



Red Hat Satellite 6.1

Installation Guide

Installing and Configuring Satellite

Edition 4

Last Updated: 2017-08-29

Red Hat Satellite 6.1 Installation Guide

Installing and Configuring Satellite
Edition 4

Red Hat Satellite Documentation Team

Legal Notice

Copyright © 2015 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to install Red Hat Satellite. It also steps through the basic configuration requirements to get Satellite running in your environment.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT SATELLITE	4
1.1. RED HAT SATELLITE 6 SYSTEM ARCHITECTURE	4
1.2. RED HAT SATELLITE 6 SYSTEM COMPONENTS	7
1.3. RED HAT SATELLITE 6 SUPPORTED USAGE	7
1.4. PREREQUISITES	8
CHAPTER 2. INSTALLING RED HAT SATELLITE SERVER	21
2.1. OBTAINING THE REQUIRED PACKAGES	21
2.2. RUNNING THE INSTALLATION AND CONFIGURATION PROGRAM	24
2.3. OPTIONAL CONFIGURATION OPTIONS	27
CHAPTER 3. LOGGING IN TO RED HAT SATELLITE	34
3.1. ORGANIZATIONS	34
3.2. CHANGING YOUR ACCOUNT PREFERENCES	36
3.3. ADDITIONAL RESOURCES	37
CHAPTER 4. POPULATING RED HAT SATELLITE WITH CONTENT	38
4.1. CONNECTED SATELLITE	38
4.2. DISCONNECTED SATELLITE	46
CHAPTER 5. CONFIGURING A SELF-REGISTERED SATELLITE	56
5.1. REGISTERING A SATELLITE TO ITSELF	56
5.2. UPDATING A SELF-REGISTERED SATELLITE	59
CHAPTER 6. MANAGING HYPERVISORS AND VIRTUAL GUEST SUBSCRIPTIONS	62
6.1. INTRODUCTION TO VIRT-WHO	62
6.2. BEFORE YOU BEGIN	63
6.3. SUPPORTED HYPERVISORS	65
6.4. SETTING UP A RED HAT ENTERPRISE VIRTUALIZATION MANAGER SERVER OR LIBVIRT (KVM) HYPERVISOR	66
6.5. USING VIRT-WHO WITH HYPER-V	68
6.6. SETTING UP A VMWARE HYPERVISOR	69
6.7. CONFIGURE VIRT-WHO WITH AN ENCRYPTED PASSWORD	72
6.8. VCENTER CONFIGURATION EXAMPLE FOR REPORTING DATA TO MULTIPLE ORGANIZATIONS	73
6.9. REGISTERING GUEST INSTANCES	75
6.10. REMOVING A GUEST ENTRY	76
6.11. REMOVING A HYPERVISOR ENTRY	76
6.12. TROUBLESHOOTING VIRT-WHO	76
CHAPTER 7. INSTALLING RED HAT SATELLITE CAPSULE SERVER	78
7.1. RED HAT SATELLITE CAPSULE SERVER SCALABILITY	78
7.2. RED HAT SATELLITE CAPSULE SERVER PREREQUISITES	79
7.3. OBTAINING THE REQUIRED PACKAGES FOR THE CAPSULE SERVER	85
7.4. RUNNING THE INSTALLATION AND CONFIGURATION PROGRAM FOR CAPSULE SERVER	87
7.5. OPTIONAL CONFIGURATION OPTIONS	90
7.6. ADDING LIFE CYCLE ENVIRONMENTS TO A RED HAT SATELLITE CAPSULE SERVER	96
7.7. REMOVING LIFE CYCLE ENVIRONMENTS FROM THE RED HAT SATELLITE CAPSULE SERVER	97
7.8. REGISTERING HOST SYSTEMS TO A RED HAT SATELLITE CAPSULE SERVER	98
7.9. CONFIGURING SATELLITE 6 WITH EXTERNAL SERVICES	100
CHAPTER 8. UPGRADING RED HAT SATELLITE SERVER AND CAPSULE SERVER	116
8.1. UPGRADING RED HAT SATELLITE	116
8.2. UPGRADING RED HAT SATELLITE CAPSULE	122

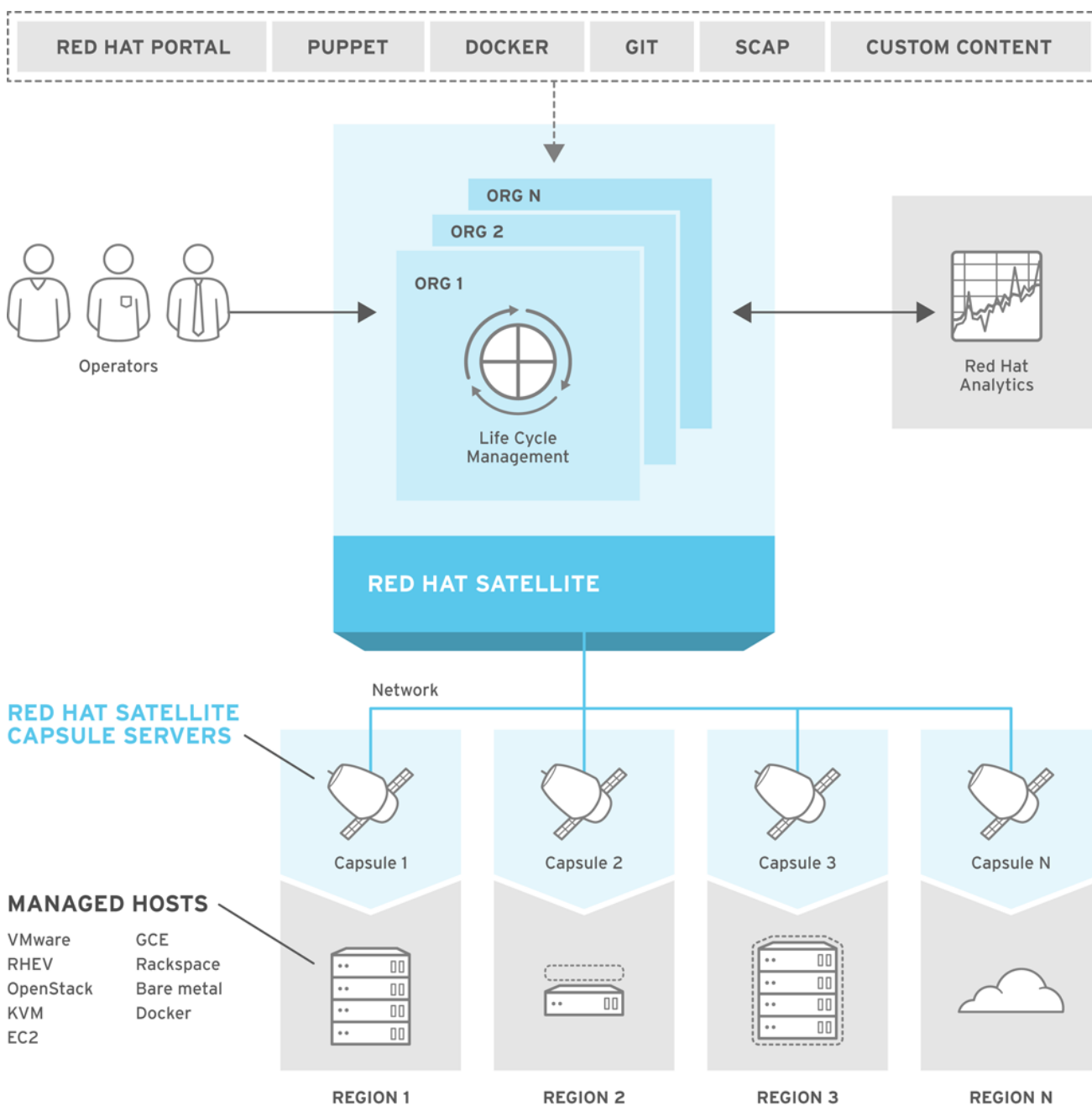
8.3. UPGRADING THE DISCOVERY FEATURE	125
8.4. UPGRADING RED HAT SATELLITE CLIENTS	126
8.5. UPGRADING BETWEEN MINOR VERSIONS OF SATELLITE	127
CHAPTER 9. NEXT STEPS	130
CHAPTER 10. UNINSTALLING RED HAT SATELLITE SERVER AND CAPSULE SERVER	131
REMOVING SATELLITE SERVER	131
REMOVING CAPSULE SERVER	131
APPENDIX A. GLOSSARY OF TERMS	132
APPENDIX B. REVISION HISTORY	136

CHAPTER 1. INTRODUCTION TO RED HAT SATELLITE

Red Hat Satellite 6 is the evolution of Red Hat's life cycle management platform. It provides the capabilities that administrators have come to expect in a tool focused on managing systems and content for a global enterprise. Satellite 6 covers the use cases requested by Satellite 5 customers, but also includes functionality that enables larger scale, federation of content, better control of systems during the provisioning process, and a much more simplified approach to life cycle management. Satellite 6 also further evolves the inherent approach to certificate-based entitlements and integrated subscription management. Satellite 6 is based on years of customer feedback and is an evolution of previous versions.

1.1. RED HAT SATELLITE 6 SYSTEM ARCHITECTURE

The following diagram represents the high-level architecture of Red Hat Satellite 6.



SATELLITE6_352601_0715

Figure 1.1. Red Hat Satellite 6 System Architecture

There are four stages through which content flows in this architecture:

External Content Sources

The Red Hat Satellite Server can consume diverse types of content from various sources. The required connection is the one with Red Hat Customer Portal, which is the primary source of software packages, errata, Puppet modules, and container images. In addition, you can use other supported content sources (Git repositories, Docker Hub, Puppet Forge, SCAP repositories) as well as your organization's internal data store.

Red Hat Satellite Server

The Red Hat Satellite Server enables you to plan and manage the content life cycle and the configuration of Capsule Servers and hosts through GUI, CLI, or API.

The Satellite Server organizes the life cycle management by using *organizations* as principal division units. Organizations isolate content for groups of hosts with specific requirements and administration tasks. For example, the OS build team can use a different organization than the web development team.

The Satellite Server also contains a fine-grained authentication system to provide Satellite operators with permissions to access precisely the parts of the infrastructure that lie in their area of responsibility.

Capsule Servers

Capsule Servers mirror content from the Satellite Server to establish content sources in various geographical locations. This allows host systems to pull content and configuration from the Satellite Capsule Servers in their location and not from the central Satellite Server. The recommended minimal number of Capsule Servers is therefore given by the number of geographic regions where the organization that uses Satellite operates.

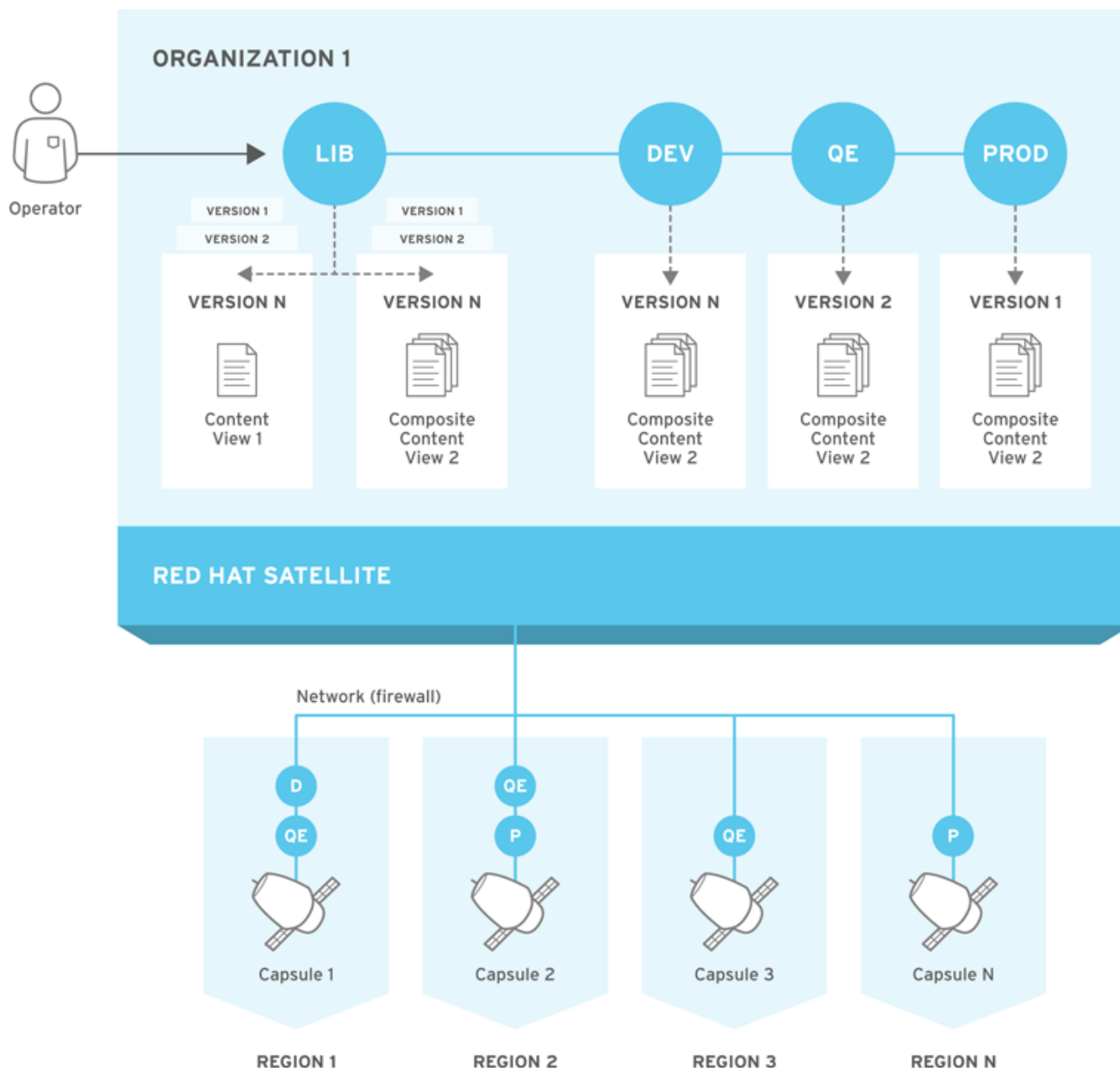
Using Content Views, you can specify the exact subset of content that the Capsule Server makes available to hosts. See [Figure 1.2, “Content Life Cycle in Red Hat Satellite 6”](#) for a closer look at life cycle management with the use of Content Views.

The communication between managed hosts and the Satellite Server is routed through the Capsule Server that can also manage multiple services on behalf of hosts. Many of these services use dedicated network ports, but the Capsule Server ensures that a single source IP address is used for all communications from the host to the Satellite Server, which simplifies firewall administration.

Managed Hosts

Hosts are the recipients of content from Capsule Servers. Hosts can be either physical or virtual (deployed on KVM, VMware vSphere, OpenStack, Amazon EC2, Rackspace Cloud Services, Google Compute Engine, or in a Docker container). The Satellite Server can have directly managed hosts. The base system running a Capsule Server is also a managed host of the Satellite Server.

The following diagram provides a closer look at the distribution of content from the Satellite Server to Capsules.



SATELLITE6_352601_0615

Figure 1.2. Content Life Cycle in Red Hat Satellite 6

By default, each organization has a Library of content from external sources. Content Views are subsets of content from the Library created by intelligent filtering. You can publish and promote Content Views into life cycle environments (typically Dev, QA, and Production). When creating a Capsule Server, you can choose which life cycle environments will be copied to that Capsule and made available to managed hosts.

Content Views can be combined to create Composite Content Views. For example, it is beneficial to have a separate Content View for packages required by an operating system and a separate one for packages required by an application. Which Content Views should be promoted to which Capsule Server depends on the Capsule's intended functionality. Any Capsule Server can run DNS, DHCP, and TFTP as infrastructure services that can be supplemented, for example, with content or configuration services.

You can update the Capsule Server by creating a new version of a Content View using synchronized content from the Library. The new Content View version is then promoted through life cycle environments. You can also create in-situ updates of Content Views,

which means that a minor version of the Content View is created in its current life cycle environment without promoting it from the Library.

1.2. RED HAT SATELLITE 6 SYSTEM COMPONENTS

Red Hat Satellite 6 consists of several open source projects which are integrated, verified, delivered and supported as Satellite 6. It is often important to understand which upstream versions of these projects are delivered. This information is maintained and regularly updated on the [Red Hat Customer Portal](#)^[1].

Red Hat Satellite 6 consists of the following open source projects:

Foreman

Foreman is an open source application used for provisioning and life cycle management of physical and virtual systems. Foreman automatically configures these systems using various methods, including kickstart and Puppet modules. Foreman also provides historical data for reporting, auditing, and troubleshooting.

Katello

Katello is a Foreman plug-in for subscription and repository management. It provides a means to subscribe to Red Hat repositories and download content. You can create and manage different versions of this content and apply them to specific systems within user-defined stages of the application life cycle.

Candlepin

Candlepin is a service within Katello that handles subscription management.

Pulp

Pulp is a service within Katello that handles repository and content management.

Hammer

Hammer is a CLI tool that provides command line and shell equivalents of most Web UI functions.

REST API

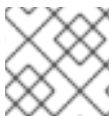
Red Hat Satellite 6 includes a RESTful API service that allows system administrators and developers to write custom scripts and third-party applications that interface with Red Hat Satellite.

1.3. RED HAT SATELLITE 6 SUPPORTED USAGE

Each Red Hat Satellite subscription includes one supported instance of Red Hat Enterprise Linux Server. This instance should be reserved solely for the purpose of running Red Hat Satellite. Using the operating system included with Satellite to run other daemons, applications, or services within your environment is not supported.



NOTE



All Red Hat Satellite components and their usage are supported within the context of Red Hat Satellite only. Third-party usage of any components falls beyond supported usage.

Support for Red Hat Satellite components is described below.

Puppet

Red Hat Satellite 6 includes supported puppet packages. The installation program allows users to install and configure Puppet Masters as a part of Red Hat Satellite Capsule Servers. The server installs the Hiera key-value database, which can be used to refine how Puppet modules are applied. A Puppet module, running on a Puppet Master on the Red Hat Satellite Server or Satellite Capsule Server, using Hiera, is supported by Red Hat.

Red Hat supports many different scripting and other frameworks, including puppet modules. Support for these frameworks is based on the article "[How does Red Hat support scripting frameworks?](#)"^[2]

Pulp

Pulp is the content management subsystem within Red Hat Satellite 6. Pulp usage is only supported via the Satellite Server web UI, CLI, and API. Direct modification or interaction with Pulp's local API or database is not supported.

Red Hat does not support direct modification with Pulp as this can cause irreparable damage to the Red Hat Satellite 6 databases.

Foreman

Foreman makes up a large amount of Red Hat Satellite's core functionality including the web UI container, users, organizations, security and other significant functions. Foreman can be extended using plug-ins. However, only Red Hat Satellite packaged plug-ins are supported. Red Hat does not support plug-ins in the Red Hat Satellite Optional repository.

Red Hat Satellite also includes components, configuration and functionality to provision and configure operating systems other than Red Hat Enterprise Linux. While these features are included and can be employed, Red Hat supports their usage for Red Hat Enterprise Linux.

Candlepin

Candlepin is the subscription management subsystem within Red Hat Satellite 6. The only supported methods of using Candlepin are through the Red Hat Satellite 6 web UI, CLI, and API.

Red Hat does not support direct modification and interactions with Candlepin, its local API or database, as this can cause irreparable damage to the Red Hat Satellite 6 databases.

Embedded Tomcat Application Server

The only supported methods of using the embedded Tomcat application server are through the Red Hat Satellite 6 web UI, API, and database. Red Hat does not support direct interactions and modifications of the embedded Tomcat application server's local API or database.

1.4. PREREQUISITES

The following conditions must be met before installing Red Hat Satellite 6:

IMPORTANT

The Red Hat Satellite server and Capsule server versions must match. For example, a Satellite 6.0 server cannot run a 6.1 Capsule server and a Satellite 6.1 server cannot run a 6.0 Capsule server. Mismatching Satellite server and Capsule server versions will result in the Capsule server failing silently.

1.4.1. Base Operating System**IMPORTANT**

Red Hat Satellite is only supported on the latest version of Red Hat Enterprise Linux 6 Server or 7 Server. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

Install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports. Register and attach a subscription to the system as follows:

```
# subscription-manager register
# subscription-manager list --available --all
# subscription-manager subscribe --pool=Red_Hat_Enterprise_Linux_Pool_Id
```

IMPORTANT

- Red Hat Satellite Server requires Red Hat Enterprise Linux installations with the @Base package group with no other package-set modifications, and without third-party configurations or software that is not directly necessary for the direct operation of the server. This restriction includes hardening or other non-Red Hat security software. If such software is required in your infrastructure, install and verify a complete working Satellite Server first, then create a backup of the system before adding any non-Red Hat software.
- Your subscription-manager 'Release' field must be set to 6Server or 7Server in order to receive the latest version of Red Hat Enterprise Linux and Red Hat Satellite during the installation. Set the field by using the command:

```
# subscription-manager release --set=Release
```

Only release versions 6Server and 7Server are supported by Red Hat Satellite.

- Update the system to the latest set of packages in Red Hat Enterprise Linux after setting the release:

```
# yum update
```

- Red Hat recommends that the Satellite Server be a freshly provisioned system that serves no other function except as a Satellite Server.

- Red Hat Satellite requires a networked base system with the following minimum specifications:
 - 64-bit architecture
 - The latest version of Red Hat Enterprise Linux 6 Server or 7 Server
 - A minimum of two CPU cores, but four CPU cores are recommended.
 - A minimum of 12 GB memory but ideally 16 GB of memory for each instance of Satellite. A minimum of 4 GB of swap space is recommended.
 - A unique hostname. The hostname can contain lower-case letters, numbers, dots (.) and hyphens (-).
 - No Java virtual machine installed on the system, remove any if they exist.
 - No Puppet RPM files installed on the system.
 - No third-party unsupported yum repositories enabled. Third-party repositories may offer conflicting or unsupported package versions that may cause installation or configuration errors.
- A current Red Hat Network subscription.
- Administrative user (**root**) access.
- Full forward and reverse DNS resolution using a fully qualified domain name. Ensure that **hostname** and **localhost** resolve correctly, using the following commands:

```
# ping -c1 localhost
# ping -c1 `hostname -f` # my_system.domain.com
```



IMPORTANT

Ensure that the host system is fully updated before installing Red Hat Satellite. Attempts to install on host systems that are not fully updated may lead to difficulty in troubleshooting, as well as unpredictable results.

1.4.2. Supported Browsers

Browser support is divided into 4 levels:

1. Level 1: Fully supported preferred browsers for ideal experience.
2. Level 2: Mostly supported. The interface functions but some design elements may not align correctly, UI controls and layout may be misaligned and there maybe degraded performance experienced.
3. Level 3: Design elements may not align correctly.
4. Level 4: Unsupported

The table below outlines the supported browsers and their level of support:

Table 1.1. Supported Browser Matrix

Browser	Version	Support Level
Firefox	3.6	L3
Firefox	17, 18, 19, 20	L4
Firefox	21	L2
Firefox	22, 23, 24	L1
Firefox	Latest	L1
Chrome	19, 20	L4
Chrome	21, 27	L2
Chrome	Latest	L1
Internet Explorer	7, 8	L4
Internet Explorer	9, 10, 11	L2
Safari	ALL	L4



NOTE

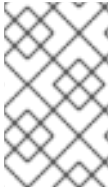
The web UI and command-line interface for Satellite Server supports English, Portuguese, Simplified Chinese, Traditional Chinese, Korean, Japanese, Italian, Spanish, Russian, French, and German.

