen**lsinitrd /boot/initrd-`uname -r` | grep virtio**

Assume that the virtual file system is `initrd`. The following command output will be displayed:

```
sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rw-r--r-- 1 root root 69575 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/xen-
blkfront.ko
-rw-r--r-- 1 root root 53415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/xen-
netfront.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-hcall
-rwxr-xr-x 1 root root 8320 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-
hcall/xen-hcall.ko

sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rw-r--r-- 1 root root 29335 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/
virtio_blk.ko
-rw-r--r-- 1 root root 57007 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/
virtio_net.ko
-rw-r--r-- 1 root root 32415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/scsi/
virtio_scsi.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/kernel/drivers/virtio
-rw-r--r-- 1 root root 19623 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio.ko
-rw-r--r-- 1 root root 38943 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_pci.ko
-rw-r--r-- 1 root root 24431 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_ring.ko
```

 NOTE

If you add built-in drivers to the `initrd` or `initramfs` file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the `lsinitrd` command. You can run the following commands to check whether the drivers are built-in ones in the kernel:

```
cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y
```

6.7 Image Encryption

How Can I Change an Unencrypted Image to an Encrypted One?

If you want to store an unencrypted image in an encrypted way, you can select an encryption key and replicate the image to generate an encrypted image.

6.8 Cloud-Init

6.8.1 What Can I Do with a Cloud-Init ECS?

Introduction to Cloud-Init

Cloud-Init is an open-source cloud initialization tool. When creating ECSs from an image with Cloud-Init, you can use user data injection to inject customized

initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloud-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs.

Installation Methods

You are advised to install Cloud-Init or Cloudbase-Init on the ECS to be used to create a private image so that new ECSs created from the private image support custom configurations.

- For Windows OSs, download and install Cloudbase-Init.
For how to install Cloudbase-Init, see [Installing and Configuring Cloudbase-Init](#).
- For Linux OSs, download and install Cloud-Init.
For how to install Cloud-Init, see [Installing Cloud-Init](#).
For how to configure Cloud-Init, see [Configuring Cloud-Init](#).

6.8.2 What Do I Do If Injecting the Key or Password Using Cloud-Init Failed After NetworkManager Is Installed?

Symptom

A major cause is that the version of Cloud-Init is incompatible with that of NetworkManager. In Debian 9.0 and later versions, NetworkManager is incompatible with Cloud-Init 0.7.9.

Solution

Uninstall the current Cloud-Init and install Cloud-Init 0.7.6 or an earlier version.

For details about how to install Cloud-Init, see [Installing Cloud-Init](#).

6.8.3 How Do I Install growpart for SUSE 11 SP4?

Scenarios

growpart for SUSE and openSUSE is an independent toolkit that does not start with **cloud-***. Perform operations in this section to install growpart:

Procedure

1. Run the following commands to check whether Cloud-Init and growpart have been installed:

```
rpm -qa | grep cloud-init
```

The command output is as follows:

```
cloud-init-0.7.8-39.2
```

```
rpm -qa | grep growpart
```

The command output is as follows:

```
growpart-0.29-8.1
```

2. Run the following command to uninstall Cloud-Init and growpart:
zypper remove cloud-init growpart
3. Run the following commands to clear residual files:
rm -fr /etc/cloud/*
rm -fr /var/lib/cloud/*
4. Run the following command to install growpart:
**zypper install http://download.opensuse.org/repositories/home:/garloff:/
OTC:/cloudinit/SLE_11_SP4/noarch/growpart-0.27-1.1.noarch.rpm**
5. Run the following command to install python-oauth:
**zypper install http://download.opensuse.org/repositories/home:/garloff:/
OTC:/cloudinit/SLE_11_SP4/x86_64/python-oauth-1.0.1-35.1.x86_64.rpm**
6. Run the following command to install Cloud-Init:
**zypper install http://download.opensuse.org/repositories/home:/garloff:/
OTC:/cloudinit/SLE_11_SP4/x86_64/cloud-init-0.7.6-27.23.1.x86_64.rpm**
7. Run the following commands to check whether growpart, python-oauth, and Cloud-Init have been installed successfully:
rpm -qa | grep growpart
The command output is as follows:
growpart-0.27-1.1
rpm -qa | grep python-oauth
The command output is as follows:
python-oauthlib-0.6.0-1.5
python-oauth-1.0.1-35.1
rpm -qa | grep cloud-init
The command output is as follows:
cloud-init-0.7.6-27.19.1
8. Run the following command to check the configuration:
**chkconfig cloud-init-local on;chkconfig cloud-init on;chkconfig cloud-
config on;chkconfig cloud-final on**

6.8.4 How Do I Configure a Linux Private Image That Can Automatically Expand Its Root Partition?

Constraints

- An image whose root partition file system is xfs cannot automatically expand its partitions.
- An image that has the LVM partition cannot automatically expand its partitions.
- Images whose file system is ext3 or ext4 are recommended.

NOTE

After OS partitions of old versions are expanded, the OS must be restarted to update the file system.

Installation of growpart on Different OSs

To enable private images to automatically expand the root partition, install growpart.

Table 6-7 growpart installation packages for different OSs

OS	Tool Package
Debian/Ubuntu	cloud-init, cloud-utils, and cloud-initramfs-growroot
Fedora/CentOS	cloud-init, cloud-utils, and cloud-utils-growpart
SUSE/openSUSE	cloud-init and growpart

NOTE

For Debian 9, use method 1 to install growpart. If the installation fails, use method 2 to install growpart.

Method 1:

Run the following command to install growpart:

```
apt-get install -y -f cloud-init cloud-utils cloud-initramfs-growroot
```

Method 2:

If method 1 fails, it may be because the installation source of Debian 9.0.0 is faulty. Therefore, you need to download dependent packages **cloud-utils** and **cloud-initramfs-growroot** and install them.

1. Run the following command to download the dependent packages:

```
wget Package download path
```

You can obtain the dependent packages from the following paths:

http://ftp.br.debian.org/debian/pool/main/c/cloud-utils/cloud-utils_0.29-1_all.deb

http://ftp.br.debian.org/debian/pool/main/c/cloud-initramfs-tools/cloud-initramfs-growroot_0.18.debian5_all.deb

2. Run the following command to rectify the dependent packages:

```
apt --fix-broken install
```

3. Run the following command to install the dependent packages:

```
dpkg -i cloud-utils package path cloud-initramfs-growroot package path
```

An example command is **dpkg -i /root/cloud-utils_0.29-1_all.deb /root/cloud-initramfs-growroot_0.18.debian5_all.deb**.

For other Debian versions, run the following command to install dependent packages:

```
apt-get update;apt-get install cloud-utils cloud-initramfs-growroot
```

Procedure

Take the following as two examples of image disk partitioning:

If the root partition is the last partition, see [Root partition at the last](#).

If the root partition is not the last partition, see [Root partition not at the last](#).

 NOTE

If the **parted** command fails, ensure that the **parted** tool has been installed on the OS. Perform the following operations to install the tool:

- For CentOS, run the following command:
yum install parted
- For Debian, run the following command:
apt-get install parted
- Root partition at the last (**/dev/xvda1: swap** and **/dev/xvda2: root**)
For example, if the system disk size of CentOS 6.5 64-bit is 40 GB, perform the following operations to configure a Linux private image that can automatically expand its root partition:

- a. Run the following command to query the partitions of **/dev/xvda**:

parted -l /dev/xvda

As shown in the command output, the root partition is the second partition and is 38.7 GB.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
```

```
Number Start End Size Type File system Flags
1 1049kB 4296MB 4295MB primary linux-swaps(v1)
2 4296MB 42.9GB 38.7GB primary ext4 boot
```

- b. Install growpart to ensure that the image can automatically expand its root partition.

Run the following command to install growpart:

yum install cloud-*

 NOTE

growpart may be contained in the **cloud-utils-growpart/cloud-utils/cloud-initramfs-tools/cloud-init** package. You can run the preceding command directly and then run the **growpart** command to check whether growpart has been installed successfully.

- c. Run the following command to obtain the file system type and UUID:

blkid

The command output is as follows:

```
/dev/xvda1: UUID="25ec3bdb-ba24-4561-bcdc-802edf42b85f" TYPE="swap"
/dev/xvda2: UUID="1a1ce4de-e56a-4e1f-864d-31b7d9dfb547" TYPE="ext4"
```

- d. Stop the ECS and use it to create a private image.