1.4.3. Storage

Satellite Server storage specifications are as follows:

- A minimum of 6 GB storage for base operating system installation of Red Hat Enterprise Linux.
- A minimum of 400 MB storage for the Red Hat Satellite 6 software installation.
- A minimum of 20 GB storage for each unique software repository. Packages that are duplicated in different repositories are only stored once on the disk. Additional repositories containing duplicate packages will require less additional storage. The bulk of storage resides on the **/var/lib/mongodb** and **/var/lib/pulp** directories. These end points are not manually configurable. Make sure that storage is available on the **/var** file system to prevent storage issues.
- A minimum of 2 GB of available storage in **/var/lib/pgsql** with the ability to grow the partition containing this directory as data storage requirements grow.

- If you are using a disconnected installation, a copy of the repositories used in the installation are stored in the **/opt/** directory. Ensure you have a minimum of 2GB of space for this file system and directory.

**NOTE**

Most Satellite Server data is stored within the **/var** directory. It is strongly recommended to mount **/var** on LVM storage that the system can scale to meet data storage requirements.

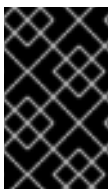
**NOTE**

The XFS file system is recommended for Red Hat Satellite 6. XFS is the default file system in Red Hat Enterprise Linux 7, which makes it the preferable base operating system. If you intend to use Red Hat Enterprise Linux 6 instead, contact your account team to learn about enabling XFS on this system. Alternatively, make sure that you have an ext4 file system with sufficient amount of inodes for your intended Satellite deployment.

The following table details recommended storage requirements for specific directories. These values are based on expected use case scenarios and may vary according to individual environments.

Table 1.2. Recommended Storage Considerations

Directory	Installation Size Requirement	Runtime Requirement with Red Hat Enterprise Linux 5/6/7 synchronized
/var/lib/pulp	1 MB	200 GB
/var/lib/mongodb	3.5 GB	15 GB
/var/log	10 MB	100 MB
/var/lib/pgsql	100 MB	250 MB

**IMPORTANT**

Several components of Red Hat Satellite are sensitive to network latency. Red Hat recommends local or SAN-based storage. Avoid NFS storage whenever possible.

1.4.4. Application Specifications

Satellite Server application installation specifications are as follows:

Red Hat recommends that a time synchronizer such as **ntp** is installed and enabled on the host operating system before installing Satellite to minimize the effects of any time drift.

For Red Hat Enterprise Linux 6, run the following commands to start the **ntpd** service and have it persist across restarts:

```
# service ntpd start
# chkconfig ntpd on
```

In Red Hat Enterprise Linux 7, **chrony** is the default time synchronizer. Run the following commands to start the **chronyd** service and have it persist across restarts:

```
# systemctl start chronyd
# systemctl enable chronyd
```

1.4.5. Network Ports Required for Satellite Communications

The tables in this section list the ports required for configuring Red Hat Satellite Server. A list of ports can also be found in the Red Hat Knowledgebase solution [Satellite 6.1 Definitive List of Ports](#).

Table 1.3. Ports for Browser-based User Interface Access to Satellite

Port	Protocol	Service	Required for
443	TCP	HTTPS	For Browser-based UI Access to Satellite
Optional			
80	TCP	HTTP	To enable redirection to HTTPS for web UI Access to Satellite

Table 1.4. Ports for Satellite to Red Hat CDN Communication

Port	Protocol	Service	Required for
443	TCP	HTTPS	Subscription Management Services, connecting to the Red Hat CDN

Table 1.5. Ports for Client to Satellite Communication

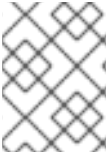
Port	Protocol	Service	Required for
53	TCP and UDP	DNS	Queries to the Satellite's integrated DNS service
67	UDP	DHCP	For Client provisioning from the integrated Capsule

Port	Protocol	Service	Required for
69	UDP	TFTP	Downloading PXE boot image files from the integrated Capsule
80	TCP	HTTP	Anaconda, yum, for obtaining Katello certificates, templates, and for downloading iPXE firmware
443	TCP	HTTPS	Subscription Management Services, yum, Telemetry Services, and for connection to the Katello Agent
5647	TCP	amqp	The Katello agent to communicate with the Satellite's Qpid dispatch router
8140	TCP	HTTPS	Puppet agent to Puppet master connections

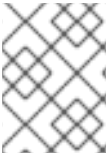
Any managed host that is directly connected to the Satellite Server is a Client in this context. This includes the base system on which a Capsule Server is running.

Table 1.6. Optional Network Ports

Port	Protocol	Service	Required for
8443	TCP	HTTP	Capsule to Client "reboot" command to a discovered host during provisioning
7911	TCP	DHCP	Capsule originated, for orchestration of DHCP records (local or external)[a]
5000	TCP	HTTP	Satellite originated, for compute resources in OpenStack or for running Docker containers
22, 16514	TCP	SSH/TLS	Satellite originated, for compute resources in libvirt
389, 636	TCP	SSH/TLS	Satellite originated, for LDAP and secured LDAP authentication sources
from 5910 to 5930	TCP	SSH/TLS	Satellite originated, for NoVNC console in Web UI to hypervisors
[a] If the DHCP service is provided by an external service, opening this port is required on the external server.			

**NOTE**

Port 8080 needs to be free, but not open, in order for subscription management services to access the Satellite Server.

**NOTE**

To configure the firewall on a **Capsule** to enable incoming connections from the **Satellite**, see [the section called “Connections from Satellite to Capsule”](#).

Connections from Client to Satellite

To configure the firewall on a **Satellite** to enable incoming connections from a **Client**, and to make these rules persistent during reboots, enter the commands below appropriate to the Red Hat release.

The ports in these commands are taken from the table [Table 1.5, “Ports for Client to Satellite Communication”](#). Note that port 80 and 443 are also listed in the [Table 1.3, “Ports for Browser-based User Interface Access to Satellite”](#). Review the commands to avoid duplicating entries.

- On a Red Hat Enterprise Linux 6 Satellite, execute as **root**:

```
# iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p udp --dport 67 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p udp --dport 69 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 5647 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 8140 -j ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

- On a Red Hat Enterprise Linux 7 Satellite, execute as **root**:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" \
--add-port="69/udp" --add-port="80/tcp" \
--add-port="443/tcp" --add-port="5647/tcp" \
--add-port="8140/tcp" \
&& firewall-cmd --permanent --add-port="53/udp" --add-port="53/tcp" \
```

```
--add-port="67/udp" \
--add-port="69/udp" --add-port="80/tcp" \
--add-port="443/tcp" --add-port="5647/tcp" \
--add-port="8140/tcp"
```

1.4.6. SELinux Policy on Satellite 6

Red Hat Satellite 6 uses a set of predefined ports, as described in the preceding section and in [Section 7.2.3, “Network Ports Required for Capsule Communications”](#). Because Red Hat recommends that SELinux on Satellite 6 systems be set to enforcing, if you need to change the port for any service, you also need to change the associated SELinux port type to allow access to the resources. For example, if you change the web UI ports (HTTP/HTTPS) to 8018/8019, you need to add these port numbers to the `httpd_port_t` SELinux port type.

[Table 1.7, “SELinux Commands to Change Default Port Assignments”](#) lists the required commands to change the Satellite 6 default ports to a user-specified port. These examples use port 99999 for demonstration purposes; ensure you change this value to suit your deployment.



NOTE

This change is also required for target ports; for example, when Satellite 6 connects to an external source, such as Red Hat Enterprise Virtualization Manager or OpenStack.

You only need to make changes to default port assignments once. Updating or upgrading Satellite has no effect on these assignments. Any updates only add default SELinux ports if no assignments exist.

Table 1.7. SELinux Commands to Change Default Port Assignments

Default Port	SELinux Command
80, 443, 8443	<code>semanage port -a -t http_port_t -p tcp 99999</code>
8080	<code>semanage port -a -t http_cache_port_t -p tcp 99999</code>
8140	<code>semanage port -a -t puppet_port_t -p tcp 99999</code>
9090	<code>semanage port -a -t websm_port_t -p tcp 99999</code>
69	<code>semanage port -a -t tftp_port_t -p udp 99999</code>
53 (TCP)	<code>semanage port -a -t dns_port_t -p tcp 99999</code>
53 (UDP)	<code>semanage port -a -t dns_port_t -p udp 99999</code>
67, 68	<code>semanage port -a -t dhcpd_port_t -p udp 99999</code>
5671	<code>semanage port -a -t amqp_port_t -p tcp 99999</code>

Default Port	SELinux Command
8000	<code>semanage port -a -t soundd_port_t -p tcp 99999</code>
7911	<code>semanage port -a -t dhcpd_port_t -p tcp 99999</code>
5000 on Red Hat Enterprise Linux 6	<code>semanage port -a -t complex_port_t -p tcp 99999</code>
5000 on Red Hat Enterprise Linux 7	<code>semanage port -a -t complex_main_port_t -p tcp 99999</code>
22	<code>semanage port -a -t ssh_port_t -p tcp 99999</code>
16514 (libvirt)	<code>semanage port -a -t virt_port_t -p tcp 99999</code>
389, 636	<code>semanage port -a -t ldap_port_t -p tcp 99999</code>
5910 to 5930	<code>semanage port -a -t vnc_port_t -p tcp 99999</code>

To allow Satellite 6 to connect to a service that is on a different port, for example, EC2 or an external repository served by an Apache **httpd** server, you need to add this port to the `virt_port_t` SELinux type, as follows:

```
# semanage port -a -t virt_port_t -p tcp 99999
```

IMPORTANT

If SELinux was *disabled* (as compared to enabled and running in permissive mode), when you installed Satellite, then you need to enable SELinux and run the following commands *in permissive mode* after you have completed the installation:

```
# foreman-selinux-enable
# foreman-selinux-relabel
```

Failure to run these commands can result in mislabeled files, AVC denials when attempting to access the web UI, and difficult troubleshooting.

Use the **semanage** command if you need to disassociate the previously used port number and port type. For example:

```
# semanage port -d -t virt_port_t -p tcp 99999
```

For more information about configuring SELinux, and ensuring that it is enabled on startup, see the following resources:

- [Enabling SELinux on Red Hat Enterprise Linux 6](#)^[3]
- [Enabling SELinux on Red Hat Enterprise Linux 7](#)^[4]

1.4.7. Considerations for Large Deployments

With more than 225 content hosts, the **qpidd** message broker can reach several system-level limits, resulting in Satellite's failure to operate. To avoid this, one or more of these limits must be increased before deploying a large number of content hosts.

Refer to the following table to confirm which values must be changed depending on the number of content hosts you plan to deploy. Then refer to the following sections for instructions on how to set these limits.

Table 1.8. Limits to be Increased for Large Deployments

Number of Content Hosts	Client Connections	File Descriptors	Parallel Asynchronous I/O Operations	Concurrent Locks	Memory Map Areas
More than 225	✓				
More than 500	✓	✓			
More than 1900	✓	✓	✓		
More than 30,000	✓	✓	✓	✓	
More than 32,900	✓	✓	✓	✓	✓

Increasing the Maximum Number of Client Connections

With more than 225 content hosts, **qpidd** reaches the maximum number of client connections. To increase it, first establish the new value of the limit that is calculated as:

$$(\text{number_of_content_hosts} \times 2) + 100$$

For example, a deployment with 300 content hosts requires at least 700 connections. Use the calculated value in **/etc/qpidd/qpidd.conf**:

```
max-connections=value
```

Increasing the Maximum Number of File Descriptors

With more than 500 content hosts, **qpidd** reaches the maximum number of file descriptors. To increase it, first establish the new value of the limit that is calculated as:

$$(\text{number_of_content_hosts} \times 4) + 500$$

For example, a deployment with 600 content hosts requires 2900 file descriptors. Use the calculated value in appropriate configuration files:

- On Red Hat Enterprise Linux 6, add the following line to **/etc/security/limits.conf**:

```
qpidd x nofile value
```

- On Red Hat Enterprise Linux 7, add the following line to **/usr/lib/systemd/system/qpidd.service** at the end of the [Service] section:

```
LimitNOFILE=value
```

Increasing the Maximum Number of Parallel Asynchronous I/O Operations

With more than 1900 content hosts, **qpidd** reaches the kernel limit of maximum parallel asynchronous I/O operations. To increase it, first establish the new value of the limit that is calculated as:

```
33 x number_of_content_hosts
```

Use the calculated value in **/etc/sysctl.conf**:

```
fs.aio-max-nr=value
```

Reload the setting by executing:

```
# sysctl -p
```

Increasing the Maximum Number of Concurrent Locks

With more than 30,000 content hosts, the back-end database of **qpidd** might reach the maximum number of concurrent locks. To increase this limit, create a configuration file in the directory where the **exchanges.db** file is stored. The directory location can vary. Confirm its location by searching the **/var/lib/qpidd/** directory:

```
# find /var/lib/qpidd -name exchanges.db
/var/lib/qpidd/qls/dat/exchanges.db
```

In the above example, **exchanges.db** is stored in the **/var/lib/qpidd/qls/dat/** directory. In this directory, create a **DB_CONFIG** file that must be owned and readable by the **qpidd** user. Add the following content to **DB_CONFIG**:

```
set_lk_max_locks 10000
set_lk_max_objects 10000
```

Increasing the Maximum Number of Memory Map Areas

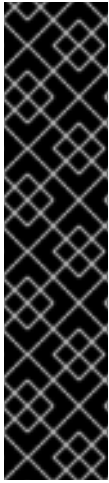
With more than 32,900 content hosts, **qpidd** reaches the kernel limit of maximum number of memory map areas per process. This problem occurs only on Red Hat Enterprise Linux 7.

Increase the limit by adding the following line to **/etc/sysctl.conf**:

```
vm.max_map_count = 655300
```

Reload the setting by executing:

```
# sysctl -p
```



IMPORTANT

It is required to restart **qpidd** to apply any changes to the aforementioned limits:

- On Red Hat Enterprise Linux 6:

```
# service qpidd restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart qpidd
```

1.4.8. Troubleshooting

Red Hat recommends to install the **sos** package on the host operating system before installing Satellite. The **sos** package provides the **sosreport** command that collects configuration and diagnostic information from a Red Hat Enterprise Linux system and is used to provide the initial analysis of a system required when opening a service request with Red Hat Technical Support. For more information on using **sosreport**, refer to the [What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later](#) article on Red Hat Customer Portal^[5].

To install the **sos** package run the following command:

```
# yum install sos
```

[1] <https://access.redhat.com/articles/1343683>

[2] <https://access.redhat.com/articles/369183>

[3] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Working_with_SELinux-Changing_SELinux_Modes.html#sect-Security-Enhanced_Linux-Enabling_and_Disabling_SELinux-Enabling_SELinux

[4] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/sect-Security-Enhanced_Linux-Working_with_SELinux-Enabling_and_Disabling_SELinux.html#sect-Security-Enhanced_Linux-Enabling_and_Disabling_SELinux-Enabling_SELinux

[5] <https://access.redhat.com/solutions/3592>

CHAPTER 2. INSTALLING RED HAT SATELLITE SERVER

This chapter describes how to obtain the required packages to install Red Satellite Server, whether you are connected to the network or not. You can then use the installation program, **katello-installer**, to install and configure the Satellite Server. Several configuration options are available; these are described in [Section 2.3, “Optional Configuration Options”](#).

2.1. OBTAINING THE REQUIRED PACKAGES

There are two ways to obtain the packages required to install a Satellite Server:

- Download the packages directly from the Red Hat Content Delivery Network (CDN).
- Download an ISO image of the packages required from an external computer.

Both methods are described in this section. However, for hosts that have network connectivity, Red Hat recommends downloading the packages directly from the CDN. Using ISO images is only recommended for hosts in a disconnected environment because ISO images may not contain the latest updates.

2.1.1. Downloading from a Connected Network

This section describes how to use Subscription Manager to download the required packages for Red Hat Satellite Server from the repository.

Procedure 2.1. To Download Satellite Server on a Certificate-managed System:

1. List all the available subscriptions to find the correct Red Hat Satellite and Red Hat Enterprise Linux product to allocate to your system:

```
# subscription-manager list --available --all
```

This command displays output similar to the following:

```
+-----+
| Available Subscriptions |
+-----+

Subscription Name: Red Hat Satellite Subscription
Provides:          Red Hat
                  Red Hat Satellite Capsule 6
                  Red Hat Enterprise Linux 7
                  Red Hat Satellite 6
SKU:              SKU123456
Pool ID:          e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Available:        6
Suggested:        1
Service Level:    Self-Support
Service Type:     L1-L3
Multi-Entitlement: No
Ends:             01/01/2022
System Type:      Physical
```

**NOTE**

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type. Take note of the pool IDs for Red Hat Satellite 6.1, Red Hat Enterprise Linux and Red Hat Software collections that correspond to your system version and product type.

2. Attach a subscription to the registered system:

```
# subscription-manager subscribe --pool=Red_Hat_Satellite_Pool_Id \
&& subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Pool_Id \
&& subscription-manager subscribe \
--pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_Id
```

3. Disable all existing repositories:

```
# subscription-manager repos --disable "*"

```

4. Enable the Red Hat Satellite and Red Hat Enterprise Linux and Red Hat Software Collections repositories. Ensure the Red Hat Enterprise Linux repository matches the specific version you are using.

For Red Hat Enterprise Linux 6:

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc-6-rpms \
--enable rhel-6-server-satellite-6.1-rpms
```

For Red Hat Enterprise Linux 7:

```
# subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-server-rhsc-7-rpms \
--enable rhel-7-server-satellite-6.1-rpms
```

**NOTE**

The commands above are based on Red Hat Enterprise Linux 6 and 7. If you are using a different version of Red Hat Enterprise Linux, change the repository based on your specific version.

5. If required, to verify what repositories have been enabled, use the **yum repolist enabled** command. For example, on Red Hat Enterprise Linux 7:

```
# yum repolist enabled
Loaded plugins: product-id, subscription-manager
repo id                                repo name
status
!rhel-7-server-rpms/x86_64             Red Hat Enterprise
Linux 7 Server (RPMs)                  9,889
!rhel-7-server-satellite-6.1-rpms/x86_64 Red Hat Satellite 6.1
```

(for RHEL 7 Server) (RPMs)	545
!rhel-server-rhsc1-7-rpms/x86_64	Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server	4,279
repolist: 14,713	

6. Install the katello package:

```
# yum install katello
```



IMPORTANT

The required packages are now installed. Proceed to [Section 2.2, “Running the Installation and Configuration Program”](#) to run the installation and configuration program.

2.1.2. Downloading from a Disconnected Network



NOTE

When the intended host for the Red Hat Satellite server is in a disconnected environment, it is possible to install the Satellite Server by using an ISO image. This method is not recommended for any other situation as ISO images may not contain the latest updates to Satellite; therefore, by installing Red Hat Satellite with an ISO Image you may be installing older versions of Satellite. Older versions may be missing bug fixes and functionality.

Prerequisites

Before installing, you must have a repository configured with Red Hat Enterprise Linux 6.6 and later or Red Hat Enterprise Linux 7.0 and later. For more information on how to update a disconnected system, in Red Hat Enterprise Linux 6 see [Upgrading the System Off-line with ISO and Yum](#) in [Deployment guide](#), and for Red Hat Enterprise Linux 7 see [Upgrading the System Off-line with ISO and Yum](#) in [System Administrator's Guide](#).

A copy of the repositories used in the installation are stored in the `/opt/` directory. Ensure you have a minimum of 2GB of space for this file system and directory.

ISO installations require imported Red Hat GPG keys before installation. Run the following command as root before running the installation script:

```
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

The following procedure details how to install Satellite Server on a host through ISO.

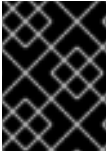
1. Download the ISO image from the Red Hat Customer Portal.
2. As the root user, mount the ISO image to a directory:

```
# mkdir /media/iso
# mount -o loop iso_filename /media/iso
```

3. Change to the `/media/iso` directory.

4. Run the installer script in the mounted directory:

```
# ./install_packages
```



IMPORTANT

The required packages are now installed. Proceed to [Section 2.2, “Running the Installation and Configuration Program”](#).

2.2. RUNNING THE INSTALLATION AND CONFIGURATION PROGRAM

Now that the required packages have been downloaded, the installation and configuration program, **katello-installer** must be run to install the Satellite Server. There are two main methods to do so:

- Manual Configuration - manually run the command and configuration options on the command-line interface (CLI).
- Automatic Configuration - most of the installation and configuration process can be automated by using an answer file.

Both methods are supported and available in this chapter. Choosing one or the other would depend on your organization's requirements.

Other configuration options are also documented in this chapter to assist in installing the Satellite Server. For example, if there is an HTTP Proxy in the host system's network, or if the organization uses customized server certificates.

2.2.1. Configuring Red Hat Satellite Manually

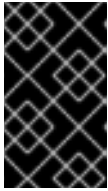
Satellite Server has an automatic initial configuration that prepares the server for use. The **katello-installer** script supports the ability to override various default settings within the different components of Satellite Server. For example, for organizations that have an existing HTTP proxy, additional configuration options need to be passed to the Satellite Server installer. See [Section 2.3, “Optional Configuration Options”](#) for other configuration options that can be used based on your environment's requirements.

Procedure 2.2. To Run the Installer Script:

1. Run the following command as the root user to manually configure Red Hat Satellite:

```
# katello-installer --foreman-initial-organization  
"initial_organization_name" \  
--foreman-initial-location "initial_location_name" \  
--foreman-admin-username admin-username \  
--foreman-admin-password admin-password
```

This script can be run multiple times without any issues.



IMPORTANT

If you do not specify any of these values, the default values are used. Use the **katello-installer --help** command to display the available options and any default values.

When the configuration script has completed successfully, it displays output similar to the following:

```
# katello-installer
Installing                               Done
[100%] [.....]
Success!
* Katello is running at https://satellite.example.com
  Default credentials are 'admin:changeme'
* Capsule is running at https://satellite.example.com:9090
* To install additional capsule on separate machine continue by
running:

capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
"~/ $CAPSULE-certs.tar"

The full log is at /var/log/katello-installer/katello-
installer.log
```

2. After configuration, run the following commands to configure the firewall to limit **elasticsearch** to the **foreman** and **root** users and make these rules persistent during reboots:

- On Red Hat Enterprise Linux 6, execute as root:

```
# iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner --
uid-owner \
foreman -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner -
uid-owner root -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

- On Red Hat Enterprise Linux 7, execute as root:

```
# firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner foreman -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner foreman -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner root -j ACCEPT \
```

```
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
foreman -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
foreman -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP
```

The Red Hat Satellite Server creates an initial organization and location called "Default Organization" and "Default Location", respectively. After the initial configuration, you can create additional organizations and locations. You can rename the default organization or location and you can delete the default organization, but you cannot delete the default location.

2.2.2. Configuring Red Hat Satellite with an Answer File

You can use *answer files* to automate installations with customized options. The initial answer file is sparsely populated. After you run **katello-installer** for the first time, the answer file is populated with the standard parameter values for installation.

The following procedure describes how to configure Red Hat Satellite Server with an answer file.

Procedure 2.3. To Configure and Use an Answer File for Installation:

1. Copy the default answer file located at **/etc/katello-installer/answers.katello-installer.yaml** to a location on your local file system:

```
# cp /etc/katello-installer/answers.katello-installer.yaml
/etc/katello-installer/my-answer-file.yaml
```

2. Open your copy of the answer file, edit the values to suit your environment, and save the file.



NOTE

The parameters for each module are specified in the module's **params.pp** file. Run the following command to view available modules with parameter files:

```
# rpm -ql katello-installer-base | grep params.pp
```

3. Open the **/etc/katello-installer/katello-installer.yaml** file and edit the answer file entry to point to your custom answer file:

```
:answer_file: /etc/katello-installer/my-answer-file.yaml
```

4. Run the **katello-installer** command.

```
# katello-installer
```

2.3. OPTIONAL CONFIGURATION OPTIONS

2.3.1. Configuring Red Hat Satellite with an HTTP Proxy

This section shows how to configure Red Hat Satellite for networks that go through an HTTP Proxy. As a prerequisite, make sure that the **http_proxy**, **https_proxy**, and **no_proxy** environment variables are not set:

```
# export http_proxy=""
# export https_proxy=$http_proxy
# export no_proxy=$http_proxy
```

Run **katello-installer** with the following options:

```
# katello-installer --katello-proxy-url=http://myproxy.example.com \
--katello-proxy-port=8080 \
--katello-proxy-username=proxy_username \
--katello-proxy-password=proxy_password
```

Where:

- **--katello-proxy-url** is the URL of the HTTP proxy server.
- **--katello-proxy-port** is the port the HTTP proxy server is listening on.
- **--katello-proxy-username** (optional) is the HTTP proxy username for authentication. If your HTTP proxy server does not require a username, you are not required to specify the username.
- **--katello-proxy-password** (optional) is the HTTP proxy password for authentication. If your HTTP proxy server does not require a password, you are not required to specify the password. The following list of special characters used in a password, as well as any whitespace, must be escaped using the back slash **** character: **] [? \ < ~ # ` ! @ \$ % ^ & * () + = } | : " ; ' , > { .** Alternatively, use quotation marks around the password.

After configuring the Satellite Server to go through the HTTP Proxy, make sure that **yum** or **subscription-manager** can connect to the Red Hat Content Delivery Network (CDN) and that the Satellite Server can synchronize its repositories to the CDN by following these steps:

Procedure 2.4. To Configure Satellite Server to Allow Red Hat Subscription Manager Access to the CDN:

1. On the network gateway and the HTTP Proxy, open the following hostnames, ports and protocols:

Table 2.1. Required Hostnames, Ports and Protocols

Hostname	Port	Protocol
subscription.rhn.redhat.com	443	https
cdn.redhat.com	443	https
*.akamaiedge.net	443	https

2. In the Satellite Server, complete the following details in the `/etc/rhsm/rhsm.conf` file. For example:

```
# an http proxy server to use (enter server FQDN)
proxy_hostname = http_proxy.example.com

# port for http proxy server
proxy_port = 3128

# user name for authenticating to an http proxy, if needed
proxy_user =

# password for basic http proxy auth, if needed
proxy_password =
```

2.3.2. Configuring Red Hat Satellite with a Custom Server Certificate

Red Hat Satellite comes with a default certificate authority (CA) used by both the server and client SSL certificates for authentication of subservices. The server and client certificates can be replaced with custom ones. For more information on creating custom certificates, see the [Red Hat Enterprise Linux 7 Security Guide](#) [6]

Custom server and client certificates may be implemented either before or after running the Katello installer. Implementing custom certificates *after* installation requires additional effort, so doing so *before* is recommended.



NOTE

The certificate's Common Name (CN) must match the fully qualified domain name of the server on which it is used.

Prerequisites

You must have the following files:

Certificate file for the Satellite Server, signed by your certificate authority (or self-signed)

Katello installer parameter `--certs-server-cert`. In this example, `satellite.crt`.

Certificate signing request file that was used to create the certificate for the Satellite Server

Katello installer parameter `--certs-server-cert-req`. In this example, `satellite.crt.req`.

Satellite Server's private key used to sign the certificate

Katello installer parameter `--certs-server-key`. In this example, `satellite.crt.key`.

CA certificate

Katello installer parameter `--certs-server-ca-cert`. In this example, `ca_cert.crt`.

If you have already run the Katello installer, see [Procedure 2.6, “To Set a Custom Server Certificate After Running the Katello Installer:”](#), otherwise see [Procedure 2.5, “To Set a Custom Server Certificate Before Running the Katello Installer:”](#).

Procedure 2.5. To Set a Custom Server Certificate Before Running the Katello Installer:



NOTE

In this example the files are stored in the directory `/root/sat_cert`. Using an absolute path in the **root** users' directory provides a fixed location that is available to all users who log in to the server with **root** permissions. Before running this command, ensure the directory already exists.

- Run the following command on the Red Hat Satellite Server to use the custom certificate.

```
# katello-installer \
--certs-server-cert /root/sat_cert/satellite.crt \
--certs-server-cert-req /root/sat_cert/satellite.crt.req \
--certs-server-key /root/sat_cert/satellite.crt.key \
--certs-server-ca-cert /root/sat_cert/ca_cert.crt
```



IMPORTANT

If you configure a Satellite Server to use custom certificates, you must do the same for all Capsule Servers. For instructions see [Section 7.5.1, “Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate”](#)

Procedure 2.6. To Set a Custom Server Certificate After Running the Katello Installer:

When the Katello installer is run for the first time without certificate parameters, it uses the default CA to sign both server and client certificates. To enforce custom certificates deployment after the Katello installer is first run, the certificates installed must be updated.



NOTE

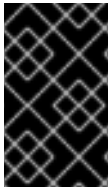
In this example the files are stored in the directory `/root/sat_cert`. Using an absolute path in the **root** users' directory provides a fixed location that is available to all users who log in to the server with **root** permissions. Before running this command, ensure the directory already exists.

1. Run the following command on the Red Hat Satellite Server to regenerate the `katello-ca-consumer` package and the Satellite Server's certificate.

```
# katello-installer \
--certs-server-cert /root/sat_cert/satellite.crt \
--certs-server-cert-req /root/sat_cert/satellite.crt.req \
--certs-server-key /root/sat_cert/private.crt.key \
--certs-server-ca-cert /root/sat_cert/ca_cert.crt \
--certs-update-server \
--certs-update-server-ca \
--certs-update-all
```

2. Run the following command on the client systems to install the new client and server certificates.

```
# rpm -Uvh http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```



IMPORTANT

If you configure a Satellite Server to use custom certificates, you must do the same for all Capsule Servers. For instructions see [Section 7.5.1, “Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate”](#).

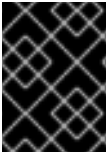
2.3.3. Configuring DNS, DHCP, and TFTP

This section describes how to configure Satellite to run BIND (**named**) to provide authoritative DNS services for the **example.com** domain and the 172.17.13.x subnet. This requires setting up a DNS zone for forward lookups, which will be contained in the **example.com** zone file. Additionally, a DNS zone for reverse lookups will be created for the 172.17.13.x subnet, which will be contained in the **13.17.172.in-addr.arpa** reverse zone file. This ensures that hosts provisioned from Satellite use the correct name resolution parameters. This section also describes how to configure the TFTP proxy so that hosts can boot using PXE.

Clients on this network will have the following characteristics:

- Have access to IP addresses in the range 172.17.13.100 to 172.17.13.150 for DHCP.
- Use the Satellite (**satellite.example.com** at 172.17.13.2) for DNS.
- Receive a **pxelinux.0** file from Satellite (**satellite.example.com** at 172.17.13.2) to enable PXE-booting.

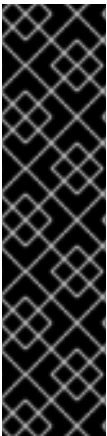
- Have host names of **hostname.example.com**, where *hostname* is configured when the host is provisioned.



IMPORTANT

This example enables DHCP services on the Satellite server. Consult your network administrator before proceeding.

Run the following **katello-installer** command as root, using the specified options to configure the required services on the Satellite server. Remember to substitute your desired administrator user name and password.



IMPORTANT

- If you have created an admin user and password by running **katello-installer** previously, do not include the **--foreman-admin-username** and **--foreman-admin-password** options in the following command.
- If you do not specify the administrator user name and password, the default user **admin** is created, and the password is automatically generated. The credentials are displayed at the end of the installation process. Make a note of this password. You can also retrieve the password from **admin_password** parameter in the **/etc/katello-installer/answers.katello-installer.yaml** file.

```
# katello-installer --foreman-admin-username admin-username \
--foreman-admin-password admin-password \
--capsule-dns true \
--capsule-dns-interface eth0 \
--capsule-dns-zone example.com \
--capsule-dns-forwarders 172.17.13.1 \
--capsule-dns-reverse 13.17.172.in-addr.arpa \
--capsule-dhcp true \
--capsule-dhcp-interface eth0 \
--capsule-dhcp-range "172.17.13.100 172.17.13.150" \
--capsule-dhcp-gateway 172.17.13.1 \
--capsule-dhcp-nameservers 172.17.13.2 \
--capsule-tftp true \
--capsule-tftp-servername $(hostname) \
--capsule-puppet true \
--capsule-puppetca true
```

At the end of the installation process, **katello-installer** outputs the status of the installation.

```
Success!
* Katello is running at https://satellite.example.com
  Default credentials are 'admin:*****'
* Capsule is running at https://satellite.example.com:9090
* To install additional capsule on separate machine continue by
running:"

capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
```

```
"~/ $CAPSULE-certs.tar"
```

The full log is at `/var/log/katello-installer/katello-installer.log`

Use a web browser to navigate to <https://satellite.example.com> to display the Satellite home page. This example uses the default organization (Default_Organization) and the default location.

Alternatively, you can configure Satellite to use external DNS and DHCP services as described in [Section 7.9, “Configuring Satellite 6 with External Services”](#). If required to allocate specific IP addresses to host names or MAC addresses, see the DHCP chapter in the [Red Hat Enterprise Linux 7 Networking Guide](#)^[7].

2.3.3.1. Additional DNS, DHCP and TFTP Options

The following table describes the various options and the values required to correctly configure the Satellite server. The **katello-installer** command uses Puppet; consequently, it will install additional packages (bind, dhcp, xinetd, and so on) and configure them to add the requested functionality.

For a complete list of available options, run **katello-installer --help**.

Table 2.2. Satellite Configuration Options

Option	Description	Value
<code>--foreman-admin-username</code>	The user name for the initial administrator.	User specified.
<code>--foreman-admin-password</code>	The password for the initial administrator.	User specified.
<code>--capsule-dns</code>	Enable DNS proxy capability	yes
<code>--capsule-dns-interface</code>	Which interface named should listen on	eth0
<code>--capsule-dns-zone</code>	The Forward DNS zone that the Satellite will host	example.com
<code>--capsule-dns-forwarders</code>	The DNS server that unknown queries are forwarded to	172.17.13.1
<code>--capsule-dns-reverse</code>	The Reverse DNS zone the Satellite hosts. This is usually the first three octets of the IP address (172.17.13) reversed , and appended with ".in-addr.arpa".	13.17.172.in-addr.arpa
<code>--capsule-dhcp</code>	Enable DHCP proxy capability	yes
<code>--capsule-dhcp-interface</code>	The interface that DHCP listens on	eth0

Option	Description	Value
--capsule-dhcp-range	The range of IP addresses to issue to clients.	172.17.13.100 172.17.13.150
--capsule-dhcp-gateway	The default gateway IP to issue to clients.	172.17.13.1
--capsule-dhcp-nameservers	The host that the clients should use for name resolution. This should be configured with the Satellite's IP in this deployment model.	172.17.13.2
--capsule-tftp	Enable TFTP proxy capability. This is needed to PXE boot the clients.	yes
--capsule-tftp-servername	Sets the TFTP host name. Set this to match the server's host name (satellite.example.com).	\$(hostname)
--capsule-puppet	Enable the Puppet Master.	yes
--capsule-puppetca	Enable the Puppet CA.	yes

[6] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_OpenSSL.html

[7] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/

CHAPTER 3. LOGGING IN TO RED HAT SATELLITE

After Red Hat Satellite has been installed and configured use the web user interface to log in to Satellite for further configuration.

These steps show how to log in to Red Hat Satellite.

1. Access the Satellite server web UI using a web browser using the host name or FQDN:

https://host_name/

To identify the Satellite servers host name, use the **hostname** command on the Satellite server. Add the **-f** option to display the FQDN:

```
# hostname -f
```

IMPORTANT

An untrusted connection warning appears on your web browser when accessing Satellite for the first time. Accept the self-signed certificate and add the Satellite URL as a security exception to override the settings. This procedure might differ depending on the browser being used.

Only do this if you are sure that the Satellite URL is a trusted source.

2. Enter the user name and password created during the configuration process. If a user was not created during the configuration process, the default user name is **admin**.

NOTE

If you have forgotten the administrative password, use the Satellite command-line interface to reset the administration user and password:

```
# foreman-rake permissions:reset  
Reset to user: admin, password: qwJxBptxb7Gfcjj5
```

This will reset the password of the default user to the one printed on the command line. Change this password upon logging in to prevent any security issues from occurring.

3.1. ORGANIZATIONS

Organizations divide hosts into logical groups based on ownership, purpose, content, security level, or other divisions.

Multiple organizations can be viewed, created, and managed within the web interface. Software and host entitlements can be allocated across many organizations, and access to those organizations controlled.

Each organization must be created and used by a single Red Hat customer account,

however each account can manage multiple organizations. Subscription manifests can only be imported into a single organization and Satellite will not upload a certificate that has already been uploaded into a different organization.

By default, **Red Hat Satellite** will have one organization already created, called "Default Organization", which can be modified to suit your own installation, or deleted. The organization name has a corresponding label **Default_Organization** for use on the command line.



IMPORTANT

If a new user is not assigned a default organization their access will be limited. To grant the user systems rights, assign them a default organization and have them log out and log back in again.

3.1.1. Creating an Organization

These steps show how to create a new organization.

Procedure 3.1. Creating an Organization

1. Click **Administer** → **Organizations**.
2. Click **New Organization**.
3. Specify the name of the new organization in the **Name** field. Take care not to add an extra space at the end of the name as this will affect the corresponding label created.
4. In the **Label** field, optionally enter a text string similar to the name but without spaces. If omitted, a label to match the name of the new organization, but with underscores in place of spaces, is created automatically. The label is for use on the command line and cannot be changed once this procedure has been completed. Having a consistent name to label correspondence will reduce errors on the command line. Consider creating names without spaces.
5. Enter a description of the new organization in the **Description** field.
6. Click **Submit**.
7. Select the hosts to assign to the new organization.
 - Click **Assign All** to assign all hosts with no organization to the new organization.
 - Click **Manually Assign** to manually select and assign the hosts with no organization.
 - Click **Proceed to Edit** to skip assigning hosts.

3.1.2. Editing an Organization

You can update your organization information as required. You cannot change the organization label.


Procedure 3.2. Editing an Organization

1. Click **Administer** → **Organizations**.
2. Click the name of the organization you want to edit.
3. Select the resource to edit.
4. Click the name of the desired items to add them to the **Selected Items** list.
5. Click **Submit**.

3.1.3. Removing an Organization

Procedure 3.3. Removing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Select **Delete** from the drop down menu to the right of the name of the organization you want to remove.
3. An alert box appears:

 Delete *Organization Name*?

4. Click the **OK** button.

Result

The organization is removed from **Red Hat Satellite**.

3.2. CHANGING YOUR ACCOUNT PREFERENCES

Setting up default account preferences ensures that subsequent logins will enable the correct context within the Red Hat Satellite Server for a specific user. It also allows changes in user preferences.

The following preferences can be changed:

1. **User** - Change personal data about your login name, as well as your password and default location/organization.
 1. First Name
 2. Surname
 3. Email Address
 4. Default Location
 5. Default Organization
 6. Password
2. **Locations** - Add or remove locations on your account based on the locations created within the Red Hat Satellite Server.

3. **Organizations** - Add or remove organizations on your user account based on the organizations created within the Red Hat Satellite Server.
4. **Roles** - Add or remove roles on your user account based on a set of roles created within the Red Hat Satellite Server.

Procedure 3.4. Changing your Account Preferences

To change these preferences:

1. At the upper right corner, hover your mouse over the **admin** user and on the drop-down menu that appears, click on **My Account**.
2. Choose the subtab of the preference you wish to change and click on the subtab.
3. Change the preferences you wish to change and click on **Submit**.



NOTE

Set your default location/organization in the **User** subtab after your initial login. This will make sure that subsequent logins will set you in the correct context for your user.

3.3. ADDITIONAL RESOURCES

For more information on configuring users in Red Hat Satellite, see the resources listed below.

- The [Users and Roles](#) chapter in the [Red Hat Satellite 6.1 User Guide](#) describes creating users and their roles.
- The [Configuring External Authentication](#) chapter in the [Red Hat Satellite 6.1 User Guide](#) describes using external authentication sources, such as LDAP or *Red Hat Enterprise Linux Identity Management (IdM)*, to derive user and user group permissions.

CHAPTER 4. POPULATING RED HAT SATELLITE WITH CONTENT

Red Hat Satellite provides multiple types of content to subscribed client hosts including software packages, errata, Puppet modules, and container images.

The primary source of this content is the Red Hat Customer Portal, in order to access it, you need to upload a *subscription manifest* file to the Satellite server. A subscription manifest provides subscriptions to client hosts through the Red Hat Satellite rather than through Red Hat Network. Obtain the subscription manifest file from the Red Hat Customer Portal as described in [Section 4.1.1.1, “Creating a Subscription Manifest”](#), or by contacting Red Hat Support.

This chapter outlines the process of populating your Red Hat Satellite server with content. Some of the following procedures are not needed frequently and are usually performed only once after installation. Others, like [Section 4.1.3, “Synchronizing Content”](#) must be repeated regularly to keep the content up to date.

The steps required to get the content from Red Hat Customer Portal to the Satellite Server depend on the type of deployment:

- If your Satellite server can access the Internet directly, see [Section 4.1, “Connected Satellite”](#).
- If your Satellite server is isolated from the Internet, see [Section 4.2, “Disconnected Satellite”](#).

4.1. CONNECTED SATELLITE

A connected Satellite server has access to the Internet and therefore can download software packages, errata, Puppet modules, and container images directly from the Red Hat Customer Portal.

4.1.1. Accessing Red Hat Content Providers

This section outlines the steps required to receive content from Red Hat Customer Portal. First create a manifest on the Red Hat Customer Portal, then upload it to the Satellite server, and enable Red Hat repositories.

4.1.1.1. Creating a Subscription Manifest

A *subscription manifest* can be obtained through the method below or by contacting Red Hat Support. The manifest is used to set up Red Hat content providers and contains repository information and subscriptions. It is used as a basis of dispensing subscriptions and Red Hat Network (RHN) content to client systems from **Red Hat Satellite**.



IMPORTANT

Manifests are organization-specific, which means you have to create and upload a separate manifest for every organization on your Satellite.

Prerequisites

You must meet the following conditions before continuing with this task:

- A Customer Portal user name and password.
- Sufficient subscriptions to add to the manifest.

Procedure 4.1. To Create a Manifest for Satellite 6:

1. Navigate to <https://access.redhat.com> and click **SUBSCRIPTIONS** on the main menu at the top of the page.
2. Scroll down to the **Red Hat Subscription Management** section, and click **Satellite** under **Subscription Management Applications**.
3. To create a manifest for a new system, click **Register a Satellite**. Select the **Satellite version** and **Name** that must match the name of the organization on your Satellite. Click **Register**.

To add or modify subscriptions of an existing manifest, click the name of the system this manifest is associated to, and click **Attach a subscription**.

4. For each subscription that you want to attach, select the check box for that subscription, and specify the quantity of subscriptions to attach.
5. Click **Attach Selected**.



NOTE

It can take several minutes for all the subscriptions to attach. Refresh the screen every few minutes until you receive confirmation that the subscriptions are attached.

6. After the subscriptions have been attached, click **Download Manifest** to generate an archive in .zip format containing the manifest for Red Hat Satellite.

4.1.1.2. Uploading a Subscription Manifest to Satellite

This section describes how to upload a subscription manifest to an organization. Because subscription manifests are organization-specific, ensure you select the correct organization before you try to upload a subscription manifest. Failing to do so will cause a permission denied error (Error 403).

Procedure 4.2. To Upload a Subscription Manifest:

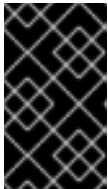
1. Log in to the **Satellite** server and select the desired organization from the menu in the top left hand corner.
2. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest** at the upper right of the page.
3. In the **Subscription Manifest** section, click **Actions** and under the **Upload New Manifest** subsection, click **Browse**.
4. Select the manifest file to upload, and then click **Upload**.

4.1.1.3. Enabling Red Hat Repositories

The Red Hat Satellite manifest file provides access to Red Hat products and repositories. Because most products have several architectures and product versions, Red Hat Satellite Server allows the Satellite administrators to choose which repositories are required by their organizations. You need to enable the repositories in Red Hat Satellite Server to prepare them for synchronization.

Procedure 4.3. To Enable Red Hat Repositories:

1. On the main menu, click **Content** → **Red Hat Repositories** and then click the tab for the type of content that you want to enable.
2. Click the product name for which you want to add repositories. This expands the list of available repository sets.
3. Click each repository set from which you want to select repositories, and select the check box for each required repository. The repository is automatically enabled. The content from this repository will be downloaded during the next synchronization, see [Section 4.1.3, “Synchronizing Content”](#). After enabling a Red Hat repository, a product for this repository is automatically created.



IMPORTANT

Ensure you enable the Satellite Tools repository. This repository provides the katello-agent and puppet-agent packages for clients registered to the Satellite Server.

The following is an example set of subscriptions that contain repositories with the latest packages for Red Hat Enterprise Linux 6:

- Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server Repository
- Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server Repository
- Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64 Repository

4.1.2. Using Products

A *product* is a group of related repositories that acts as the smallest unit of the synchronization process. Products ensure that repositories that depend on each other are synchronized together. For Red Hat repositories, products are created automatically after enabling the repository. Therefore, you only need to create products manually for repositories with custom or third-party content.

4.1.2.1. Creating a Product

These steps show how to create a new product.

Procedure 4.4. To Create a Product:

1. Click **Content** → **Products**.
2. Click **New Product**.
3. Specify the name of the new product in the **Name** field.

4. Specify the label for the new product in the **Label** field.
5. Select a GPG key from the **GPG Key** drop-down menu.
6. Select a synchronization plan from the **Sync Plan** drop-down menu. You can also select the **New Sync Plan** link to create a new synchronization plan.
7. Enter a description of the new product in the **Description** field.
8. Click **Save**.

4.1.2.2. Adding Repositories to a Product

These steps show how to add repositories to a product in **Red Hat Satellite**.

Procedure 4.5. To Add Repositories to a Product:

1. Click **Content** → **Products**.
2. Click the product to add a repository.
3. Click **Repositories**.
4. Click **Create Repository**.
5. Specify the name of the new repository in the **Name** field.
6. Specify a label for the new repository in the **Label** field.
7. Select the type of the repository from the **Type** drop-down menu.
8. Specify the URL of the repository in the **URL** field.
9. Choose whether to publish the repository via HTTP by selecting **Publish via HTTP**.
10. Select a GPG key for the repository from the **GPG Key** drop-down menu.
11. Click **Create**.

4.1.2.3. Using Bulk Actions for Products

This section describes how to use bulk actions to synchronize or remove products in Red Hat Satellite. The procedure described here requires that at least one product be available.

Procedure 4.6. To Synchronize Multiple Products:

1. Navigate to **Content** → **Products**.
2. Select the check box for the products you want to work with.
3. Click **Bulk Actions**.
4. Click the **Product Sync** tab and then click **Sync Now**.

Procedure 4.7. To Remove Multiple Products:

1. Navigate to **Content → Products**.
2. Select the check box for the products you want to work with.
3. Click **Bulk Actions**.
4. Click **Remove Products** and then click **Remove**.

Procedure 4.8. To Update Synchronization Plans for Multiple Products:

1. Navigate to **Content → Products**.
2. Select the check box for the products you want to work with.
3. Click **Bulk Actions**.
4. Click the **Alter Sync Plans** tab. Depending on the type of action you want to perform select from the following alternatives.
 - To create a new synchronization plan, click **Create Sync Plan**. Specify the required details and click **Save**.
 - To remove the synchronization plans from the selected products, click **Unattach Sync Plan**.
 - To update the synchronization plans for the selected products, click **Update Sync Plan**.

4.1.2.4. Using Repository Discovery

Repository discovery enables you to search using a URL to discover repositories available to include in a product.