```
[root@sluo-ecs-e6dc-resizefs ~]# poweroff
Connection closed by foreign host.
Disconnected from remote host at 11:08:54.
Type 'help' to learn how to use Xshell prompt.
```

- e. Use the created image to provision an ECS with a 50 GB system disk. Log in to the ECS and run the following command to query the expanded partitions:

parted -l /dev/xvda

As shown in the command output, the root partition has been expanded automatically.

```

Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

NumberStartEndSizeTypeFile systemFlags
1 1049kB 4296MB 4295MB primary linux-swaps(v1)
2 4296MB 53.7GB 49.4GB primary ext4 boot

```

- f. Run the following command to check whether disks are attached to the ECS successfully:

df -Th

The command output is as follows:

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/xvda2	ext4	49.4G	2.6G	46.8G	4%	/dev/shm
tmpfs	tmpfs	4295M	0	4295M	0%	/

- Root partition not at the last (for example, **/dev/xvda1: root** and **/dev/xvda2: swap**)

For example, if the system disk size of CentOS 7.3 64-bit is 40 GB, perform the following operations to configure a Linux private image that can automatically expand its root partition:

- a. Run the following command to query the partitions of **/dev/xvda**:

parted -l /dev/xvda

As shown in the command output, the root partition is the first partition and is 40.9 GB. The swap partition is the second partition.

```

Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	41.0GB	40.9GB	primary	ext4	boot
2	41.0GB	42.9GB	2000MB	primary	linux-swaps(v1)	

- b. Run the following command to check the configuration of the **/etc/fstab** file:

tail -n 3 /etc/fstab

As shown in the command output, UUIDs of the two partitions are displayed.

```

#
UUID=7c4fce5d-f8f7-4ed6-8463-f2bd22d0ddea /          ext4 defaults 1 1
UUID=5de3cf2c-30c6-4fb2-9e63-830439d4e674 swap      swap defaults 0 0

```

- c. Run the following command to open the **/etc/fstab** file and press **i** to enter editing mode:

vi /etc/fstab

- d. Delete the swap partition configuration, press **Esc** to exit editing mode, and run the following command to save the configuration:

wq!

- e. Run the following command to check whether the configuration has been modified:

tail -n 3 /etc/fstab

As shown in the command output, only the UUID of the root partition is displayed.

```
UUID=7c4fce5d-f8f7-4ed6-8463-f2bd22d0ddea / ext4 defaults 1 1
```

- f. Run the following command to stop the swap device:

```
swapoff -a
```

- g. Run the following command to query the partitions of **/dev/xvda**:

```
parted /dev/xvda
```

The command output is as follows:

```
[root@test-0912 bin]# parted /dev/xvda
GNU Parted 3.1
Using /dev/xvda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

- h. Run the following command to query the disk partitions:

```
p
```

The command output is as follows:

```
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
 1 1049kB 4296MB 4295MB primary linux-swap(v1)
 2 4296MB 42.9GB 38.7GB primary xfs boot
(parted)
```

- i. Run the following command to delete the second partition:

```
rm 2
```

The command output is as follows:

```
(parted) rm 2
(parted)
```

- j. Run the following command to query the disk partitions:

```
p
```

The command output is as follows:

```
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
 1 1049kB 41.0GB 40.9GB primary ext4 boot
```

- k. Enter **quit**.

- l. Run the following command to query the partitions of **/dev/xvda**:

```
parted -l /dev/xvda
```

As shown in the command output, the swap partition is deleted.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
 1 1049kB 41.0GB 40.9GB primary ext4 boot
```

- m. Install growpart to ensure that the image can automatically expand its root partition.

Run the following command to install growpart:

yum install cloud-*

 NOTE

growpart may be contained in the **cloud-utils-growpart/cloud-utils/cloud-initramfs-tools/cloud-init** package. You can run the preceding command directly and then run the **growpart** command to check whether growpart has been installed successfully.

- n. Run the following command to expand the swap partition of the **/dev/xvda** disk to the first partition to which the root partition belongs:

growpart /dev/xvda 1

The command output is as follows:

```
CHANGED: partition=1 start=2048 old: size=79978496 end=79980544 new:
size=83873317,end=83875365
```

- o. Run the following command to query the partitions of **/dev/xvda**:

parted -l /dev/xvda

The command output is as follows:

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	42.9GB	42.9GB	primary	ext4	boot

- p. Run the following command to obtain the file system type and UUID:

blkid

The command output is as follows:

```
/dev/xvda1: UUID="7c4fce5d-f8f7-4ed6-8463-f2bd22d0ddea" TYPE="ext4"
```

- q. Stop the ECS and use it to create a private image.

```
[root@sluo-ecs-e6dc-resizefs ~]# poweroff
Connection closed by foreign host.
Disconnected from remote host at 11:08:54.
Type 'help' to learn how to use Xshell prompt.
```

- r. Use the created image to provision an ECS with a 100 GB system disk. Log in to the ECS and run the following command to query the partitions of **/dev/xvda**:

parted -l /dev/xvda

As shown in the command output, the root partition has been expanded to 107 GB.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	107GB	107GB	primary	ext4	boot

 NOTE

The value of **Size** is the size of the expanded root partition.

6.9 ECS Creation

6.9.1 Can an ECS Created from a Private Image Have Different Hardware Specifications from the ECS Used to Create the Private Image?

When you use a private image to create an ECS, you can specify the system disk size of the ECS. However, the disk size must be greater than or equal to the system disk size in the image and less than 1024 GB. Specifications of the CPU, memory, bandwidth, and data disks can be different from those in the image if necessary.

6.9.2 Can I Specify the System Disk Size When I Create an ECS Using an Image?

If you use an image to create an ECS, you can specify the system disk size of the ECS. However, the disk size must be greater than or equal to the system disk size in the image and less than 30,768 GB.

NOTE

You must ensure that your ECS OS supports the system disk size you specified during ECS creation.

6.9.3 What Do I Do If No Partition Is Found During the Startup of an ECS Created from an Imported Private Image?

Symptom

This may be caused by the disk partition ID change after the cross-platform image import. As a result, no partition can be found based on the original disk partition ID in the image. In this case, you need to change the disk partition in the image (**UUID=UUID of the disk partition**).

Solution

The following uses openSUSE 13.2 as an example to describe how to change the partition name.

1. Run the following command to query the disk partition ID:

```
ls -l /dev/disk/by-id/
```

The example command output is as follows.

```
total 0
lrwxrwxrwx 1 root root 10 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001 -> ../../xvda
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part1 -> ../../xvda1
lrwxrwxrwx 1 root root 12 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part10 -> ../../xvda10
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part2 -> ../../xvda2
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part5 -> ../../xvda5
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part6 -> ../../xvda6
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part7 -> ../../xvda7
```

```
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part8 -> ../xvda8
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part9 -> ../xvda9
lrwxrwxrwx 1 root root 10 Jul 22 01:35 ata-QEMU_HARDDISK_QM00005 -> ../xvde
lrwxrwxrwx 1 root root 10 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001 -> ../xvda
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part1 -> ../xvda1
lrwxrwxrwx 1 root root 12 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part10 -> ../xvda10
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part2 -> ../xvda2
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part5 -> ../xvda5
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part6 -> ../xvda6
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part7 -> ../xvda7
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part8 -> ../xvda8
lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part9 -> ../xvda9
lrwxrwxrwx 1 root root 10 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00005 -> ../xvde
```

ata-QEMU_HARDDISK_xxx and **scsi-SATA_QEMU_HARDDISK_xxx** indicate that the disk of the ECS is simulated using Quick EMUlator (QEMU). The content at the left of **->** is the disk partition ID, and that at the right of **->** is the partition name.