Procedure 4.9. To Use Repository Discovery:

1. Navigate to **Content → Products**.
2. Click **Repo Discovery**.
3. Insert the URL where the repositories are located in the **Yum Repo Discovery** field.
4. Click **Discover**.

A list of the repositories at the URL is displayed under **Results**.

5. Click **Discovered URLs** to add the repositories to the product.
6. Click **Create selected**.
7. Choose whether to add the repositories to an existing product or create a new product.
 - a. To add the repositories to an existing product:
 - i. Select **Existing Product**.
 - ii. Select the required product from the drop-down menu.

- b. To create a new product to add the repositories to:
 - i. Select **New Product**.
 - ii. Enter the **Name** and **Label** for the new product and select a **GPG Key** from the drop-down menu.
8. Select **Serve via HTTP** to serve the repository via HTTP.
9. Edit the **Name** and **Label** for the **Selected URLs**.
10. Click **Create**.

4.1.2.5. Removing a Product

This section describes how to remove products from Red Hat Satellite.

Procedure 4.10. To Remove a Product from Satellite:

1. Navigate to **Content → Products**.
2. Select the check box next to the products you want to remove.
3. Click **Bulk Actions** and then click **Remove Products**.
4. Click **Remove** to confirm that you want to remove the products.

4.1.3. Synchronizing Content

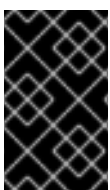
Synchronization is the act of coordinating updates between the Red Hat Satellite repositories and the source repositories being used. It is a required step after enabling repositories, in order to populate the Red Hat Satellite with content from the source repositories.

Constant, scheduled synchronization will result in:

- Data integrity between packages
- Updated packages, security fixes, and errata

Red Hat Satellite's synchronization management capabilities allows an organization's administrators to create synchronization plans to configure how often a host should look for and install updates. Synchronization plans are then paired with the product repositories to specify a synchronization schedule that will allow products to be updated at specific intervals that are convenient for the organization's network.

4.1.3.1. Synchronization Status



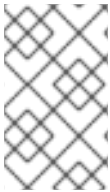
IMPORTANT

The manual synchronization of repositories is required after enabling them. It is at this point that the local repository in the Satellite is populated by the required packages.

These steps show how to synchronize products in Red Hat Satellite.

Procedure 4.11. Synchronize Products

1. Navigate to **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product repositories available for synchronization is displayed.
2. Click the arrow next to the product name to see available content.
3. Select the content you want to synchronize.
4. Click **Synchronize Now** to starting synchronizing. The status of the synchronization process will appear in the **Result** column. If synchronization is successful, **Sync complete** will appear in the **Result** column. If synchronization failed, **Error syncing** will appear.



NOTE

Content synchronization can take a long time. The length of time required depends on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

4.1.3.2. Creating a Synchronization Plan

Regular, frequent synchronization is required to maintain data integrity between packages as well as making sure that packages are updated to the latest security fixes. Red Hat Satellite provides the ability to create scheduled synchronization plans that allow package updates at intervals convenient to the organization.

Procedure 4.12. To Create a Synchronization Plan:

1. Navigate to **Content** → **Sync Plans**.
2. Click **New Sync Plan** to create a new synchronization plan.
3. Specify the **Name**, **Description**, **Interval** and **Start Date** for the plan.
4. Click **Save**.

After creating a synchronization plan, select products that will be synchronized according to this plan as described in [Section 4.1.3.3, “Applying a Synchronization Schedule”](#).



NOTE

Synchronization plans are applied per product, therefore all repositories associated with the product are synchronized with the same frequency. It is not possible to set different synchronization intervals for repositories in the same product.

4.1.3.3. Applying a Synchronization Schedule

After you have created a synchronization plan, you need to associate products with that plan to create a synchronization schedule. The following procedure describes how to create a synchronization schedule in Red Hat Satellite 6.

Procedure 4.13. To Create a Synchronization Schedule:

1. Click **Content** → **Sync Plans** and select the synchronization plan you want to implement.
2. Click **Products** → **Add** in the synchronization plan main page.
3. Select the check box of the product to associate with the synchronization plan.
4. Click **Add Selected**.

4.1.4. Using a Content ISO for Initial Synchronization

Even if the Satellite Server can connect directly to the Red Hat Customer Portal, you can perform the initial synchronization from a locally mounted content ISO. Such synchronization is not limited by network bandwidth and therefore is usually faster, especially when synchronizing large repositories for the first time. Once the initial synchronization is completed from the content ISO, you can switch back to downloading content through the network connection.

A connection to the Red Hat Customer Portal is required for downloading repository metadata. The following example uses the Red Hat Satellite content ISO, but you can also use the exported content from **katello-disconnected** (see [Section 4.2.2, “Using the Synchronization Host”](#)).

Example 4.1. Synchronizing a Repository from a Local Source

This example shows how to perform the first synchronization of the Red Hat Enterprise Linux 6 repository from a content ISO.

1. Download the content ISO for Red Hat Enterprise Linux 6 from the Red Hat Customer Portal (see [Section 4.2.1, “Using Content ISO”](#) for detailed instructions). Copy the content ISO to your Satellite server, for example to the **/root/isos/** directory.
2. On the Satellite server, create a mount point, mount the ISO and copy its content to a writable directory that Satellite can access, in this example **/mnt/rhel6/**:

```
# mkdir /mnt/iso
# mount -o loop /root/isos/sat-6-isos--rhel-6-server-x86_64.iso
/mnt/iso
# cp -ruv /mnt/iso/ /mnt/rhel6/
```

Then unmount the ISO and remove the mount point:

```
# umount /mnt/iso
# rmdir /mnt/iso
```

3. Set the correct SELinux context and ownership for the content directory:

```
# chcon -R --type=httpd_sys_rw_content_t /mnt/rhel6/
# chown -R apache:apache /mnt/rhel6/
```

4. Create or edit the **/etc/pulp/content/sources/conf.d/local.conf** file. Insert the following text to the file:

```
[rhel-6-server]
enabled: 1
priority: 0
expires: 3d
name: Red Hat Enterprise Linux 6 Server
type: yum
base_url:
file:///mnt/rhel6/content/dist/rhel/server/6/6Server/x86_64/os/
```

The **base_url** path may differ in your content ISO. The directory specified in **base_url** must contain the **repodata** directory, otherwise the synchronization will fail. To synchronize multiple repositories, create a separate entry for each of them in the configuration file **/etc/pulp/content/sources/conf.d/local.conf**.

5. In the Satellite web UI, navigate to **Content → Red Hat Repositories** and select the repository to be enabled, in this example *Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server*.

Under **Content → Sync Status** select the repository to be synchronized and click **Synchronize Now**.

Note that there is no indication in the Satellite web UI of which source is being used. In case of problems with a local source, Satellite pulls content through the network. To monitor the process, run the following command in the console on Satellite (limited to Red Hat Enterprise Linux 7 base systems):

```
# journalctl -f -l SYSLOG_IDENTIFIER=pulp | grep -v worker[\\-
,\\. ]heartbeat
```

The above command displays interactive logs. First, the Satellite server connects to the Red Hat Customer Portal to download and process repository metadata. Then, the local repository is loaded. In case of any errors, cancel the synchronization in the Satellite web UI and verify your configuration.

6. After successful synchronization you can detach the local source by removing its entry from **/etc/pulp/content/sources/conf.d/local.conf**.

4.2. DISCONNECTED SATELLITE

In high security environments where hosts are required to function in a closed network disconnected from the Internet, the Red Hat Satellite can provision systems with the latest security updates, errata, and packages. The recommended way to populate a disconnected Satellite with content is by using an ISO file downloaded from the Red Hat Customer Portal. Alternatively, you can configure a synchronization host.

4.2.1. Using Content ISO

The following procedure shows how to use the content ISO to add content to Red Hat Satellite.

1. Download the product ISO from the Red Hat Customer Portal, as follows:
 - a. Go to **Downloads** (at the very top of the window) and select **Red Hat Satellite**.

- b. Open the **Content ISOs** tab. All products to which the account is subscribed are listed there.
 - c. Click the link for the product name, such as Red Hat Enterprise Linux 6 Server (x86_64)(2015-03-12) to download the ISO.
 - d. Save to media.
2. Copy all of the Satellite content ISOs to a directory that Satellite can access. This example uses `/root/isos`.
3. Create a local directory that will be shared via **httpd** on the Satellite. This example uses `/var/www/html/pub/sat-import/`.

```
# mkdir -p /var/www/html/pub/sat-import/
```

4. Recursively copy the contents of the first ISO to the local directory:

```
# mkdir /mnt/iso
# mount -o loop /root/isos/first_iso /mnt/iso
# cp -ruv /mnt/iso/* /var/www/html/pub/sat-import/
# umount /mnt/iso
# rmdir /mnt/iso
```

5. Repeat the above step for each ISO until you have copied all the data from the series of ISOs into the local directory `/var/www/html/pub/sat-import/`.
6. Ensure that the SELinux contexts are correct:

```
# restorecon -rv /var/www/html/pub/sat-import/
```

7. Modify the default provider URL the Satellite web interface:

- a. Log in to the Satellite web interface.
- b. Select the required organization from the **Organization** menu.
- c. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest**.
- d. On the **Subscription Manifest** information screen select the **Actions** tab. Under **Red Hat Provider Details** click the edit icon on the **Red Hat CDN URL** entry and change it to the Satellite host name with the newly created directory, for example:

```
http://server.example.com/pub/sat-import/
```

Click **Save**.

- e. Click **Browse** to choose the manifest file.
- f. Click **Upload** to import your manifest.

**NOTE**

The Satellite is now acting as its own CDN with the files located in **`http://localhost`**. This is not a requirement. The CDN can be hosted on a different machine inside the same disconnected network as long as it is accessible to the Satellite server via HTTP.

8. To enable the repositories from the local CDN, click **Content → Red Hat Repositories**
9. Click **Content → Sync Status**.
10. Select the repositories you want to synchronize and click **Synchronize Now**.

Once the synchronize finishes, the disconnected Satellite is now ready to serve the content to hosts.

4.2.2. Using the Synchronization Host

**IMPORTANT**

The synchronization host feature is planned to be deprecated in future releases of Red Hat Satellite. Therefore, it is recommended to use the procedure described in [Section 4.2.1, “Using Content ISO”](#)

The diagram below illustrates how a disconnected Satellite is able to keep its content updated even without an Internet connection. An intermediary system with an Internet connection is needed to act as a synchronization host. This synchronization host is in a separate network from the Satellite server.

The synchronization host imports content from the Red Hat Content Delivery Network (CDN) through pulp. The content is then exported onto a media, such as DVDs, CDs, or external hard drives and transferred to the disconnected Satellite server. The following sections in this chapter will guide you through the whole process.

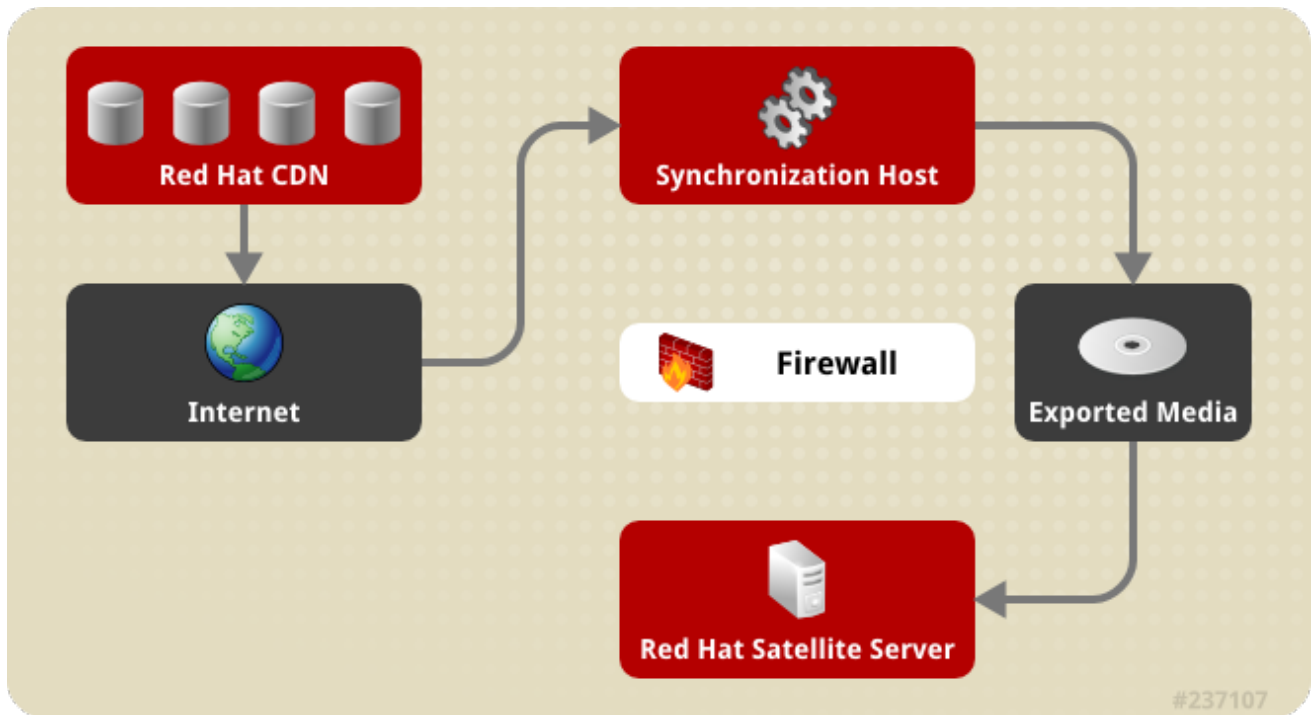


Figure 4.1. Disconnected Satellite

4.2.2.1. Configuring the Synchronization Host

The following section shows how to configure the synchronization host.

Prerequisites

To import content from the Red Hat Content Distribution Network (CDN), the synchronization host requires:

- An Internet connection
- Valid Red Hat Network subscriptions
- A valid manifest (See [Section 4.1.1.1, “Creating a Subscription Manifest”](#) for instructions on how to obtain one.)

Procedure 4.14. To Configure a Host to Synchronize and Export Content from the Red Hat CDN:

1. Use Red Hat Subscription Manager to register the synchronization host to RHN.
2. List all the available subscriptions to find the correct Red Hat Satellite product to allocate to your system:

```
# subscription-manager list --available --all
```

This command displays output similar to the following:

```
+-----+
| Available Subscriptions |
+-----+
```

```
ProductName:      Red Hat Satellite
ProductId:        SKU123456
PoolId:           e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Quantity:         10
Multi-Entitlement: No
Expires:          08/20/2013
MachineType:      physical
```

**NOTE**

The Product ID and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

3. Subscribe to the required pool IDs:

```
# subscription-manager subscribe \
--pool=Red_Hat_Satellite_Pool_ID \
--pool=Red_Hat_Enterprise_Linux_Pool_ID \
--pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_ID
```

4. Disable all existing repositories:

```
# subscription-manager repos --disable "*" 
```

5. Enable the Red Hat Satellite and Red Hat Enterprise Linux and Red Hat Software Collections repositories. Ensure the Red Hat Enterprise Linux repository matches the specific version you are using.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc1-6-rpms \
--enable rhel-6-server-satellite-6.1-rpms
```

**NOTE**

The commands above are based on Red Hat Enterprise Linux 6. If you are using a different version of Red Hat Enterprise Linux, change the repository based on your specific version.

6. Install katello-utils:

```
# yum install katello-utils
```

katello-utils includes the **katello-disconnected** utility that is required to set up repositories for import while qpid related packages are necessary for **pulp** configuration.

7. Generate a 32-character alphanumeric string for the **oauth_secret** entry in the **/etc/pulp/server.conf** file:

```
$ tr -dc "[:alnum:]" < /dev/urandom | head -c 32
```

8. In the `/etc/pulp/server.conf`, uncomment the `[oauth]` entry and add the randomly-generated value from the previous step as the `oauth_secret` value:

```
[oauth]
enabled: true
oauth_key: katello
oauth_secret: v8SeYqvS5QUfmg0dIrJOBG58LAHDRZnN
```

9. Disable authentication in `/etc/qpid/qpid.conf`:

```
# Configuration file for qpid. Entries are of the form:
#   name=value
#
# (Note: no spaces on either side of '=').
# Run "qpid --help" or see "man qpid" for more details.

auth=no
```

All incoming connections authenticate using the Satellite's default realm.

10. Configure the connection from **katello-disconnected** to Pulp with the previously generated value as your `--oauth-secret` option:

```
# katello-disconnected setup --oauth-key=katello --oauth-
secret=v8SeYqvS5QUfmg0dIrJOBG58LAHDRZnN
```

This places a configuration value in `~/.katello-disconnected`.

11. Configure Pulp on the synchronization server:

```
sudo service qpid start
sudo chkconfig qpid on
sudo service mongod start
sleep 10
sudo chkconfig mongod on
sudo -u apache pulp-manage-db
sudo service httpd restart
sudo chkconfig httpd on
sudo chkconfig pulp_workers on
sudo service pulp_workers start
sudo chkconfig pulp_celerybeat on
sudo service pulp_celerybeat start
sudo chkconfig pulp_resource_manager on
sudo service pulp_resource_manager start
```

12. Import the manifest to set up the list of available repositories to synchronize based on the selected subscriptions:

```
# katello-disconnected import -m ./manifest.zip
```

The synchronization host is now ready to synchronize content from the Red Hat CDN.

4.2.2.2. Synchronizing Content

By default, **katello-disconnected** enables all repositories that are included in the manifest for synchronization. Synchronization time is directly related to the amount of repositories to be synchronized. If the manifest has a large amount of repositories, the synchronization will take time and network resources.

katello-disconnected allows for the synchronization of specific repositories. This section will set up Pulp for synchronizing content.

1. Disable all repositories:

```
# katello-disconnected disable --all
```

katello-disconnected enables all repositories by default.

2. Choose which repositories you wish to sync by listing all available repositories from the manifest:

```
# katello-disconnected list --disabled
rhel-6-server-rhn-tools-rpms-6_6-x86_64
rhel-6-server-rhn-tools-rpms-6Server-x86_64
rhel-6-server-kickstart-6Server-x86_64
rhel-6-server-kickstart-6_6-x86_64
rhel-6-server-rh-common-rpms-6_6-x86_64
rhel-6-server-rpms-6_6-x86_64
```

3. Enable the chosen repositories for synchronization:

```
# katello-disconnected enable -r rhel-6-server-rh-common-rpms-6_6-
x86_64
```

4. Create the repositories and push them to **Pulp** to allow synchronization:

```
# katello-disconnected configure
```



NOTE

The configure option for **katello-disconnected** reads the manifest, creates pulp repositories, and generates scripts before synchronization. It needs to be run each time a repository is enabled or disabled.

5. Synchronize the repositories:

```
# katello-disconnected sync
```

You can use the **watch** option to monitor the synchronization process.

```
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-rh-common-rpms-6_6-x86_64

running:
rhel-6-server-rh-common-rpms-6_6-x86_64
```



```
...
finished:
rhel-6-server-rh-common-rpms-6_6-x86_64
```

```
Watching finished
```

4.2.2.3. Exporting Content

The synchronized content needs to be exported to enable importing into the disconnected Red Hat Satellite. An external export media such as a CD, DVD, or external hard drive is required for this procedure. Perform the following steps:

1. Export the synchronized repositories:

```
# katello-disconnected export -t /var/tmp/export
```

You can use the **watch** option to monitor the synchronization process. The output will look similar to:

```
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-rh-common-rpms-6_6-x86_64

finished:
rhel-6-server-rh-common-rpms-6_6-x86_64
Watching finished
Done watching ...
  Copying content to /var/tmp/export
  Archiving contents of /var/tmp/export into 4600M tar archives.
  NOTE: This may take a while.
tar: Removing leading `/' from member names

Done exporting content, please copy /var/tmp/export/* to your
disconnected host
```

This operation will create the following files in **/var/tmp/export**:

```
# ls /var/tmp/export/
content-export-00 content-export-01 content-export-02
expand_export.sh
```

2. Copy the files from **/var/tmp/export** to the external media.



NOTE

If the files are too big for your external media, the files can be copied sequentially in a series of DVDs.

The synchronized content has now been exported and ready for importing to the disconnected Satellite server.

4.2.2.4. Importing Content to a Disconnected Satellite Server

Before importing content, ensure that the directory and file system containing the exports has enough space to contain the extracted archives. For example, if your export is 40 GB, the disconnected Satellite Server directory and file system where you are importing the content will need an extra 40 GB of space to expand it on the same file system.

1. Copy all of the Satellite Content ISOs to a directory that the Satellite can access. This example uses `/root/isos`.
2. Create a local directory that will be shared via **httpd** on the Satellite. This example uses `/var/www/html/pub/sat-import/`.