2. Run the following command to query the disk partition UUID:

ls -l /dev/disk/by-uuid/

The example command output is as follows.

```
total 0
lrwxrwxrwx 1 root root 11 Jul 22 01:35 45ecd7a0-29da-4402-a017-4564a62308b8 -> ../xvda5
lrwxrwxrwx 1 root root 11 Jul 22 01:35 55386c6a-9e32-41d4-af7a-e79596221f51 -> ../xvda9
lrwxrwxrwx 1 root root 11 Jul 22 01:35 55f36660-9bac-478c-a701-7ecc5347f789 -> ../xvda8
lrwxrwxrwx 1 root root 11 Jul 22 01:35 780f36bc-0ada-4c98-9a8d-44570d65333d -> ../xvda1
lrwxrwxrwx 1 root root 11 Jul 22 01:35 b3b7c47f-6a91-45ef-80d6-275b1cc16e19 -> ../xvda6
lrwxrwxrwx 1 root root 11 Jul 22 01:35 ea63b55d-3b6e-4dcd-8986-956b72bac3e9 -> ../xvda7
lrwxrwxrwx 1 root root 12 Jul 22 01:35 eb3cc645-925e-4bc5-bedf-c2a6f3b65809 -> ../xvda10
```

The content at the left of **->** is the disk partition UUID, and that at the right of **->** is the partition name. Obtain the relationship between the disk partition name, partition ID, and partition UUID.

3. Run the following command to check the partition names in the **/etc/fstab** file:

vi /etc/fstab

The example command output is as follows.

```
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part5 / ext3 defaults,errors=panic 1 1
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part1 /boot ext3 defaults,errors=panic 1 2
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part6 /home ext3 nosuid,errors=panic 1 2
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part10 /opt ext3 defaults,errors=panic 1 2
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part7 /tmp ext3 nodev,nosuid,errors=panic 1 2
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part9 /usr ext3 defaults,errors=panic 1 2
/dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part8 /var ext3 nodev,nosuid,errors=panic 1 2
sysfs /sys sysfs noauto 0 0
proc /proc proc defaults 0 0
usbfs /proc/bus/usb usbfs noauto 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0
/dev/cdrom /media/ udf,iso9660 noexec,noauto,nouser,nodev,nosuid 1 2
tmpfs /dev/shm tmpfs noexec,nodev,nosuid 0 0
```

The values in the first column are the disk partition IDs.

4. Press **i** to enter editing mode. Change the disk partition ID in the row that contains **/dev/disk/xxx** in the **/etc/fstab** file in step 3 to **UUID=UUID of the disk partition** based on the query results in step 1 and step 2.

The modified content is as follows.

```
UUID=45ecd7a0-29da-4402-a017-4564a62308b8 / ext3 defaults,errors=panic 1 1
UUID=780f36bc-0ada-4c98-9a8d-44570d65333d /boot ext3 defaults,errors=panic 1 2
UUID=b3b7c47f-6a91-45ef-80d6-275b1cc16e19 /home ext3 nosuid,errors=panic 1 2
UUID=eb3cc645-925e-4bc5-bedf-c2a6f3b65809 /opt ext3 defaults,errors=panic 1 2
UUID=ea63b55d-3b6e-4dcd-8986-956b72bac3e9 /tmp ext3 nodev,nosuid,errors=panic 1 2
UUID=55386c6a-9e32-41d4-af7a-e79596221f51 /usr ext3 defaults,errors=panic 1 2
```

```
UUID=55f36660-9bac-478c-a701-7ecc5347f789 /var ext3 nodev,nosuid,errors=panic 1 2
sysfs /sys sysfs noauto 0 0
proc /proc proc defaults 0 0
usbfs /proc/bus/usb usbfs noauto 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0
/dev/cdrom /media/ udf,iso9660 noexec,noauto,nouser,nodev,nosuid 1 2
tmpfs /dev/shm tmpfs noexec,nodev,nosuid 0 0
```

NOTE

Ensure that the UUIDs are correctly modified. Otherwise, the ECS cannot start properly.

5. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

6. Check the partition names in the system boot configuration file.

The system boot configuration files for different OSs vary. Confirm the boot configuration file of the current OS.

- Grand Unified Boot Loader (GRUB) configuration file

- /boot/grub/grub.conf
- /boot/grub/menu.lst
- /boot/grub/grub.cfg
- /boot/grub2/grub.cfg

- Syslinux configuration file

- /extlinux.conf
- /boot/syslinux/extlinux.conf
- /boot/extlinux/extlinux.conf
- /boot/syslinux/syslinux.cfg
- /syslinux/syslinux.cfg
- /syslinux.cfg

The boot file in this example is **/boot/grub/menu.lst**. Run the following command to check it:

vi /boot/grub/menu.lst

```
default 0
timeout 3
title xxx Server OS - xxxxxx
kernel /boot/vmlinuz-3.0.101-0.47.52-default root=/dev/disk/by-id/scsi-
SATA_QEMU_HARDDISK_QM00001-part5 resume= memmap=0x2000000$0x3E000000
nmi_watchdog=2 crashkernel=512M-:256M console=ttyS0,115200 console=tty0 xen_emul_unplug=all
initrd /boot/initrd-3.0.101-0.47.52-default
```

7. Press **i** to enter editing mode and change the partition names in the system boot configuration file.

Change the disk partition name in the **/boot/grub/menu.lst** file in **6** to **UUID=UUID of the disk partition** based on the query results in **1** and **2**.

```
default 0
timeout 3
title xxx Server OS - xxxxxx
kernel /boot/vmlinuz-3.0.101-0.47.52-default root=UUID=45ecd7a0-29da-4402-a017-4564a62308b8
```

```
resume= memmap=0x2000000$0x3E000000 nmi_watchdog=2 crashkernel=512M-:256M  
console=ttyS0,115200 console=tty0 xen_emul_unplug=all  
initrd /boot/initrd-3.0.101-0.47.52-default
```

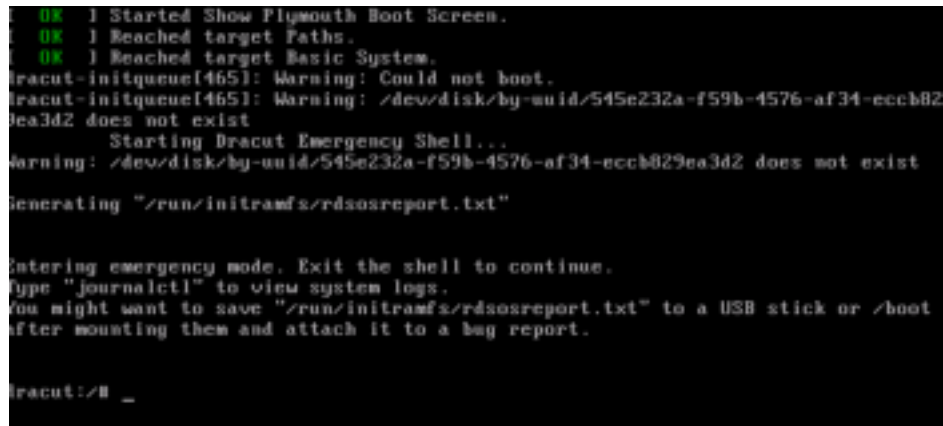
8. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

6.9.4 What Do I Do If the Disks of an ECS Created from a CentOS Image Cannot Be Found?

Symptom

Generally, this is because the `xen-blkfront.ko` module is not loaded during the startup. You need to modify OS kernel startup parameters. [Figure 6-8](#) shows the startup screen after the login to the ECS.

Figure 6-8 Startup screen



```
OK | Started Show Plymouth Boot Screen.  
OK | Reached target Paths.  
OK | Reached target Basic System.  
dracut-initqueue[465]: Warning: Could not boot.  
dracut-initqueue[465]: Warning: /dev/disk/by-uuid/545e232a-f59b-4576-af34-ecb829ea3d2 does not exist  
Starting Dracut Emergency Shell...  
Warning: /dev/disk/by-uuid/545e232a-f59b-4576-af34-ecb829ea3d2 does not exist  
Generating "/run/initramfs/rdsosreport.txt"  
  
Entering emergency mode. Exit the shell to continue.  
Type "journalctl" to view system logs.  
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot  
after mounting them and attach it to a bug report.  
  
dracut:/# _
```

Solution

Perform the following operations to modify OS kernel boot parameters:

NOTE

These operations must be performed after the OS starts. You are advised to modify kernel boot parameters in the ECS used for creating the image.