```
# mkdir -p /var/www/html/pub/sat-import/
```

3. Recursively copy the contents of the first ISO to the local directory:

```
# mkdir /mnt/iso
# mount -o loop /root/isos/first iso /mnt/iso
# cp -ruv /mnt/iso/* /var/www/html/pub/sat-import/
# umount /mnt/iso
# rmdir /mnt/iso
```

4. Repeat the above step for each ISO until you have copied all the data from the series of ISOs into the local directory `/var/www/html/pub/sat-import/`.

5. Ensure that the SELinux contexts are correct:

```
# restorecon -rv /var/www/html/pub/sat-import/
```

6. Change the default provider URL in the Satellite web interface:

- a. Log in to the Satellite web interface and select the required organization.
- b. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest**.
- c. On the **Subscription Manifest** information screen select the **Actions** tab. Under **Red Hat Provider Details**, click the edit icon next to the **Red Hat CDN URL** entry and change the URL to reference the location that the ISOs were copied to. This example uses the Satellite fully qualified domain name (FQDN) `server.example.com`, so the URL is:

```
http://server.example.com/pub/sat-import/
```

- d. Click **Browse** to choose the manifest file.
 - e. Click **Upload** to import your manifest.
7. Enable the repositories from the local CDN:
 - a. Click **Content** → **Red Hat Repositories**
 - b. Enable the repositories that were enabled and synchronized in the Synchronizing Content section.

8. Click **Content** → **Sync Status**.
9. Select the repositories you want to synchronize and click **Synchronize Now**.

**NOTE**

The Satellite is now acting as its own CDN with the files located in **`http://localhost`**. This is not a requirement. The CDN can be hosted on a different machine inside the same disconnected network as long as it is accessible to the Satellite server via HTTP.

Once the synchronize finishes, the disconnected Satellite is now ready to serve the content to client systems.

4.2.3. Migrating from Disconnected to Connected Satellite

If your environment changed from disconnected to connected, you can reconfigure a disconnected Satellite to pull content directly from Red Hat Customer Portal:

1. Ensure the correct organization is selected. Navigate to **Content** → **Red Hat Subscriptions** and click **Manage Manifest**.
2. On the **Subscription Manifest** screen select the **Actions** tab. Click the edit icon next to the **Red Hat CDN URL** entry and insert the following URL:

```
https://cdn.redhat.com
```

Click **Save**.

On next synchronization, Satellite will pull content directly from Red Hat Customer Portal.

CHAPTER 5. CONFIGURING A SELF-REGISTERED SATELLITE

A Red Hat Satellite server is normally registered to the Red Hat Customer Portal, then activated as a Satellite Server and gets new content from the Red Hat Customer Portal. A self-registered Red Hat Satellite 6 Server is registered to itself rather than the Red Hat Customer Portal.

Once a Red Hat Satellite 6 server is installed, there are several advantages to registering it as a client to itself:

- The same life cycle management procedures can be applied to the Satellite 6 server itself that have been applied to the rest of the managed estate.
- By subscribing the Satellite 6 server to its own content views, it will receive the same updates on the same schedule as the rest of the managed hosts.
- A virt-who service can be run directly on the Satellite 6 server without the need for an additional host.

There are also several limitations of a self-registered Satellite server:

- A self-registered Satellite Server cannot test package updates by using life cycle environments. It is essential to make a full backup of a self-registered Satellite Server before doing an upgrade to untested packages.
- Not all puppet modules are supported by a self-registered Satellite server. When applying puppet modules to a self-registered Satellite server ensure that they will not create an unsupported configuration.

5.1. REGISTERING A SATELLITE TO ITSELF

Before a self-registered Satellite can be configured to get updates from itself, the Satellite subscription must be added to the Satellite's manifest. When the subscription is in the manifest, the appropriate Satellite repositories can be synchronized into the Satellite.

Procedure 5.1. To Register a Satellite to Itself:

1. If the Satellite is already registered to the Red Hat Customer Portal, unregister the Satellite from the Red Hat Customer Portal using the following commands:

```
# subscription-manager remove --all
# subscription-manager unregister
```

2. The Satellite subscription on the Red Hat Customer Portal is now available and can be transferred into the Satellite's manifest. For further information on Manifests see [Section 4.1.1, "Accessing Red Hat Content Providers"](#).

Transfer the subscription to the Satellite's manifest:

- a. Navigate to <https://access.redhat.com> and click **SUBSCRIPTIONS** on the main menu at the top of the page.
- b. Scroll down to the **Red Hat Subscription Management** section, and click **Satellite** under **Subscription Management Applications**.

- c. Select the required Satellite server by clicking its host name in the table.
 - d. Click **Attach a subscription** and select subscriptions you want to attach. Specify the quantity for each subscription, and click **Attach Selected**.
3. Refresh the manifest on the Satellite Server:
 - a. Log in to the **Satellite** server.
 - b. Ensure that the correct organization is selected.
 - c. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest** at the upper right of the page.
 - d. In the **Subscription Manifest** section, click **Actions** and under the **Subscription Manifest** subsection, click **Refresh Manifest**.
4. Enable Red Hat repositories using the Satellite web interface:
 - a. Click **Content** → **Red Hat Repositories**.
 - b. Navigate to the required repositories. Click each repository set from which you want to select repositories and select the check box for each required repository. The repository is automatically enabled.

For Red Hat Enterprise Linux 6 the repositories that need to be enabled are:

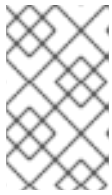
- Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
- Red Hat Satellite 6.1 for RHEL 6 Server RPMs x86_64
- Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server x86_64 6Server
- Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64 Repository

For Red Hat Enterprise Linux 7 the repositories that need to be enabled are:

- Red Hat Enterprise Linux 7 Server RPMs x86_64 6Server
- Red Hat Satellite 6.1 for RHEL 7 Server RPMs x86_64
- Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64 6Server
- Red Hat Enterprise Linux 7 Server - Satellite Tools RPMs x86_64 Repository

5. Synchronize the Satellite server:
 - a. Navigate to **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product repositories available for synchronization is displayed.
 - b. Click the arrow next to the product name to see available content.
 - c. Select the content you want to synchronize.
 - d. Click **Synchronize Now** to starting synchronizing. The status of the

synchronization process will appear in the **Result** column. If synchronization is successful, **Sync complete** will appear in the **Result** column. If synchronization failed, **Error syncing** will appear.



NOTE

Content synchronization can take a long time. The length of time required depends on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

6. Optionally, create a content view to represent the Satellite server. This will allow the Satellite to follow the same life cycle management procedures as the rest of the content on the server. For further information about content views see the Content Views chapter in the [Red Hat Satellite 6.1 User Guide](#)
 - a. To create a content view:
 - i. Log into the web interface as a Satellite administrator.
 - ii. Click **Content** → **Content Views**.
 - iii. Click **Create New View**.
 - iv. Specify the **Name** of the content view. The **Label** field is automatically populated when the **Name** field is filled out. Optionally, provide a description of the content view.
 - v. Click **Save**.
 - b. Edit the content view to add the Red Hat Enterprise Linux server and Satellite repositories:
 - i. Click **Content** → **Content Views** and choose the Content View to add repositories to.
 - ii. Click **Yum Content** and select **Repositories** from the drop-down menu. From the submenu, click **Add**.
 - iii. Select the required repositories to add and click Add Repositories. The required repositories for a self-registered Satellite are all the repositories for the Satellite itself, any supporting repositories and the repository for the Base OS. The repositories required for a self-registered Satellite are listed in Step 4 of this procedure.
7. Download and install the required certificates by running:

```
# rpm -Uvh /var/www/html/pub/katello-ca-consumer-latest.noarch.rpm
```

8. Register the Satellite server and attach the appropriate entitlements using subscription manager. When registering the server you must specify the organization to which the server belongs, also the life cycle environment.

```
# subscription-manager register --org=organization --
environment=environment
```

Example 5.1.

```
# subscription-manager register --org=ExampleCompany --
environment=Library
```

You will be prompted for your Red Hat Satellite user name and password. The Satellite Server administrator can configure new users. See the [Users and Roles](#) chapter in the [Red Hat Satellite 6.1 User Guide](#) for more information.

- Find the pool IDs for the Satellite and for Red Hat Enterprise Linux by running the following command:

```
# subscription-manager list --available
```

- Attach the entitlements by running the following command:

```
# subscription-manager attach --pool Red_Hat_Satellite_Pool_ID --
pool Red_Hat_Enterprise_Linux_ID
```

A content host has now been created for the Satellite server inside of the Satellite server.

- Enable the repositories required for the Satellite server by running the following command:

```
# subscription-manager repos --enable=repository-to-be-enabled
```

See Step 4 of this procedure for the list of repositories that need to be enabled.

- Install the Katello Agent package to allow errata management and package installation through the Satellite web interface. The `katello-agent` package depends on the `goferd` package that provides the `goferd` service. The `goferd` service must be enabled so that the Red Hat Satellite Server or Capsule Server can provide information about errata that are applicable for content hosts.

To install the `katello-agent` run the following command:

```
# yum install katello-agent
```

The `goferd` service is started and enabled automatically after successful installation of `katello-agent`.

5.2. UPDATING A SELF-REGISTERED SATELLITE

A self-registered Red Hat Satellite server is registered to itself rather than directly to the Red Hat Customer Portal. A self-registered Satellite server is able to synchronize with the Red Hat Customer Portal then apply updates to itself at the same time as providing other required updates.

Procedure 5.2. To Update a Self-Registered Satellite:

- It is essential to make a full backup of a self-registered Satellite server prior to doing

an upgrade as package updates cannot be tested. For instructions on how to backup and, if necessary, restore a Satellite server see [Backup and Disaster Recovery](#) in the Red Hat Satellite 6.1 User Guide.

- a. Ensure your backup location has enough disk space to contain a copy of all of the following directories:

- `/etc/`
- `/var/lib/pulp`
- `/var/lib/mongodb`
- `/var/lib/pgsql/`

This can be a considerable amount of space so plan accordingly.

- b. Stop all services:

```
# katello-service stop
```

- c. Run the backup script:

```
# /usr/bin/katello-backup backup_directory
```

This process can take a long time to complete, due to the amount of data to copy.

- d. Restart all services:

```
# katello-service start
```

2. Synchronize to Satellite server:

- a. Navigate to **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product repositories available for synchronization is displayed.
- b. Click the arrow next to the product name to see available content.
- c. Select the content you want to synchronize.
- d. Click **Synchronize Now** to starting synchronizing. The status of the synchronization process will appear in the **Result** column. If synchronization is successful, **Sync complete** will appear in the **Result** column. If synchronization failed, **Error syncing** will appear.



NOTE

Content synchronization can take a long time, and depends on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

3. Optionally, publish and promote the required content views:

- a. After a content view has been created, it needs to be published in order for it to be visible and usable by hosts. Before publishing the content view definition, make sure that the content view definition has the necessary products, repositories and filters.

To publish the content view:

- i. Click **Content** → **Content Views**.
 - ii. Click on the content view that represents the Satellite server.
 - iii. Click **Publish New Version**.
 - iv. Fill in a comment.
 - v. Click **Save**.
- b. After the content view has been published it needs to be promoted into the required life cycle environment.

To promote the content view:

- i. On the main menu, click **Content** → **Content Views**.
 - ii. In the **Name** column, select the content view that represents the Satellite server.
 - iii. On the **Versions** tab, identify the latest version, and click **Promote**.
 - iv. Identify the promotion path where you want to promote the content view, select the appropriate life cycle environment, and click **Promote Version**.
 - v. After the promotion has completed, the **Versions** tab updates to display the new status of your content views.
4. Update the Satellite server:

```
# yum update
# katello-installer --upgrade
```

5. Restart the services:

```
# katello-service restart
```

CHAPTER 6. MANAGING HYPERVISORS AND VIRTUAL GUEST SUBSCRIPTIONS

Red Hat Satellite can track the hypervisors (hosts) that are attached directly to it and the subscriptions of those hosts. However, the hypervisors' guests are not indexed through this mechanism. For the security of the hypervisor infrastructure, this host to guest mapping is not provided during Satellite registration.

The **virt-who** utility collects information about the connection between the hypervisor and its virtual guests and provides Subscription Manager with a mapping file containing the hypervisor-guest pairs. This utility is provided both in the main Red Hat Enterprise Linux repository (rhel-6-server-rpms or rhel-7-server-rpms) as well as in the Red Hat Satellite Tools repository (rhel-6-server-satellite-tools-6.1-rpms or rhel-7-server-satellite-tools-6.1-rpms). The Satellite Tools repository is the recommended source of **virt-who** for Satellite installations. To enable this repository on Red Hat Enterprise Linux 7, execute:

```
# subscription-manager repos --enable=rhel-7-server-satellite-tools-6.1-rpms
```

Then install **virt-who** as follows:

```
# yum install virt-who
```

6.1. INTRODUCTION TO VIRT-WHO

Red Hat uses **virt-who** to keep track of the hypervisors' subscriptions and the guests who can inherit those subscriptions. The **virt-who** system:

1. Scans the hypervisor (host) management platform and its guests
2. Creates the host/guest mapping
3. Communicates this host/guest mapping to Satellite

This host/guest mapping associates every guest with a specific host. Then, a subscription service can attach a single subscription to a virtual host and apply an included and inheritable subscription to a guest, rather than consuming two separate subscriptions for each instance.

After you start **virt-who** the first time, a **virt-who** daemon automatically runs in the background and makes updates based on a schedule you select (the default is hourly).

6.1.1. The Universally Unique Identifier (UUID)

The **virt-who** system makes this host/guest association by extracting a universally unique identifier (UUID) for each guest from the hypervisor and then associating each UUID with its hypervisor in the Satellite inventory.

6.1.2. Important Conditions for virt-who to Correctly Attach Subscriptions

These factors must be true for the subscription service to recognize the host/guest association and correctly attach subscriptions:

- The **virt-who** system must be run periodically to detect new guest instances.
- The hypervisor and the guest systems must be registered to the same subscription service (that is, the same Satellite organization).
- The hypervisor must have a subscription attached to it that includes virtual subscriptions or inheritable subscriptions.

6.1.3. Subscription Status and virt-who

A registered host is assigned a subscription status color based on its installed products and attached subscriptions. When you first register a virtual guest, the host list displays that virtual guest's host subscription status as yellow. The reason is that the Satellite does not know which hypervisor the guest resides on. You must run **virt-who** so that the Satellite knows which hypervisor the guest resides on. With the default **auto-attach** configuration enabled, and assuming **virt-who** runs successfully, the guest subscription displays as green in 24 hours.

6.2. BEFORE YOU BEGIN

6.2.1. Prerequisites

To install and run **virt-who**:

- You must have credentials that allow **virt-who** to communicate with:
 - a Satellite user account
 - your virtualization system
- The system running **virt-who** is registered already to the Satellite server (**virt-who** will use the host credentials).
- The ports configured for your hypervisor allow communication (the default **virt-who** port is 443).

6.2.2. User Login for virt-who

Login credentials to the data center are required for the following hypervisor types:

- Red Hat Enterprise Virtualization Manager
- VMware vSphere
- Microsoft Hyper-V

**NOTE**

When configuring the permissions to the login credentials, the permissions must allow access to the virtual machines and hypervisors. Red Hat recommends the following:

- The login has read-only permission.
- The login is for a service account or non-user login.
- The password does not expire.

6.2.3. virt-who Configuration File Location

The **virt-who** configuration is stored in the following configuration files:

- **/etc/sysconfig/virt-who** (default)

Sample Configuration File:

```
$ cat /etc/sysconfig/virt-who
[rdu]
VIRTWHO_BACKGROUND=1
VIRTWHO_DEBUG=1
VIRTWHO_ESX=1
VIRTWHO_ESX_OWNER=Organization_label
VIRTWHO_ESX_SERVER=vcenter-server.example.com
VIRTWHO_ESX_USERNAME=esx-readonly-user
VIRTWHO_ESX_PASSWORD=password
VIRTWHO_ESX_ENV=Library
```

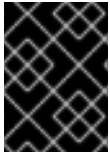
- **/etc/virt-who.d/exampleconfig.conf** (only for encrypted passwords)

Sample Configuration File:

```
$ cat /etc/virt-who.d/exampleconfig.conf
[rdu]
type=abc
owner=virtwho
server=abc-server.example.com
username=root
password=password
rhsm_username=admin
rhsm_password=admin
#rhsm_encrypted_password=61fdel1a1e2cbe95faef0ef0ecfd85057
```

6.2.4. Limitations Related to Satellite Organizations

Subscriptions are arranged according to Satellite organizations. Although the current **virt-who** can report to multiple Satellite organizations, you cannot share subscriptions across organizations.



IMPORTANT

You must have one virtual data center subscription for each organization and for each hypervisor.

6.3. SUPPORTED HYPERVISORS

The **virt-who** system can work with any of the hypervisors outlined in the following table.

Table 6.1. Supported Hypervisors

If you have...	Go here for setup instructions...	Warnings:
<ul style="list-style-type: none"> A Red Hat Enterprise Linux system running KVM Red Hat Enterprise Virtualization Manager 	Section 6.4, “Setting up a Red Hat Enterprise Virtualization Manager Server or Libvirt (KVM) Hypervisor”	None
Microsoft Hyper-V	Section 6.5, “Using virt-who with Hyper-V”	You cannot install virt-who directly on the Hyper-V hypervisors. Instead, you must install virt-who on a Red Hat Enterprise Linux platform that can communicate with the Hyper-V server.
VMware: vCenter, vSphere, or ESX	Section 6.6, “Setting up a VMware Hypervisor”	You cannot install virt-who directly on the VMware hypervisors. Instead, you must install virt-who on a Red Hat Enterprise Linux platform that can communicate with the vCenter server.

6.3.1. Rerunning virt-who

Rerunning **virt-who** does not change a previously created hypervisor's environment or content view. This lets you manually move a hypervisor to a different environment and content view in Satellite. You can also change the **virt-who** host without impacting existing hypervisors. To rerun **virt-who**, use the command option:

```
# virt-who --one-shot
```

Re-registering the host on which **virt-who** is running to a new organization creates new hypervisors in that organization. Previously created hypervisors in another organization remain unchanged (until you delete them manually). If you add an organization, you must restart **virt-who**.

6.4. SETTING UP A RED HAT ENTERPRISE VIRTUALIZATION MANAGER SERVER OR LIBVIRT (KVM) HYPERVISOR

1. Configure Subscription Manager on the virtual system to use Satellite and the CA certificate:

```
# rpm -ivh \
http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

2. Register the Red Hat Enterprise Linux system (which communicates with Red Hat Enterprise Virtualization Manager) to Satellite:

```
# subscription-manager register --username=admin --password=secret -
--org=organization_label --auto-attach
```

The organization label is available in the Satellite web UI. If another system is already registered to that organization, then you can get the label by using the **subscription-manager orgs** command.

3. Install the virt-who packages on the hypervisor.



NOTE

For both the Red Hat Enterprise Virtualization Manager server and the libvirt (KVM) hypervisor, Red Hat recommends that you install the virt-who package on a physical system.

```
# yum install virt-who
```

4. Edit the **virt-who** configuration file (**/etc/sysconfig/virt-who**) and set the parameters as follows:

For a Red Hat Enterprise Virtualization Manager server:

```
VIRTWHO_DEBUG=1
VIRTWHO_SATELLITE6=1
VIRTWHO_RHEVM=1
VIRTWHO_RHEVM_OWNER=organization_label
VIRTWHO_RHEVM_ENV=environment
VIRTWHO_RHEVM_SERVER=RHEV-server_URL
VIRTWHO_RHEVM_USERNAME=desired_user_name
VIRTWHO_RHEVM_PASSWORD=desired_password
```

Note that to determine the organization label for the VIRTWHO_RHEVM_OWNER parameter execute the **subscription-manager identity** command. The user name for the VIRTWHO_RHEVM_USERNAME parameter has the form admin@internal. With the VIRTWHO_SATELLITE6 parameter enabled, **virt-who** sends reports to the Satellite server.

For a libvirt (KVM) hypervisor:

```
VIRTWHO_BACKGROUND=1
```

```
VIRTWHO_DEBUG=1
VIRTWHO_SATELLITE6=1
VIRTWHO_LIBVIRT=1
```

With the `VIRTWHO_SATELLITE6` parameter enabled, `virt-who` sends reports to Red Hat Satellite.

5. Start and enable the **virt-who** service:

- On Red Hat Enterprise Linux 6:

```
# service virt-who start
# chkconfig virt-who on
```

- On Red Hat Enterprise Linux 7:

```
# systemctl start virt-who
# systemctl enable virt-who
```

6. After starting the **virt-who** service, monitor the `/var/log/rhsm/rhsm.log/` file on the same system to confirm whether or not hosts and guests mappings are sent.

```
2015-01-10 13:44:38,651 [DEBUG] @subscriptionmanager.py:112 -
Sending update in hosts-to-guests mapping: {44454c4c-3900-1057-804c-
b2c04f375231: [42346e7b-f3df-6651-4d43-6de0c769c6c7, 564ddf1c-1eec-
aba5-aec4-03d311ca298e, 4234ee7d-b239-ebb1-738f-55a83861d1a5,
42343eb8-838f-18f3-24f9-682455093072, 42345839-6316-6733-f5a1-
bd4213d693b3, 42344725-cf73-f8d9-6bff-c88d4df5c67c]}
```

7. On the Satellite server, go to **Host → Content Hosts** and confirm that host (hypervisor) system profiles display. By default, the hypervisor name is as follows:

For a Red Hat Enterprise Virtualization Manager server:

```
hypervisor UUID
```

For a libvirt (KVM) hypervisor:

```
hypervisor UUID
```

If desired, change this name in the Red Hat Satellite UI by editing the system entry.

- To make virtual subscriptions available for virtual machines, the host system needs a subscription. To know on which host the virtual machine is running, open the virtual machine profile from the **Content Hosts** page. In the **Details** tab, the virtual machine displays as **Virtual Host UUID**. Click the UUID link that opens the host system profile. Then, in the **Subscriptions** tab, assign the subscription to the host system. If you have multiple hypervisors running Red Hat Enterprise Linux guests, attach a subscription to all the hypervisors.
- To consume the subscription assigned to the hypervisor profile on the machine running **virt-who**, unsubscribe and then auto subscribe:

```
# subscription-manager remove --all
# subscription-manager attach --auto
```

10. Confirm whether the subscription attached to the hypervisor is consumed by the guest running **virt-who**:

```
# subscription-manager list --consumed
```

11. When you install new virtual machines on the hypervisor, you must register the new virtual machines and use the subscription attached to the hypervisor:

```
# rpm -ivh \
http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

12. Register the new virtual machines and use the subscription attached to the hypervisor:

```
# subscription-manager register --org=organization_label
# subscription-manager attach --auto
# subscription-manager list --consumed
```

6.5. USING VIRT-WHO WITH HYPER-V

1. To make the **virt-who** connection to Hyper-V work, enable Windows Remote Management and either HTTP or HTTPS listener must be running. On the Hyper-V server:

```
# winrm quickconfig
```

2. The firewall must allow remote administration. On the Hyper-V server:

```
# netsh advfirewall firewall set rule group="Remote Administration"
new enable=yes
```

3. If you are using HTTP, enable the unencrypted connection. On the Hyper-V server:

```
# winrm set winrm/config/service @{AllowUnencrypted="true"}
```

4. Only Basic and NTLM authentication methods are supported. To verify that either Basic or Negotiate is enabled (True):

```
# winrm get winrm/config/service/auth
```

5. On the Red Hat server, log in as root. Install the virt-who package:

```
# yum install virt-who
```

6. Edit the **/etc/sysconfig/virt-who** file and set the parameters as follows:


```

VIRTWHO_BACKGROUND=1
VIRTWHO_DEBUG=1
VIRTWHO_ONE_SHOT=0
VIRTWHO_INTERVAL=0
VIRTWHO_SATELLITE6=1
VIRTWHO_HYPERV=1
VIRTWHO_HYPERV_OWNER=Satellite_Organization
VIRTWHO_HYPERV_ENV=Library
VIRTWHO_HYPERV_SERVER=IP or FQDN
VIRTWHO_HYPERV_USERNAME=Your_User_Name (you must use your Hyper-V
administrator account)
VIRTWHO_HYPERV_PASSWORD=Your_Password

```

With the `VIRTWHO_SATELLITE6` parameter enabled, `virt-who` sends reports to Red Hat Satellite.

7. Start and enable the **virt-who** service:

- On Red Hat Enterprise Linux 6:

```

# service virt-who start
# chkconfig virt-who on

```

- On Red Hat Enterprise Linux 7:

```

# systemctl start virt-who
# systemctl enable virt-who

```

8. Optional: To configure the **virt-who** service to use a Windows domain account, edit your username with a double backslash in the **virt-who** configuration file.

For example:

```

VIRTWHO_HYPERV_USERNAME="MYDOMAIN\\user"

```

6.6. SETTING UP A VMWARE HYPERVISOR

The `virt-who` packages that create the host/guest mapping are available for Red Hat Enterprise Linux. In a VMware environment, you must have Red Hat Enterprise Linux 6.6 or later available to run the **virt-who** service which connects to the VMware hypervisor.

The system running `virt-who` requires open access to vCenter on ports 80 and 443. Before following these steps, create a firewall exception to allow connections on port 80 and 443 from the Red Hat Satellite server to the vCenter:

- On Red Hat Enterprise Linux 6:

```

# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
\
&& iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& service iptables save

```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

- On Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-port="80/tcp" --add-port="443/tcp" \
&& firewall-cmd --permanent --add-port="80/tcp" --add-port="443/tcp"
```

Perform the following steps to set up a VMware hypervisor:

1. Configure Subscription Manager on the virtual system to use the Satellite and the CA certificate, as follows:

```
# rpm -ivh \
http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

2. Register the Red Hat Enterprise Linux system (which communicates with the VMware server) to Satellite.

```
# subscription-manager register --username=admin --password=secret -
--org=organization_label --auto-attach
```

The organization label is available in the Satellite UI for the organization. If another system is already registered to that organization, then you can get the label by using the **subscription-manager orgs** command.

3. Install the virt-who packages.

```
# yum install virt-who
```

4. On the Red Hat Enterprise Linux system (which communicates with the VMware hypervisor), edit the **virt-who** configuration file (**/etc/sysconfig/virt-who**) and set the following parameters (to identify the location of your ESX management server):

```
VIRTWHO_BACKGROUND=1
VIRTWHO_DEBUG=1
VIRTWHO_SATELLITE6=1
VIRTWHO_ESX=1
VIRTWHO_ESX_OWNER=Organization_label
VIRTWHO_ESX_SERVER=vcenter-server.example.com
VIRTWHO_ESX_USERNAME=esx-readonly-user
VIRTWHO_ESX_PASSWORD=MyGNU4pass!!
VIRTWHO_ESX_ENV=Library
```

The **VIRTWHO_ESX_USERNAME** is the local VMware vCenter or Microsoft Active Directory user with read-only permission to the virtual machines and hypervisors. With the **VIRTWHO_SATELLITE6** parameter enabled, virt-who sends reports to Red Hat Satellite.

5. Start and enable the **virt-who** service:

- On Red Hat Enterprise Linux 6:

```
# service virt-who start
# chkconfig virt-who on
```

- On Red Hat Enterprise Linux 7:

```
# systemctl start virt-who
# systemctl enable virt-who
```

The data are added to the following file:

/var/lib/virt-who/hypervisor-systemid-UUID

6. After starting the **virt-who** service, monitor the **/var/log/rhsm/rhsm.log** file on the same system to confirm whether or not hosts and guests mappings are sent.