1. Run the following command to log in to the OS:
lsinitrd /boot/initramfs-`uname -r`.img |grep -i xen
 - If the command output contains **xen-blkfront.ko**, contact the customer service.
 - If no command output is displayed, go to [2](#).
2. Back up the GRUB configuration file.
 - If the ECS runs CentOS 6, run the following command:
cp /boot/grub/grub.conf /boot/grub/grub.conf.bak
 - If the ECS runs CentOS 7, run the following command:
cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.bak
3. Use the vi editor to open the GRUB configuration file. Run the following command (using CentOS 7 as an example):

vi /boot/grub2/grub.cfg

4. Add **xen_emul_unplug=all** to the default boot kernel.

NOTE

Search for the line that contains **root=UUID=** and add **xen_emul_unplug=all** to the end of the line.

```
menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core) with debugging' --class centos --class gnu-  
linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-  
advanced-bf3cc825-7638-48d8-8222-cd2f412dd0de' {  
    load_video  
    set gfxpayload=keep  
    insmod gzio  
    insmod part_msdos  
    insmod ext2  
    set root='hd0,msdos1'  
    if [ x$feature_platform_search_hint = xy ]; then  
        search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' bf3cc825-7638-48d8-8222-  
cd2f412dd0de  
    else  
        search --no-floppy --fs-uuid --set=root bf3cc825-7638-48d8-8222-cd2f412dd0de  
    fi  
    linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=UUID=bf3cc825-7638-48d8-8222-  
cd2f412dd0de xen_emul_unplug=all ro crashkernel=auto rhgb quiet systemd.log_level=debug  
systemd.log_target=kmsg  
    initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img  
}
```

5. Press **Esc**, enter **:wq**, and press **Enter** to exit the vi editor.
6. Create an image using the ECS, upload and register the image on the cloud platform.

6.9.5 What Do I Do If an ECS Created from a Windows Image Failed to Start When I Have Selected Enable Automatic Configuration During Image Registration?

Symptom

This issue is probably caused by the failure of offline VirtIO driver injection.

Solution

When you inject the VirtIO driver for a Windows ECS offline, there are some restrictions:

- If the boot mode in the image file is UEFI, the VirtIO driver cannot be injected offline.
- It is recommended that you disable Group Policy Object (GPO). Some policies may cause the failure to inject the VirtIO driver offline.
- It is recommended that you stop the antivirus software. Otherwise, the VirtIO driver may fail to be injected offline.

To update the VirtIO driver, you must install UVP VMTools. For how to install UVP VMTools, see [Optimizing a Windows Private Image](#).

6.9.6 What Do I Do If an Exception Occurs When I Start an ECS Created from an Image Using the UEFI Boot Mode?

Symptom

An ECS created from a private image using the UEFI boot mode cannot start.

Possible Causes

The image OS uses the UEFI boot mode, but the uefi attribute is not added to the image attributes.

Solution

1. Delete the ECS that failed to start.
2. Call the API to update the image attributes and change the value of **hw_firmware_type** to **uefi**.
API URI: PATCH /v2/cloudimages/{image_id}
For details about how to call the API, see "Updating Image Information" in *Image Management Service API Reference*.
3. Use the updated image to create an ECS.

A Change History

Released On	Description
2021-03-30	This issue is the fourth official release. Modified the following content: - Added CentOS 7.8/7.9, Ubuntu 20.04, and SLES 12 SP5 in Formats and OSs Supported for External Image Files. - Added the startup file <code>/boot/efi/EFI/euleros/grub.cfg</code> of EulerOS 2.9 in Changing the Disk Identifier in the GRUB Configuration File to UUID. - Added the configuration of Cloud-Init 18.3 and later versions in Configuring Cloud-Init.
2020-12-31	This issue is the third official release. Added the following content: Checking the Disk Capacity of an Image
2019-11-30	This issue is the second official release. Optimized the document, including structure adjustment, content optimization, and flowchart supplementation.
2018-08-16	This issue is the first official release.