```
2015-01-10 13:44:38,651 [DEBUG] @subscriptionmanager.py:112 -
Sending update in hosts-to-guests mapping: {44454c4c-3900-1057-804c-
b2c04f375231: [42346e7b-f3df-6651-4d43-6de0c769c6c7, 564ddf1c-1eec-
aba5-aec4-03d311ca298e, 4234ee7d-b239-ebb1-738f-55a83861d1a5,
42343eb8-838f-18f3-24f9-682455093072, 42345839-6316-6733-f5a1-
bd4213d693b3, 42344725-cf73-f8d9-6bff-c88d4df5c67c]}
```

7. On the Satellite server, go to **HOSTS** → **CONTENT HOST** and confirm that host (hypervisor) systems profiles display.

By default, the hypervisor name is **esx hypervisor UUID**. If desired, change this name in the Red Hat Satellite GUI by editing the system entry.

8. To make virtual subscriptions available for virtual machines, the host system needs a subscription. To know on which host the virtual machine is running, open the virtual machine profile from the **Content Host** page. In the **Details** tab, the virtual machine displays as **Virtual Host UUID**. Click the UUID link that opens the host system profile. Then, in the **Subscriptions** tab, assign the subscription to the host system. If you have multiple VMware hypervisors running Red Hat Enterprise Linux guests, then attach a subscription to all the VMware hypervisors.
9. To attach the subscription assigned to the hypervisor profile on the machine running **the virt-who** service, unsubscribe and then auto subscribe:

```
# subscription-manager remove --all
# subscription-manager attach --auto
```

10. Confirm whether the subscription attached to the hypervisor is consumed by the guest running **virt-who**:

```
# subscription-manager list --consumed
```

11. When you install new virtual machines on the hypervisor, you must register the new virtual machines and use the subscription attached to the hypervisor:

■

```
# rpm -ivh \
http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

12. Register the new virtual machines and use the subscription attached to the hypervisor:

```
# subscription-manager register --org=organization_label
# subscription-manager attach --auto
# subscription-manager list --consumed
```

6.7. CONFIGURE VIRT-WHO WITH AN ENCRYPTED PASSWORD

virt-who can encrypt the passwords for the hypervisor and give you the string to use. The encrypted password is located in the `/etc/virt-who.d/` configuration file.

To generate an encrypted password:

1. Verify `/var/lib/virt-who/key` encryption file has root read and write permission.
2. To get an encrypted password string, run the **virt-who-password** as root:

```
# virt-who-password
Password:
Use the following as a value for the encrypted_password key in the
configuration file:
encrypted_password_string
```

Type the password of your hypervisor and write down the encrypted string.

3. Create a new configuration file for **virt-who** inside `/etc/virt-who.d/`.



NOTE

Since a configuration file is created under `/etc/virt-who.d/`, do not specify the hypervisor details in `/etc/sysconfig/virt-who`. For more information, see the man page:

```
$ man virt-who-config
```

For example, on vCenter:

```
# vi /etc/virt-who.d/config
[config]
type=esx
server=vcenter/esx_host>
username=vcenter/esx_username
encrypted_password=encrypted_password_string
owner=owner
env=Library
```

4. Verify that the `/var/lib/virt-who/key` encryption key file has root read and write permission.

```
# ll /var/lib/virt-who/key
-rw----- . 1 root root 130 Jun 29 14:43 /var/lib/virt-who/key
```

5. After the configuration change, restart the **virt-who** service.

- On Red Hat Enterprise Linux 6:

```
# service virt-who restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart virt-who
```

6. To determine the value of **owner** in the `/etc/virt-who.d/` configuration file, run the following command. The org ID *string* is the *owner* value:

```
# subscription-manager identity
org ID : string
```

6.8. VCENTER CONFIGURATION EXAMPLE FOR REPORTING DATA TO MULTIPLE ORGANIZATIONS

In this example, you have two vCenter environments, and you want to do the following:

- Place hypervisors from the first instance of vCenter into the Organization 'Engineering' on your Satellite 6.
- Place hypervisors from the second instance of vCenter into the Organization 'Operations' on your Satellite 6.

NOTE

You must have **virt-who** running on two systems, one for each organization. The following system hostnames denote the difference between the two virt-who systems:

```
hostname - eng-virt-who.example.com (virt-who instance reports
hypervisors in vCenter1 to the 'Engineering' Organization)
hostname - ops-virt-who.example.com (virt-who instance reports
hypervisors in vCenter2 to the 'Operations' Organization)
```

This example uses the following information:

```
Vcenter1:
Hostname - vcenter1.example.com
username - read_write@vsphere.local
password - supersecret
```

```
Vcenter2:  
Hostname - vcenter2.example.com  
username - read_only@vsphere.local  
password - notsosecret
```

Procedure 6.1. Part 1

1. On system **eng-virt-who.example.com**, install **virt-who**:

```
[root@eng-virt-who.example.com]# yum install virt-who
```

2. Create an encrypted password string for **vcenter1**:

```
[root@eng-virt-who.example.com]# virt-who-password  
Password: type the 'supersecret' password  
Use following as value for encrypted_password key in the  
configuration file:  
5e7367195d9fe2aa4b6667f93f17c5bd
```

3. Edit **/etc/virt-who.d/vcenter-1** and add the following content:

```
[vcenter-1]  
type=esx  
server=vcenter1.example.com  
username=read_only@vsphere.local  
encrypted_password=5e7367195d9fe2aa4b6667f93f17c5bd  
owner=Engineering  
env=Library
```

4. Restart **virt-who**.

- On Red Hat Enterprise Linux 6:

```
# service virt-who restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart virt-who
```

Procedure 6.2. Part 2

On system **ops-virt-who.example.com**, complete the following steps:

1. Install **virt-who**:

```
[root@ops-virt-who.example.com]# yum install virt-who
```

2. Create an encrypted password string for **vcenter2**:

```
[root@ops-virt-who.example.com]# virt-who-password  
Password: type the 'notsosecret' password
```

```
Use following as value for encrypted_password key in the
configuration file:
4ff5da2eee0648d99fd0c24337f98bd6
```

3. Edit **/etc/virt-who.d/vcenter-2** and add the following content:

```
[vcenter-2]
type=esx
server=vcenter2.example.com
username=read_only@vsphere.local
encrypted_password=4ff5da2eee0648d99fd0c24337f98bd6
owner=Operations
env=Library
```

4. Restart **virt-who**.

- On Red Hat Enterprise Linux 6:

```
# service virt-who restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart virt-who
```

6.9. REGISTERING GUEST INSTANCES

Register a virtual system the same as a physical system.

The **virt-who** service must be running on the virtual host or on a hypervisor in the environment (for VMware). This ensures that the **virt-who** service maps the guest to a physical host, so the system is registered as a virtual system. Otherwise, the virtual instance is treated as a physical instance.

1. Configure Subscription Manager on the virtual system to use the Satellite service and the CA certificate.

```
# rpm -Uvh \
http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

2. Register the system to the same organization as its host.

```
# subscription-manager register --username=admin --password=secret -
-org=organization_label --auto-attach
```

The organization ID is available in the Portal entry for the organization. If another system is already registered to that organization, then get the organization ID by using the following command:

```
# subscription-manager orgs
```

6.10. REMOVING A GUEST ENTRY

To remove a guest entry, you must unregister the guest from the Satellite.

```
# subscription-manager unregister
```

If the system has been deleted, however, the virtual service (like **virt-who**) cannot tell whether the service is deleted or paused. In that case, manually remove the system from Satellite.

1. Log into the Satellite UI.
2. In the top menu, hover over the **Systems** item and click the **All** item.
3. In the left column, click the name of the system.
4. At the top of the system's details page, click the **Remove System** link.

6.11. REMOVING A HYPERVISOR ENTRY

1. Unregister the hypervisor.

```
# subscription-manager unregister
```

2. For VMware, delete the UUID file to remove the host/guest mapping records: **/var/lib/virt-who/hypervisor-systemid-UUID**

6.12. TROUBLESHOOTING VIRT-WHO

This section lists selected problems that can occur when integrating Satellite with virt-who.

Scenario 1: You have Satellite running together with a hypervisor. You install another hypervisor and run the **virt-who** command. The host list in the Satellite web UI now displays two green hypervisors. One hypervisor has a subscription attached, and you create a guest ID. Run **virt-who** again. The host list now displays two green hypervisors, but the new guest ID is displayed as red.

Solution: The hypervisor tool migrated the guest from hypervisor 1 to hypervisor 2. To fix this problem, choose one of the following options:



- Move the virtual data subscription to hypervisor 2.
- Move the guest to hypervisor 1.
- Stop using this guest.

Scenario 2: In Satellite, you provision a guest on a hypervisor that does not have a subscription. The host list in the Satellite web UI displays the hypervisor as yellow. 24 hours later, the hypervisor is displayed as red.

Solution: The hypervisor probably does not have a correctly attached subscription. Obtain a subscription for this hypervisor.

Scenario 3: Either of the following error messages display:

■

-  Host unknown status
-  Late binding to a host through virt-who (host/guest mapping)

Solution: Search for the error output from **virt-who** in the **/var/log/rhsm/rhsm.log** file. Then, search the errors in the knowledgebase in [Red Hat Customer Portal](#).

Scenario 4: In Satellite, you provision a guest on a hypervisor that has a subscription. The host list in the Satellite web UI displays the hypervisor as yellow.

Solution: Either wait for **virt-who** to run and fix the problem itself, or run **virt-who** manually.

CHAPTER 7. INSTALLING RED HAT SATELLITE CAPSULE SERVER

The Red Satellite Capsule Server is a Satellite component that provides federated services to discover, provision, and configure hosts outside of the primary Satellite server. A Satellite Capsule Server provides the following features:

- Pulp Server features, including:
 - Repository synchronization
 - Content delivery
- Red Hat Satellite Provisioning Smart Proxy features, including:
 - DHCP, including ISC DHCP servers
 - DNS, including Bind
 - Any UNIX-based TFTP server
 - Puppet Master servers from 0.24
 - Puppet CA to manage certificate signing and cleaning
 - Baseboard Management Controller (BMC) for power management

The Satellite Capsule Server is a means to scale out the Satellite installation. Organizations can create various capsules in different geographical locations where the data centers are located. These are centrally managed through the Satellite Server. When a Satellite user promotes content to the production environment, the Satellite Server will push the content from the Satellite Server to each of the Satellite Capsule Servers. Host systems pull content and configuration from the Satellite Capsule Servers in their location and not from the central Satellite Server.

Creating various Satellite Capsule Servers will decrease the load on the central server, increase redundancy, and reduce bandwidth usage.

7.1. RED HAT SATELLITE CAPSULE SERVER SCALABILITY

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit but has been tested on a Satellite Server with a Red Hat Enterprise Linux 6.6 and 7 hosts. Currently, running fourteen capsules with two vCPUs have been tested without issues.

7.1.1. Capsule Scalability with Puppet Clients

Capsule scalability depends heavily on the following factors, especially when managing puppet clients:

- Number of CPUs
- Run-interval distribution
- Number of puppet classes

The Capsule Server has a concurrency limitations of 100 concurrent puppet agents running at any single point in time. Running more than 100 concurrent puppet agents will result in a 503 HTTP error.

For example, assuming that the puppet agent runs are evenly distributed with less than 100 concurrent puppet agents running at any single point during a run-interval, a Capsule Server with four CPUs can expect a maximum of 1250-1600 puppet clients with a moderate workload of 10 puppet classes assigned to each puppet client. Depending on the number of puppet clients required, the Satellite installation can scale out the number of Capsule Servers to support them.

Based on the following assumptions:

- There are no external puppet clients reporting directly to the Satellite 6 integrated capsule.
- All other puppet clients report directly to an external capsule.

Puppet scalability within Satellite on Red Hat Enterprise Linux 6.6 Capsules is as follows:

- With minimum number of CPUs (two CPUs):
 - At 1 puppet class per host: Not tested
 - At 10 puppet classes per host: Maximum of 1020-860
 - At 20 puppet classes per host: Maximum of 375-330
- With recommended number of CPUs (four CPUs):
 - At 1 puppet class per host: Maximum of 2250-1875
 - At 10 puppet classes per host: Maximum of 1600-1250
 - At 20 puppet classes per host: Maximum of 700-560



NOTE

The information above represents an evenly distributed run interval of all puppet agents. Any deviation runs the risk of filling the passenger request queue and is subject to the concurrency limitation of 100 concurrent requests.

7.2. RED HAT SATELLITE CAPSULE SERVER PREREQUISITES

The Satellite Capsule's requirements are identical to the Satellite Server. These conditions must be met before installing Red Hat Satellite Capsule:



IMPORTANT

The Red Hat Satellite server and Capsule server versions must match. For example, a Satellite 6.0 server cannot run a 6.1 Capsule server and a Satellite 6.1 server cannot run a 6.0 Capsule server. Mismatching Satellite server and Capsule server versions will result in the Capsule server failing silently.

7.2.1. Base Operating System

Install the operating system from disc, local ISO image, kickstart, or any other methods that Red Hat supports. Red Hat Satellite Capsule requires Red Hat Enterprise Linux installations with the @Base package group with no other package-set modifications, and without third-party configurations or software that is not directly necessary for the direct operation of the server. This restriction includes hardening or other non-Red Hat security software. If such software is required in your infrastructure, install and verify a complete working Red Hat Satellite Capsule first, then create a backup of the system before adding any non-Red Hat software.

When installing Red Hat Enterprise Linux from CD or ISO image, there is no need to select any package groups; Red Hat Satellite Capsule only requires the base operating system installation. When installing the operating system via kickstart, select the @Base package group.

- Red Hat Satellite Capsule requires a networked base system with the following minimum specifications:
 - 64-bit architecture.
 - The latest version of Red Hat Enterprise Linux 6 Server or 7 Server.
 - A minimum of two CPU cores, but four CPU cores are recommended.
 - A minimum of 12 GB memory but ideally 16 GB of memory for each Satellite instance. A minimum of 4 GB of swap is recommended.
 - A minimum of 5 GB storage for the base install of Red Hat Enterprise Linux, 300 MB for the installation of Red Hat Satellite Capsule and at least 10 GB storage for each unique software repository to be synchronized in the /var file system.

Packages that are duplicated in different repositories are only stored once on the disk. Additional repositories containing duplicate packages will require less additional storage.



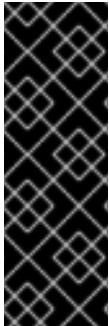
NOTE

The bulk of storage resides on the **/var/lib/mongodb** and **/var/lib/pulp** directories. These end points are not manually configurable. Ensure that storage is available on the **/var** file system to prevent storage issues.

- No Java virtual machine installed on the system, remove any if they exist.
 - No Puppet RPM files installed on the system.
 - No third-party unsupported yum repositories enabled. Third-party repositories may offer conflicting or unsupported package versions that may cause installation or configuration errors.
- Administrative user (**root**) access.
 - Full forward and reverse DNS resolution using a fully qualified domain name. Check that **hostname** and **localhost** resolve correctly, using the following commands:

```
# ping -c1 localhost
# ping -c1 `hostname -f` # my_system.domain.com
```

- Ensure the Satellite Server's base system can resolve the Capsule's host name.
- Available subscriptions on the Red Hat Satellite Server.



IMPORTANT

Make sure that the host system is fully updated before installing Red Hat Satellite Capsule Server. Attempts to install on host systems running Red Hat Enterprise Linux that are not fully updated may lead to difficulty in troubleshooting, as well as unpredictable results.

Red Hat recommends that the Satellite Capsule system be a freshly provisioned system that serves no other function except as a Satellite Capsule.

7.2.2. Application Specifications

Satellite application installation specifications are as follows:

It is recommended that a time synchronizer such as **ntpd** is installed and enabled on Satellite. Run the following command to start the time synchronizer and have it persist across restarts:

For Red Hat Enterprise Linux 6:

```
# chkconfig ntpd on; service ntpd start
```

For Red Hat Enterprise Linux 7:

```
# systemctl start chronyd; systemctl enable chronyd
```

7.2.3. Network Ports Required for Capsule Communications

The tables in this section list the ports required for configuring a Red Hat Satellite Capsule. A list of ports can also be found in the Red Hat Knowledgebase solution [Satellite 6.1 Definitive List of Ports](#).

Table 7.1. Ports for Satellite to Capsule Communication

Port	Protocol	Service	Required for
9090	TCP	HTTPS	Connections to the proxy in the Capsule
80	TCP	HTTP	Satellite to Capsule, for downloading a bootdisk (Optional)
443	TCP	HTTPS	Connections to the Pulp server in the Capsule [a]
[a] Added in Satellite 6.1.9			

Port	Protocol	Service	Required for
------	----------	---------	--------------

Table 7.2. Ports for Capsule to Satellite Communication

Port	Protocol	Service	Required for
443	TCP	HTTPS	Connections to Katello, Foreman, Foreman API, and Pulp
5646	TCP	amqp	Capsule's Qpid dispatch router to Qpid dispatch router in the Satellite
5647	TCP	amqp	The Katello agent to communicate with the Satellite's Qpid dispatch router

The base system on which a Capsule Server is running is a managed host, a client, that is directly connected to the Satellite Server. See [Table 1.5, “Ports for Client to Satellite Communication”](#).

Table 7.3. Ports for Client to Capsule Communication

Port	Protocol	Service	Required for
53	TCP and UDP	DNS	Queries to the DNS service
67	UDP	DHCP	For Client provisioning from the Capsule
69	UDP	TFTP	Downloading PXE boot image files
80	TCP	HTTP	Anaconda, yum, and for obtaining Katello certificate updates
443	TCP	HTTPS	Anaconda, yum, Telemetry Services, and Puppet
5647	TCP	amqp	The Katello agent to communicate with the Capsule's Qpid dispatch router
8000	TCP	HTTPS	Anaconda to download kickstart templates to hosts, and for downloading iPXE firmware
8140	TCP	HTTPS	Puppet agent to Puppet master connections

Port	Protocol	Service	Required for
8443	TCP	HTTPS	Subscription Management Services connection to the reverse proxy for the certificate-based API
9090	TCP	HTTPS	Sending generated SCAP reports to the proxy in the Capsule for spooling

Connections from Satellite to Capsule

To configure the firewall on a **Capsule** to enable incoming connections from the **Satellite**, and to make these rules persistent during reboots, enter the commands below appropriate to the Red Hat release.

The ports in these commands are taken from the table [Table 7.1, “Ports for Satellite to Capsule Communication”](#). Note that port 9090 is also listed in the [Table 7.3, “Ports for Client to Capsule Communication”](#). Review the commands to avoid duplicating entries.

- On a Red Hat Enterprise Linux 6 Capsule, execute as **root**:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 9090 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables restart
# chkconfig iptables on
```

- On a Red Hat Enterprise Linux 7 Capsule, execute as **root**:

```
# firewall-cmd --add-port="9090/tcp" \
--add-port="443/tcp" \
&& firewall-cmd --permanent --add-port="9090/tcp" \
--add-port="443/tcp"
```

Connections from Capsule to Satellite

To configure the firewall on a **Satellite** to enable incoming connections from a **Capsule**, and to make these rules persistent during reboots, enter the commands below appropriate to the Red Hat release.

The ports in these commands are taken from the table [Table 7.2, “Ports for Capsule to Satellite Communication”](#). Note that port 443 and 5647 are also listed in the [Table 1.5, “Ports for Client to Satellite Communication”](#). Review the commands to avoid duplicating entries.

- On a Red Hat Enterprise Linux 6 Satellite, execute as **root**:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
```

```
&& iptables -A INPUT -m state --state NEW -p tcp --dport 5646 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 5647 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables restart
# chkconfig iptables on
```

- On a Red Hat Enterprise Linux 7 Satellite, execute as **root**:

```
# firewall-cmd --add-port="443/tcp" \
--add-port="5646/tcp" --add-port="5647/tcp" \
&& firewall-cmd --permanent --add-port="443/tcp" \
--add-port="5646/tcp" --add-port="5647/tcp"
```

Connections from Client to Capsule

To configure the firewall on a **Capsule** to enable incoming connections from a **Client**, and to make these rules persistent during reboots, enter the commands below appropriate to the Red Hat release.

The ports in these commands are taken from the table [Table 7.3, “Ports for Client to Capsule Communication”](#). Note that port 443 and 9090 are also listed in the [Table 7.1, “Ports for Satellite to Capsule Communication”](#). Review the commands to avoid duplicating entries.

- On a Red Hat Enterprise Linux 6 Capsule, execute as **root**:

```
# iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p udp --dport 67 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p udp --dport 69 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 5647 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 8000 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 8140 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 8443 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 9090 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:


```
# service iptables restart
# chkconfig iptables on
```

- On a Red Hat Enterprise Linux 7 Capsule, execute as **root**:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" \
--add-port="69/udp" --add-port="80/tcp" \
--add-port="443/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp" \
&& firewall-cmd --permanent --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" \
--add-port="69/udp" --add-port="80/tcp" \
--add-port="443/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```



NOTE

For information on SELinux types for the ports mentioned in this section, see [Section 1.4.6, “SELinux Policy on Satellite 6”](#)

7.3. OBTAINING THE REQUIRED PACKAGES FOR THE CAPSULE SERVER

Prerequisites

- The Satellite Server's base system must be able to resolve the host name of the Capsule Server's base system.
- You will need a Red Hat Satellite user name and password.
- Register the Capsule Server to the Red Hat Satellite Server to use the Red Hat Satellite Server products and subscriptions:

1. Install the Red Hat Satellite Server's CA certificate in the Capsule Server:

```
# rpm -Uvh http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2. Register the Capsule server with your organization by using the organization **label**:

```
# subscription-manager register --org organization_label
```

You will be prompted for your Red Hat Satellite user name and password. The Satellite Server administrator can configure new users. See the [Users and Roles](#) chapter in the [Red Hat Satellite 6.1 User Guide](#) for more information.

Procedure 7.1. To Install a Satellite Capsule Server on a Certificate-managed System:

1. List all the available subscriptions to find the correct Red Hat Satellite and Red Hat Enterprise Linux product to allocate to your system:

```
# subscription-manager list --available --all
```

The screen displays:

```
Subscription Name: Red Hat Satellite Capsule Server
Provides:          Red Hat Satellite Proxy
                  Red Hat Satellite Capsule
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Satellite Capsule
                  Red Hat Enterprise Linux Server
                  Red Hat Enterprise Linux High Availability (for
RHEL Server)
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
SKU:               MCT0369
Pool ID:           9e4cc4e9b9fb407583035861bb6be501
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2015
System Type:       Physical
```



NOTE

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

2. Subscribe to the required pool IDs:

```
# subscription-manager subscribe --
pool=Red_Hat_Satellite_Capsule_Pool_Id
```

3. Disable all existing repositories:

```
# subscription-manager repos --disable "*"
```

4. Enable the Satellite and Red Hat Enterprise Linux repositories by running **subscription-manager**. You might need to alter the Red Hat Enterprise Linux repository to match the specific version you are using. If enabling a repository unexpectedly fails, check the correct repository is enabled on the Satellite Server. In the web UI, navigate to **Content** → **Red Hat Repositories** and check the status of the repository under **Content** → **Sync Status**.

For Red Hat Enterprise Linux 6:

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-6-server-satellite-capsule-6.1-rpms
```

-

For Red Hat Enterprise Linux 7:

```
# subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-7-server-satellite-capsule-6.1-rpms
```

5. If required, to verify what repositories have been enabled, use the **yum repolist enabled** command. For example, on Red Hat Enterprise Linux 7:

```
# yum repolist enabled
Loaded plugins: langpacks, product-id, subscription-manager
repo id                                repo name
status
!rhel-7-server-rpms/7Server/x86_64    Red Hat
Enterprise Linux 7 Server (RPMs)      7,617
!rhel-7-server-satellite-capsule-6.1-rpms/x86_64 Red Hat Satellite
Capsule 6.1 (for RHEL 7 Server) (RPMs) 176
repolist: 7,793
```

6. Run the following command as the **root** user to install the **capsule-installer** package:

```
# yum install capsule-installer
```

The **capsule-installer** package provides the **capsule-installer** functionality.

7.4. RUNNING THE INSTALLATION AND CONFIGURATION PROGRAM FOR CAPSULE SERVER

Prerequisites

You must meet the following conditions before continuing on this task:

- Install the Red Hat Satellite Server.
- Red Hat recommends that SELinux on the Satellite 6 Capsule Server is set to enforcing.
- Create a Capsule Server certificate on the Satellite Server:

1. On the Satellite Server, use the **capsule-certs-generate** command:

```
# capsule-certs-generate --capsule-fqdn capsule.example.com --
certs-tar ~/capsule.example.com-certs.tar
```

Where:

- **capsule-fqdn** is the Satellite Capsule Server's fully qualified domain name. Mandatory.
- **certs-tar** is the name of the file to generate that will contain the certificate for the Satellite Capsule installer.

The **capsule-certs-generate** command returns the installation instructions with the commands to be executed on the Capsule Server, however if you have followed the procedure in the previous section then you have already installed

the Satellite's CA certificate contained in the `katello-ca-consumer-latest` package and registered the Capsule to the Satellite.

Note that the syntax of those commands depends on the parameters of **capsule-certs-generate** and the fully qualified domain name of your Satellite. For example, the **capsule-certs-generate** command executed on Satellite with FQDN *satellite.example.com* generates the following output:

To finish the installation, follow these steps:

1. Ensure that the `capsule-installer` package is available on the system.
2. Copy `~/capsule.example.com-certs.tar` to the capsule system `capsule.example.com`
3. Run the following commands on the capsule (possibly with the customized parameters, see `capsule-installer --help` and documentation for more info on setting up additional services):

```
rpm -Uvh http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
subscription-manager register --org "Default_Organization"
capsule-installer --parent-fqdn
"satellite.example.com"\
    --register-in-foreman "true"\
    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLua"\
    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
    --certs-tar
"~/capsule.example.com-certs.tar"\
    --puppet "true"\
    --puppetca "true"\
    --pulp "true"
```



IMPORTANT

The **capsule-certs-generate** command returns the arguments required to successfully install a Capsule with the **capsule-installer** command. The **--foreman-oauth-key** and **--foreman-oauth-secret** arguments are always required, the **--pulp-oauth-secret** argument is required if the Capsule will host content (the **--pulp** option set to true). See [Section 7.4.1, “Installing a Capsule Server”](#) for more information on installing a Capsule.

2. Copy the archive file created by **capsule-certs-generate**, in this case called **capsule.example.com-certs.tar**, from the Satellite Server to the Capsule Server.

**NOTE**

If you have a custom certificate, see [Section 7.5.1, “Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate”](#) for instructions.

The following sections will assist in configuring a Satellite Capsule Server for use with your Red Hat Satellite Server. This includes the following types of Satellite Capsule Servers:

- Satellite Capsule Server with content functionality.
- Satellite Capsule Server without content functionality.

7.4.1. Installing a Capsule Server

You can install a Capsule Server by using customized parameters, depending on your intended use case. See **capsule-installer --help** for a list of the available parameters.

To install a Capsule by using the default method, run the following command (also found in the output from **capsule-certs-generate**):

```
# capsule-installer --parent-fqdn "satellite.example.com"\
--register-in-foreman "true"\
--foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
--foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLua"\
--pulp-oauth-secret
"doajBEXqNcAny93ZbciFyysWaiwt6BWU"\
--certs-tar "~/capsule.example.com-
certs.tar"
--puppet "true"
--puppetca "true"
--pulp "true"
```

To enable or disable other services, run **capsule-installer --help** and specify the desired value from the list of command options.

7.4.2. Verifying Your Capsule Server Installation

If the configuration is successful, run this command as the root user on the Satellite Capsule Server:

```
# echo $?
```

This command should return a "0" to indicate success. If it does not, check the **/var/log/katello-installer/capsule-installer.log** file to debug the cause of failure. This log file contains the output generated by the **capsule-certs-generate** and **capsule-installer** commands.

The Satellite Capsule Server should also appear in the Satellite Server's User Interface under **Infrastructure** → **Capsules**.



NOTE

If the new capsule does not appear under **Infrastructure** → **Capsules**, you might have to associate it with your organization. Navigate to **Administer** → **Organizations**. On the **Organizations** page, the following message indicates an unassigned capsule:

Notice: There is 1 host with no organization assigned

On the same page, select your organization and pick the capsule from the list on the **Capsule** tab.

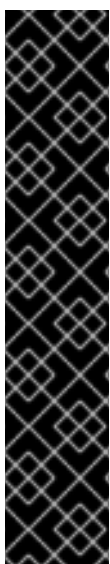
7.5. OPTIONAL CONFIGURATION OPTIONS

The following sections show how to enable additional configuration options for the Satellite Capsule Server.

7.5.1. Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate

Red Hat Satellite comes with a default certificate authority (CA) used by both the server and client SSL certificates for authentication of subservices. The server and client certificates can be replaced with custom ones. For more information on creating custom certificates, see the [Red Hat Enterprise Linux 7 Security Guide](#)^[8]

Custom server and client certificates may be implemented either when the command **capsule-certs-generate** is first run or any time afterward. If **capsule-certs-generate** has not been run before, see [Procedure 7.2, “To Set a Custom Server Certificate When Running capsule-certs-generate for the First Time:”](#), otherwise see [Procedure 7.3, “To Set a Custom Server Certificate After Running capsule-certs-generate:”](#).

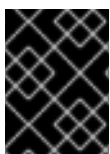


IMPORTANT

When using custom SSL certificates with chained trusts or issuers, include all certificates in the chain into a single file and use that file as the CA certificate value to **katello-installer** parameter **--certs-server-ca-cert**. It is important to concatenate the certificates in the right order so that the trust chain can be validated.

```
# cat 1st_ca.cert 2nd_ca.cert 3th_ca.cert >
/root/sat_cert/ca.bundle
# katello-installer --certs-server-ca-cert
/root/sat_cert/ca.bundle --certs-update-server-ca
```

The certificate's Common Name (CN) must match the fully qualified domain name of the server on which it is used.



IMPORTANT

The certificate's Common Name (CN) must match the fully qualified domain name of the server on which it is used.

Prerequisites

You must have the following files:

Certificate file for the Capsule Server.

Capsule certificates generate parameter **--server-cert**. In this example, **capsule.crt**.

Certificate signing request file for the Capsule Server.

Capsule certificates generate parameter **--server-cert-req**. In this example, **capsule.crt.req**.

Capsule Server's private key used to sign the certificate.

Capsule certificates generate parameter **--server-key**. In this example, **capsule.key**.

CA certificate.

Capsule certificates generate parameter **--server-ca-cert**. In this example, example **cacert.crt**.

Other capsule-certs-generate Parameters

- The parameter **--certs-tar** specifies the name of the archive file to be output by the **capsule-certs-generate**.
- The parameter **--capsule-fqdn** is the Satellite Capsule Server's fully qualified domain name.

Procedure 7.2. To Set a Custom Server Certificate When Running **capsule-certs-generate** for the First Time:



NOTE

In this example the files are stored in the directory **/root/sat_cert**. Using an absolute path in the **root** users' directory provides a fixed location that is available to all users who log in to the server with **root** permissions. Before running this command, ensure the directory already exists.

1. Run the following command on the Red Hat Satellite Server to create the certificates archive:

```
# capsule-certs-generate \
--capsule-fqdn "capsule.example.com" \
--certs-tar /root/sat_cert/capsule.example.com-certs.tar \
--server-cert /root/sat_cert/capsule.crt \
--server-cert-req /root/sat_cert/capsule.crt.req \
--server-key /root/sat_cert/capsule.key \
--server-ca-cert /root/sat_cert/cacert.crt
```

Where:

- **--capsule-fqdn** is the Satellite Capsule Server's fully qualified domain name. Mandatory.
- **--certs-tar** is the name of the tar file to be generated that contains the

certificate to be used by the Satellite Capsule installer.

- **--server-cert** is the path to your certificate, signed by your certificate authority (or self-signed).
 - **--server-cert-req** is the path to your certificate signing request file that was used to create the certificate.
 - **--server-key** is the private key used to sign the certificate.
 - **--server-ca-cert** is the path to the CA certificate on this system.
2. Copy the generated archive file, **capsule.example.com-certs.tar**, from the Satellite Server to the Satellite Capsule Server.
 3. On the Satellite Capsule Server:
 - a. Run the following commands to register your Satellite Capsule Server to the Satellite Server:

```
# rpm -Uvh http://satellite.example.redhat.com/pub/katello-ca-
consumer-latest.noarch.rpm
# subscription-manager register --org "ACME_Corporation" --env
[environment]/[content_view_name]
```



NOTE

The Satellite Capsule Server must be assigned to an organization, because it requires an environment to synchronize content from the Satellite Server. Only organizations have environments.

Assigning a location is optional, but recommended, to indicate proximity to the hosts that the Satellite Capsule Server is managing.

- b. Depending on the desired Satellite Capsule Server type, choose one of the following options:

- **Satellite Capsule Server with content functionality**

Run the following command on the Satellite Capsule Server to enable the custom certificate. The significant parameter is **--pulp="true"**, which indicates that content functionality is to be enabled.

```
# capsule-installer --pulp="true" \
--qpid-router="true" \
--puppet="true" \
--puppetca="true" \
--reverse-proxy="true" \
--certs-tar "~/capsule.example.com-certs.tar"
```

- **Satellite Capsule Server without content functionality**

Run the following command on the Satellite Capsule Server to enable the custom certificate. The significant parameter is **--pulp="false"**, which indicates that content functionality is not to be enabled.

–


```
# capsule-installer --pulp="false" \
--qpid-router="false" \
--puppet="true" \
--puppetca="true" \
--reverse-proxy="true" \
--certs-tar "~/capsule.example.com-certs.tar"
```

Procedure 7.3. To Set a Custom Server Certificate After Running `capsule-certs-generate`:

Using custom server certificates for the Satellite Server means that the same custom server certificates need to be deployed in the Satellite Capsule Servers. Each Satellite Capsule Server requires the following steps:

1. Run the following command as the **root** user on the Satellite Server to generate a new certificate based on your custom server certificate:



NOTE

In this example the files are stored in the directory `/root/sat_cert`. Using an absolute path in the **root** users' directory provides a fixed location that is available to all users who log in to the server with **root** permissions. Before running this command, ensure the directory already exists.

```
# capsule-certs-generate \
--capsule-fqdn "capsule.example.com" \
--certs-tar /root/sat_cert/capsule-certs.tar \
--server-cert /root/sat_cert/capsule.crt \
--server-cert-req /root/sat_cert/capsule.crt.req \
--server-key /root/sat_cert/capsule.key \
--server-ca-cert /root/sat_cert/cacert.crt \
--certs-update-server
```

2. Copy the generated archive file, **capsule.example.com-certs.tar**, from the Satellite Server to the Satellite Capsule host system.
3. On the Satellite Capsule Server, re-run the **capsule-installer** command to refresh the certificates. Depending on the desired Satellite Capsule Server type, choose one of the following options:

- **Satellite Capsule Server with content functionality**

Run the following command on the Satellite Capsule Server to refresh the certificates. The significant parameter is **--pulp="true"**, which indicates that content functionality is to be enabled.

```
# capsule-installer --pulp="true" \
--qpid-router="true" \
--puppet="true" \
--puppetca="true" \
--reverse-proxy="true" \
--certs-tar "capsule.example.com-certs.tar"
```

- **Satellite Capsule Server without content functionality**

Run the following command on the Satellite Capsule Server to refresh the certificates. The significant parameter is **--pulp="false"**, which indicates that content functionality is not to be enabled.

```
# capsule-installer --pulp="false" \  
--qpid-router="false" \  
--puppet="true" \  
--puppetca="true" \  
--reverse-proxy="true" \  
--certs-tar "capsule.example.com-certs.tar"
```

7.5.2. Using Power Management Features on Managed Hosts

When you enable the *baseboard management controller* (BMC) module on the Capsule Server, you can use power management commands on managed hosts using the *intelligent platform management interface* (IPMI) or a similar protocol.

The BMC service on the satellite Capsule Server allows you to perform a range of power management tasks. The underlying protocol for this feature is IPMI; also referred to as the BMC function. IPMI uses a special network interface on the managed hardware that is connected to a dedicated processor that runs independently of the host's CPUs. In many instances the BMC functionality is built into chassis-based systems as part of chassis management (a dedicated module in the chassis).

To take advantage of BMC features you need to add a new network interface of type "BMC" to each managed host. IPMI interfaces are nearly always password protected, to prevent unauthorized people on the same network from gaining control of that host. Satellite uses this NIC to pass the appropriate credentials to the host.

Red Hat Satellite supports by extension everything that either `ipmitool` or `freeipmi` BMC providers support. You can switch between the two per capsule. Note that different hardware vendors might not implement all IPMI specifications, bugs, and so on.

7.5.2.1. Installing a Capsule Server with BMC Options

This section shows how to enable the BMC module as part of the Capsule Server installation process.

Prerequisites

Have a *baseboard management controller* (BMC) provider set up for your deployment of Capsule Server.

To add BMC functionality, you will need to append the options to the **capsule-installer**. You are required to choose either a Capsule Server with content functionality or one without. See [Section 7.4.1, "Installing a Capsule Server"](#) for more information.

Append the following lines to the command in each option:

```
--bmc "enabled"\  
--bmc_default_provider "freeipmi"
```

For example:

- For Capsule Server Installations with content functionality:

```
# capsule-installer --pulp=true
--foreman-oauth-key "xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
--foreman-oauth-secret "w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
--pulp-oauth-secret "doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
--certs-tar "~/capsule.example.com-certs.tar"\
--qpid-router=true\
--puppet=true\
--puppetca=true\
--reverse-proxy=true\
--bmc "enabled"\
--bmc_default_provider "freeipmi"
```

- For Capsule Server Installations without content functionality:

```
# capsule-installer --pulp=false
--foreman-oauth-key "xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
--foreman-oauth-secret "w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
--certs-tar "~/capsule.example.com-certs.tar"\
--qpid-router=false\
--puppet=true\
--puppetca=true\
--reverse-proxy=true
```

For more information and how to configure a BMC interface, see [To Add a BMC Interface](#) in the [Red Hat Satellite 6.1 User Guide](#)

7.5.3. Provisioning Options for Capsule Server

There are several options that provide provisioning services such as TFTP, DHCP, DNS and Realm that can be added to either type of Capsule Server. For a full list of options, use the command:

```
# capsule-installer --help
```

Here is an example of installing a capsule server with full provisioning services:

```
# capsule-installer --tftp=true\
--foreman-oauth-key "xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
--foreman-oauth-secret "w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
--certs-tar "~/capsule.example.com-certs.tar"\
--templates=true\
--dhcp=true\
--dhcp-gateway=192.168.122.1\
--dhcp-nameservers=192.168.122.1\
--dhcp-range="192.168.122.100 192.168.122.200"\
--dhcp-interface=eth0\
--dns=true\
--dns-forwarders=8.8.8.8\
--dns-interface=eth0\
--dns-zone=example.com
```

Ensure the **dns-interface** argument is set with the correct network interface name for the DNS server to listen on. Also ensure that the **dhcp-interface** argument is set correctly with

the interface name for the DHCP server. After configuration, create a subnet on the Satellite server under **Infrastructure** → **Subnets** for the Capsule which registers automatically.



NOTE

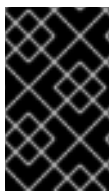
While it is possible to define the same DHCP range with the capsule-installer command and in the Satellite GUI, it is a good practice to make these ranges disjoint. In the Satellite GUI, select a range from outside the pool defined with capsule-installer, but still in the range defined on the subnet. For the example above, it is recommended to define the DHCP range from 192.168.122.1 to 192.168.122.99 in the Satellite GUI which gives the following IP address distribution:

- 192.168.122.1 to 192.168.122.99 (*reservation pool*) are addresses reserved during bare-metal provisioning by Satellite.
- 192.168.122.100 to 192.168.122.200 (*lease pool*) are addresses reserved for dynamic clients in the subnet (discovered hosts and unmanaged hosts).

It is possible to install a Satellite Capsule without installing DNS and DHCP services on the same server. Instead you can configure the Satellite Capsule to use external DNS and DHCP services as described in [Section 7.9, “Configuring Satellite 6 with External Services”](#). Alternatively, you can manually allocate specific IP addresses to host names or MAC addresses, see the DHCP chapter in the [Red Hat Enterprise Linux 7 Networking Guide](#)^[9].

7.6. ADDING LIFE CYCLE ENVIRONMENTS TO A RED HAT SATELLITE CAPSULE SERVER

If the newly created Red Hat Satellite Capsule Server has content functionality enabled, the Satellite Capsule Server needs an environment added to the Satellite Capsule Server. Adding an environment to the Red Hat Satellite Capsule Server will allow the Satellite Capsule Server to synchronize content from the Satellite Server and provide content to host systems.



IMPORTANT

The Satellite Capsule Server is configured through the Satellite Server's command line interface (CLI). Execute all **hammer** commands on the Satellite Server.

Procedure 7.4. To Add Environments to the Satellite Capsule Server:

1. Log in to the Satellite Server CLI as root.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **id**:

```
# hammer capsule list
```

The Satellite Capsule Server's details can be verified using the command:

```
# hammer capsule info --id capsule_id_number
```

3. Verify the list of life cycle environments available for the Red Hat Capsule Server and note down the **environment id**:

```
# hammer capsule content available-lifecycle-environments --id capsule_id_number
```

Where:

- **available-lifecycle-environments** are life cycle environments that are available to the Satellite Capsule but are currently not attached to the Satellite Capsule.

4. Add the life cycle environment to the Satellite Capsule Server:

```
# hammer capsule content add-lifecycle-environment --id capsule_id_number --environment-id environment_id_number
```

Where:

- *capsule_id_number* stands for the Satellite Capsule Server's identification number.
- *environment_id_number* stands for the life cycle environment's identification number.

Repeat this step for every life cycle environment to be added to the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

```
# hammer capsule content synchronize --id capsule_id_number
```

When an external Satellite Capsule Server has various life cycle environments, and only one life cycle environment needs to be synchronized, it is possible to target a specific environment by specifying the environment identification:

```
# hammer capsule content synchronize --id external_capsule_id_number --environment-id environment_id_number
```

7.7. REMOVING LIFE CYCLE ENVIRONMENTS FROM THE RED HAT SATELLITE CAPSULE SERVER

There are multiple reasons to remove life cycle environments from the Red Hat Satellite Capsule Server. For example:

- When life cycle environments are no longer relevant to the host systems
- When life cycle environments have been incorrectly added to the Satellite Capsule Server

Procedure 7.5. To remove a life cycle environment from the Satellite Capsule Server

Server:

1. Log in to the Satellite Server CLI as the root user.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **id**:

```
# hammer capsule list
```

The Satellite Capsule Server's details can be verified using the command:

```
# hammer capsule info --id capsule_id_number
```

3. Verify the list of life cycle environments currently attached to the Red Hat Capsule Server and take note of the **environment id**:

```
# hammer capsule content lifecycle-environments --id  
capsule_id_number
```

4. Remove the life cycle environment from the Satellite Capsule Server:

```
# hammer capsule content remove-lifecycle-environment --id  
capsule_id_number --environment-id environment_id
```

Where:

- *capsule_id_number* is the Satellite Capsule Server's identification number.
- *environment_id* is the life cycle environment's identification number.

Repeat this step for every life cycle environment to be removed from the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

```
# hammer capsule content synchronize --id capsule_id_number
```

7.8. REGISTERING HOST SYSTEMS TO A RED HAT SATELLITE CAPSULE SERVER

Prerequisites

The client system must be configured for registration. See the following chapters in the [Red Hat Satellite 6.1 User Guide](#) for information about configuring a client to register with a Red Hat Satellite Capsule:

- [Configuring Hosts](#)
- [Configuring Activation Keys](#)

Ensure the Satellite tools repository appropriate to the host to be registered is enabled and synchronized. If required, see [Procedure 8.3, "Enable New Red Hat Repositories"](#) and [Section 4.1.3, "Synchronizing Content"](#).

Register systems to a Satellite Capsule as follows:

Procedure 7.6. Registering Host Systems to the Capsule Server

1. In the web UI, select **Hosts** → **Content Hosts** and then click **Register Content Host**.
2. Choose the required Capsule Server in the **Content Source** drop-down list.
3. Connect to the host and install the bootstrap RPM:

```
# rpm -Uvh http://capsule.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

Where *capsule.example.com* is the host name of the Capsule to be used as the content source. If the Satellite Server's integrated Capsule is to be used, then use the Satellite Server's host name.

4. Run **subscription-manager** in a console on the client host.

- a. You can use an Activation Key to register:

```
# subscription-manager register --org=organization_label --activationkey="activation_key"
```

- b. Alternatively:

- authenticate with a user name and password:

```
# subscription-manager register --org=organization_label --environment="Library"
```

- and attach a subscription:

```
# subscription-manager list --available --all
# subscription-manager attach --pool=pool_ID
```

5. Enable the Satellite tools repository:

```
# subscription-manager repos --enable=rhel-version-server-satellite-tools-6.1-rpms
```

Replace *version* with **6** or **7** depending on the Red Hat Enterprise Linux version you are using.

6. Enable any additional repositories required for this host:

```
# subscription-manager repos --enable=repository-to-be-enabled
```

7. Install **katello-agent** for remote actions and displaying errata information:

```
# yum install katello-agent
```

Your content host is now registered to a Satellite Capsule Server.

7.9. CONFIGURING SATELLITE 6 WITH EXTERNAL SERVICES

By default, the Capsule installer installs and configures the **TFTP** service available in Red Hat Enterprise Linux. It can optionally install **DNS** and **DHCP** services. If required to use Capsule with external services, prevent installation of the unwanted services by running the installer with the relevant options set to **false**.

Example 7.1. Installing Capsule Without Services

To install Capsule without the default installation of **TFTP**, enter a command as follows:

```
# katello-installer \  
--capsule-tftp false
```

If Capsule has already been installed, execute the installer again with the relevant options set to **false** to reset the configuration files back to the desired state. This will not uninstall the packages for the services, such as `bind` or `tftp-server`. If required, uninstall the unused packages manually.

Example 7.2. Reinstalling Capsule Without Services

To install Capsule without installing **DNS**, **DHCP**, and **TFTP**, enter a command as follows:

```
# katello-installer \  
--capsule-dns false \  
--capsule-dns-managed false \  
--capsule-dhcp false \  
--capsule-dhcp-managed false \  
--foreman-proxy-tftp false
```

IMPORTANT

These procedures were written and tested on Red Hat Enterprise Linux 7.1. They are based on the use of **NFSv3**. The procedures should work for other releases, such as Red Hat Enterprise Linux 6 or Red Hat Enterprise Linux 7.0, but note there may be differences in **NFS** exporting. See the [Red Hat Enterprise Linux 7 Storage Administration Guide](#) and [Red Hat Enterprise Linux 6 Storage Administration Guide](#) for more information on exporting file systems using **NFS**.

In the example configurations below, the subnet is **192.168.38.0/24**, the domain is called **virtual.lan**, the server for the external services is **192.168.38.2/24**, and the Capsule Server is at **192.168.38.1/24**.

7.9.1. Configuring an External DNS Service

Deploy a Red Hat Enterprise Linux Server (Red Hat Enterprise Linux 7.1 or later is recommend) and install the ISC DNS service (packages `bind` and `bind-utils` are required):

```
# yum install bind bind-utils
```


Procedure 7.7. Configuring the External DNS Server

Configure the external **DNS** server as follows:

1. Create the configuration for the domain with a configuration similar to the following:

```
# cat /etc/named.conf
include "/etc/rndc.key";

controls {
    inet 192.168.38.2 port 953 allow { 192.168.38.1; 192.168.38.2; }
    keys { "capsule"; };
};

options {
    directory "/var/named";
    forwarders { 8.8.8.8; 8.8.4.4; };
};

include "/etc/named.rfc1912.zones";

zone "38.168.192.in-addr.arpa" IN {
    type master;
    file "dynamic/38.168.192-rev";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};

zone "virtual.lan" IN {
    type master;
    file "dynamic/virtual.lan";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};
```

Note that the **inet** line must be entered as one line in the configuration file.

The example above configures a domain **virtual.lan** as one subnet **192.168.38.0/24**, a security key named **foreman**, and sets forwarders to Google's public **DNS** addresses (**8.8.8.8** and **8.8.4.4**).

2. Create a key file:

```
# ddns-confgen -k capsule
```

The above command can take a long time as the program is reading a pseudo random device. For testing or proof-of-concept deployments, an insecure non-blocking device can be used as follows:

```
# ddns-confgen -k capsule -r /dev/urandom
```

3. The above command will print the key section with some instructions as comments. Copy and paste the key section into a separate file named **/etc/rndc.key**, which is included by a statement in **named.conf**, so that the file looks as follows:

```
# cat /etc/rndc.key
key "capsule" {
    algorithm hmac-sha256;
    secret "GeBbgGoLedEAAwNQPtPh3zP56MJbkwM84UJDtaUS9mw=";
};
```

This is the secret key that is used to change **DNS** server configuration, keep it safe and make sure only **root** can read and write it. This file will be copied over to Capsule server in a later step.

4. Create zone files as follows:

```
# cat /var/named/dynamic/virtual.lan
$ORIGIN .
$TTL 10800      ; 3 hours
virtual.lan     IN SOA  service.virtual.lan.
root.virtual.lan. (
                        9           ; serial
                        86400        ; refresh (1 day)
                        3600         ; retry (1 hour)
                        604800       ; expire (1 week)
                        3600         ; minimum (1 hour)
                    )
                        NS           service.virtual.lan.
$ORIGIN virtual.lan.
$TTL 86400      ; 1 day
capsule         A       192.168.38.1
service         A       192.168.38.2
```

5. Create the reverse zone file:

```
# cat /var/named/dynamic/38.168.192-rev
$ORIGIN .
$TTL 10800      ; 3 hours
38.168.192.in-addr.arpa IN SOA  service.virtual.lan.
root.38.168.192.in-addr.arpa. (
                        4           ; serial
                        86400        ; refresh (1 day)
                        3600         ; retry (1 hour)
                        604800       ; expire (1 week)
                        3600         ; minimum (1 hour)
                    )
                        NS           service.virtual.lan.
$ORIGIN 38.168.192.in-addr.arpa.
$TTL 86400      ; 1 day
1               PTR     capsule.virtual.lan.
2               PTR     service.virtual.lan.
```



IMPORTANT

Make sure there are no extra non-US-ASCII characters as BIND is sensitive to this.

Procedure 7.8. Testing and Starting the DNS Service

To test the configuration and start the **DNS** service, proceed as follows:

1. Validate the syntax as follows:

```
# named-checkconf -z /etc/named.conf
```

2. Start the server:

- On Red Hat Enterprise Linux 7:

```
# systemctl restart named
```

- On Red Hat Enterprise Linux 6:

```
# service named restart
```

3. Try to add a new host dynamically:

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. Test that the **DNS** service can resolve the new host added in the previous step:

```
# nslookup aaa.virtual.lan 192.168.38.2
```

5. If required, delete the new entry:

```
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

6. Configure the firewall for external access to the **DNS** service (**UDP** and **TCP** on port 53):

- On a Red Hat Enterprise Linux 6 Satellite, execute as root:

```
# iptables -A INPUT -m state --state NEW -p udp --dport 53 -j
ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

- On a Red Hat Enterprise Linux 7 Satellite, execute as root:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
&& firewall-cmd --permanent --add-port="53/udp" --add-
port="53/tcp"
```

Procedure 7.9. Configuring a Capsule Server to Use an External DNS Service

To configure a Capsule Server to use an external **DNS** service, proceed as follows:

1. Ensure that the **nsupdate** utility, from the **bind-utils** package, is installed:

```
# yum install bind-utils
```

2. Copy the **/etc/rndc.key** file from the services server to the Capsule Server. For example:

- On the services server:

```
scp localfile username@hostname:remotefile
```

- Alternatively, on the Capsule Server:

```
scp username@hostname:remotefile localfile
```

3. Make sure the key file has the correct owner, permissions, and SELinux label:

```
# ls -l /etc/rndc.key
-rw-r----- . root named system_u:object_r:dnsssec_t:s0
/etc/rndc.key
```

4. The Capsule uses the **nsupdate** utility to update **DNS** records on the remote server. Before configuring it, test adding one additional host remotely as follows:

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
# nslookup aaa.virtual.lan 192.168.38.2
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

5. Run the **katello-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/dns.yml** file. Enable the smart-proxy module setting provider to be **nsupdate**, add the address to the **DNS** server, and set the default time to live for records created by this Capsule. For example:

```
# katello-installer --foreman-proxy-dns=true --foreman-proxy-dns-
managed=false --foreman-proxy-dns-provider=nsupdate --foreman-proxy-
```

```
dns-server="192.168.38.2" --foreman-proxy-keyfile=/etc/rndc.key --
foreman-proxy-dns-ttl=86400
```

6. Restart foreman-proxy service:

- On Red Hat Enterprise Linux 7:

```
# systemctl restart foreman-proxy
```

- On Red Hat Enterprise Linux 6:

```
# service foreman-proxy restart
```

7. View the Satellite Server GUI in your browser;

https://satellite_host.example.com.

8. Select **Infrastructure** → **Capsules**. Locate the Capsule being configured and select **Refresh features** from the drop-down list. The **DNS** feature should appear.

9. Select **Infrastructure** → **Capsules** and associate the **DNS** service with the appropriate subnets and domain.

7.9.2. Configuring an External DHCP Service

Deploy a Red Hat Enterprise Linux Server (Red Hat Enterprise Linux 7.1 or later is recommend) and install the ISC DHCP server package `dhcp`.

```
# yum install dhcp
```



NOTE

External DHCP configuration via NFS is no longer supported starting from Satellite 6.3. Due to optimizations via inotify, DHCP Capsule will no longer detect changes in remote files.

Procedure 7.10. Configuring the External DHCP Server

Configure the external **DHCP** server as follows:

1. Generate a security token in an empty directory as follows:

```
# dnssec-keygen -a HMAC-MD5 -b 512 -n HOST omapi_key
```

The above command can take a long time, for less-secure proof-of-concept deployments you can use a non-blocking random number generator:

```
# dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
```

This will create the key pair in two files in the current directory.

2. Copy the secret hash from the key:

```
# cat Komapi_key.+.private |grep ^Key|cut -d ' ' -f2
```

3. Edit the **dhcpcd** configuration file for all the subnets, and add the secret key from the previous step:

```
# cat /etc/dhcp/dhcpd.conf
default-lease-time 604800;
max-lease-time 2592000;
log-facility local7;

subnet 192.168.38.0 netmask 255.255.255.0 {
    range 192.168.38.10 192.168.38.100;
    option routers 192.168.38.1;
    option subnet-mask 255.255.255.0;
    option domain-search "virtual.lan";
    option domain-name "virtual.lan";
    option domain-name-servers 8.8.8.8;
}

omapi-port 7911;
key omapi_key {
    algorithm HMAC-MD5;
    secret "jNSE5YI3H1A80j/tkV4...A2Z0Hb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

4. Delete the two key files from the directory where you created them.
5. For each subnet defined (**192.168.38.0** in this example) define **Subnet** on the Satellite server. It is recommended to set up a lease range and reservation range separately to prevent conflicts. In this example, the lease range is **192.168.38.10** to **192.168.38.100** so the reservation range (defined in Satellite GUI) would be **192.168.38.101** to **192.168.38.250**. Do not set **DHCP Capsule** for the defined **Subnet** yet.

Note that ISC DHCP listens only on interfaces that match defined subnets. In this example, the server has an interface that routes to **192.168.38.0** subnet directly.

6. Configure the firewall for external access to the **DHCP** service:

- On a Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-service dhcp \
&& firewall-cmd --permanent --add-service dhcp
```

- On a Red Hat Enterprise Linux 6:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 67 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

7. Configuration files are read by **foreman-proxy** user, first determine the UID and GID

numbers of the **foreman-proxy** user on the Capsule Server, then create the same user and group with same IDs on this server:

```
# groupadd -g 990 foreman-proxy
# useradd -u 992 -g 990 -s /sbin/nologin foreman-proxy
```

8. Configuration files must be readable for this user. Recent dhcp package updates removed read and execute flags from the configuration directory which prevents that. To restore the required flags and prevent this change in behavior on the next package update, enter the following commands:

```
# chmod o+rx /etc/dhcp/
# chmod o+r /etc/dhcp/dhcpd.conf
# chattr +i /etc/dhcp/ /etc/dhcp/dhcpd.conf
```

9. Start the **DHCP** service:

- On Red Hat Enterprise Linux 7:

```
# systemctl start dhcpd
```

- On Red Hat Enterprise Linux 6:

```
# service dhcpd start
```

10. Export **DHCP** configuration and leases file using **NFS**, so that the Capsule Server can read it:

```
# yum install nfs-utils
# systemctl enable rpcbind nfs-server
# systemctl start rpcbind nfs-server nfs-lock nfs-idmapd
```

11. Create the **DHCP** configuration and leases files to be exported using **NFS**:

```
# mkdir -p /exports/var/lib/dhcpd /exports/etc/dhcp
```

12. Add the newly created mount point to **/etc/fstab** file:

```
/var/lib/dhcpd /exports/var/lib/dhcpd none bind,auto 0 0
/etc/dhcp /exports/etc/dhcp none bind,auto 0 0
```

13. Mount the file systems in **/etc/fstab**:

```
# mount -a
```

14. Ensure the following lines are present in **/etc/exports**:

```
/exports
192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)

/exports/etc/dhcp
192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

```
/exports/var/lib/dhcpd
192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

15. Reload the **NFS** server:

```
# exportfs -rva
```

16. Configure the firewall for the **DHCP omapi** port **7911** for the Capsule Server:

- On a Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-port="7911/tcp" \
&& firewall-cmd --permanent --add-port="7911/tcp"
```

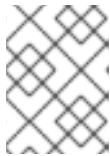
- On a Red Hat Enterprise Linux 6:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 7911 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

17. This step is common to both the **DHCP** and **TFTP** procedures and need only be carried out once per system. If required, follow this step to configure the firewall for external access to the **NFS** service.



NOTE

In this guide the clients are configured to use **NFSv3** and this step is therefore **NFSv3** specific.

- On Red Hat Enterprise Linux 7:

It is recommended to use **firewalld** daemon's **NFS** service option because **NFS** uses multiple ports to initiate connections. To do so, enter the following commands:

```
# firewall-cmd --zone public --add-service mountd \
&& firewall-cmd --zone public --add-service rpc-bind \
&& firewall-cmd --zone public --add-service nfs \
&& firewall-cmd --permanent --zone public --add-service mountd \
&& firewall-cmd --permanent --zone public --add-service rpc-bind \
&& firewall-cmd --permanent --zone public --add-service nfs
```

For additional information on using **NFSv3** behind a firewall on Red Hat Enterprise Linux 7, see the “Running NFS Behind a Firewall” section in the [Red Hat Enterprise Linux 7 Storage Administration Guide](#) and the “Securing NFS” section in the [Red Hat Enterprise Linux 7 Security Guide](#).

- On Red Hat Enterprise Linux 6:

Configure ports for **NFSv3** in the `/etc/sysconfig/nfs` file as follows:

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Restart the service for the changes to take effect:

```
# service nfs restart
```

Add the following rules to the `/etc/sysconfig/iptables` file by entering commands as follows:

```
# iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp
--dport 111 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 111 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 2049 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 2049 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 32803 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 32769 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 892 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 892 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 875 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 875 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 662 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 662 -j ACCEPT \
&& service iptables save
```

Restart the firewall for the changes to take effect:

```
# service iptables restart
```

For additional information on using **NFSv3** behind a firewall on Red Hat Enterprise Linux 6, see the [Red Hat Enterprise Linux 6 Storage Administration Guide](#) and the “Running NFS Behind a Firewall” section in the “Securing NFS” section in the [Red Hat Enterprise Linux 6 Security Guide](#).

Procedure 7.11. Configuring a Capsule Server to Use an External DHCP Service

To configure a Capsule Server to use an external **DHCP** service, proceed as follows:

1. Install the **NFS** client:

```
# yum install nfs-utils
```

2. Create the **DHCP** directories to prepare for **NFS**:

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

3. Change the file owner as follows:

```
# chown -R foreman-proxy /mnt/nfs
```

4. Try to reach the **NFS** server and verify RPC communication paths:

```
# showmount -e 192.168.38.2
# rpcinfo -p 192.168.38.2
```

5. Add these **two** lines to the **/etc/fstab** file:

```
192.168.38.2:/exports/etc/dhcp /mnt/nfs/etc/dhcp nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcp_etc_t:s0"
0 0

192.168.38.2:/exports/var/lib/dhcpd /mnt/nfs/var/lib/dhcpd nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcpd_state_t:s0"
0 0
```

6. Mount the file systems in **/etc/fstab**:

```
# mount -a
```

7. Try to read the relevant files:

```
# su foreman-proxy -s /bin/bash
bash-4.2$ cat /mnt/nfs/etc/dhcp/dhcpd.conf
bash-4.2$ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases
bash-4.2$ exit
```

In case of problems, investigate the **NFS** configuration, logs, and firewall rules.

8. On the Capsule Server, run the **katello-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/dhcp.yml** file:

```
# katello-installer --foreman-proxy-dhcp=true --foreman-proxy-dhcp-
provider=isc --foreman-proxy-dhcp-config
/mnt/nfs/etc/dhcp/dhcpd.conf --foreman-proxy-dhcp-leases
/mnt/nfs/var/lib/dhcpd/dhcpd.leases --foreman-proxy-dhcp-key-
name=omapi_key --foreman-proxy-dhcp-key-
secret=jNSE5YI3H1A80j/tkV4...A2Z0Hb6zv315CkNAY7DMYYCj48Umw== --
foreman-proxy-dhcp-server dhcp.example.com
```

Ensure the **dhcp_key_secret** value is correctly entered without quotes. The trailing **=** character is optional.

9. Restart the proxy:

- On Red Hat Enterprise Linux 7:

```
# systemctl restart foreman-proxy
```

- On Red Hat Enterprise Linux 6:

```
# service foreman-proxy restart
```

10. View the Satellite Server GUI in your browser;

https://satellite_host.example.com.

11. Select **Infrastructure** → **Capsules**. Locate the Capsule and select **Refresh features** from the drop-down list. The **DHCP** feature should appear.

12. Select **Infrastructure** → **Capsules** and associate the **DHCP** service with the appropriate subnets and domain.

7.9.3. Configuring an External TFTP Service

Deploy a Red Hat Enterprise Linux Server (Red Hat Enterprise Linux 7.1 or later is recommend).

Procedure 7.12. Configuring the TFTP Server

Configure the external **TFTP** server as follows:

1. Install and enable the **TFTP** server:

```
# yum install tftp-server syslinux
```

- On Red Hat Enterprise Linux 7, enable and activate the **tftp.socket** unit:

```
# systemctl enable tftp.socket
# systemctl start tftp.socket
```

- On Red Hat Enterprise Linux 6, enable and start the **xinetd** service:

```
# service xinetd enable
# service xinetd start
```

2. Configure the PXELinux environment as follows:

```
# mkdir -p /var/lib/tftpboot/{boot,pxelinux.cfg}
# cp /usr/share/syslinux/{pxelinux.0,menu.c32,chain.c32}
/var/lib/tftpboot/
```

3. Restore SELinux file contexts:

```
# restorecon -RvF /var/lib/tftpboot/
```

4. Create the **TFTP** directory to be exported using **NFS**:

```
# mkdir -p /exports/var/lib/tftpboot
```

5. Add the newly created mount point to the **/etc/fstab** file:

```
/var/lib/tftpboot /exports/var/lib/tftpboot none bind,auto 0 0
```

6. Mount the file systems in **/etc/fstab**:

```
# mount -a
```

7. Ensure the following lines are present in **/etc/exports**:

```
/exports  
192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)
```

```
/exports/var/lib/tftpboot  
192.168.38.1(rw,async,no_root_squash,no_subtree_check,nohide)
```

The first line is common to the **DHCP** configuration and therefore should already be present if the previous procedure was completed on this system.

8. Reload the **NFS** server:

```
# exportfs -rva
```

9. This step is common to both the **DHCP** and **TFTP** procedures and need only be carried out once per system. If required, follow this step to configure the firewall for external access to the **NFS** service.



NOTE

In this guide the clients are configured to use **NFSv3** and this step is therefore **NFSv3** specific.

- On Red Hat Enterprise Linux 7:

It is recommended to use **firewalld** daemon's **NFS** service option because **NFS** uses multiple ports to initiate connections. To do so, enter the following commands:

```
# firewall-cmd --zone public --add-service mountd \  
&& firewall-cmd --zone public --add-service rpc-bind \  
&& firewall-cmd --zone public --add-service nfs \  
&& firewall-cmd --permanent --zone public --add-service mountd \  
&& firewall-cmd --permanent --zone public --add-service rpc-bind \  
\   
&& firewall-cmd --permanent --zone public --add-service nfs
```

For additional information on using **NFSv3** behind a firewall on Red Hat Enterprise Linux 7, see the “Running NFS Behind a Firewall” section in the [Red Hat Enterprise Linux 7 Storage Administration Guide](#) and the “Securing NFS” section in the [Red Hat Enterprise Linux 7 Security Guide](#).

- On Red Hat Enterprise Linux 6:

Configure ports for **NFSv3** in the `/etc/sysconfig/nfs` file as follows:

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Restart the service for the changes to take effect:

```
# service nfs restart
```

Add the following rules to the `/etc/sysconfig/iptables` file by entering commands as follows:

```
# iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p udp
--dport 111 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 111 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 2049 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 2049 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 32803 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 32769 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 892 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 892 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 875 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 875 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 662 -j ACCEPT \
&& iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 662 -j ACCEPT \
&& service iptables save
```

Restart the firewall for the changes to take effect:

```
# service iptables restart
```

For additional information on using **NFSv3** behind a firewall on Red Hat Enterprise Linux 6, see the [Red Hat Enterprise Linux 6 Storage Administration](#)

[Guide](#) and the “Running NFS Behind a Firewall” section in the “Securing NFS” section in the [Red Hat Enterprise Linux 6 Security Guide](#).

Procedure 7.13. Configure the Firewall for External access to the TFTP service

- Configure the firewall for external access to the **TFTP** service (**UDP** on port 69):
 - On a Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-port="69/udp" \
&& firewall-cmd --permanent --add-port="69/udp"
```

- On a Red Hat Enterprise Linux 6:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 69 -j
ACCEPT \
&& service iptables save
```

Make sure the **iptables** service is started and enabled:

```
# service iptables start
# chkconfig iptables on
```

Procedure 7.14. Configuring a Capsule Server to Use an External TFTP Service

To configure a Capsule Server to use an external **TFTP** service, proceed as follows:

1. Create the **TFTP** directory to prepare for **NFS**:

```
# mkdir -p /mnt/nfs/var/lib/tftpboot
```

2. In the **/etc/fstab**, add a line as follows:

```
192.168.38.2:/exports/var/lib/tftpboot /mnt/nfs/var/lib/tftpboot nfs
rw,vers=3,auto,nosharecache,context="system_u:object_r:tftpd_dir_rw_t:
s0" 0 0
```

3. Mount the file systems in **/etc/fstab**:

```
# mount -a
```

4. Run the **katello-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/tftp.yml** file:

```
# katello-installer --foreman-proxy-tftp=true --foreman-proxy-tftp-
root /mnt/nfs/var/lib/tftpboot
```

If the **TFTP** service is running on a different server than the **DHCP** service, update the **tftp_servername** setting with the FQDN or **IP** address of that server.

```
# katello-installer --foreman-proxy-tftp-servername=new_FQDN
```

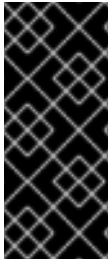
5. View the Satellite Server GUI in your browser;
`https://satellite_host.example.com`.
6. Select **Infrastructure** → **Capsules** in the user interface. Locate the Capsule and select **Refresh features** from the drop-down list. The **TFTP** feature should appear.
7. Select **Infrastructure** → **Capsules** and associate the **TFTP** service with the appropriate subnets and domain.

[8] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_OpenSSL.html

[9] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/

CHAPTER 8. UPGRADING RED HAT SATELLITE SERVER AND CAPSULE SERVER

The Satellite Server and Capsule Servers are upgraded independently. Upgrade the Satellite server first, and then upgrade any Capsules. Satellite 6.0 Capsules are not compatible with Satellite 6.1, and must be upgraded before attempting to synchronize any repositories. You must also manually upgrade Satellite clients to the new version of katello-agent after upgrading the Server and Capsules.

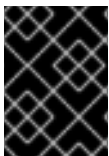


IMPORTANT

The Red Hat Satellite server and Capsule server versions must match. For example, a Satellite 6.0 server cannot run a 6.1 Capsule server and a Satellite 6.1 server cannot run a 6.0 Capsule server. Mismatching Satellite server and Capsule server versions will result in the Capsule server failing silently.

Supported upgrade paths for Satellite 6.1 GA:

- Satellite 6.0 to Satellite 6.1
- Satellite 6.1 Public Beta (non-production) to Satellite 6.1 GA



IMPORTANT

Upgrading from Satellite 6.1 Public Beta in a production environment to Satellite 6.1 is not supported.

The following conditions must be met before upgrading Red Hat Satellite 6:

- Verify that the Satellite has the **6.1 satellite-tools** and **capsule** repositories fully synchronized and available to update the Satellite Capsule servers to the latest upgrade package versions.
- Ensure that the existing Content Views are updated to include the newly synchronized repositories. If you use Activation Keys for content host registration, ensure that your Activation Key is updated with the newly synchronized repositories. If you created a new Content View for these repositories, include this Content View in the Activation Key. See the [Red Hat Satellite 6.1 User Guide](#)^[10] for more information on Activation Keys.
- Refresh subscriptions to include the newly synchronized repositories both on Capsules and Hosts.
- In the Satellite web UI, navigate to **Monitor → Tasks** and check for running tasks. It is recommended that you wait for the tasks to complete. It is possible to cancel some tasks, but you should follow the guidance in the Red Hat Knowledgebase solution [How to manage paused tasks on Red Hat Satellite 6](#) to understand which tasks are safe to cancel and which are not safe to cancel.

8.1. UPGRADING RED HAT SATELLITE

This section shows how to upgrade Red Hat Satellite from version 6.0 or 6.1 Public Beta (non-production) to 6.1.

Prerequisites

Upgrade to the latest minor version of Red Hat Satellite 6.0 before proceeding. Direct upgrade to 6.1 from earlier minor versions is not supported.

The Red Hat Satellite 6.1 release requires the Red Hat Satellite 6.1 Tools repository to be available in the subscription manifest. This repository provides the katello-agent and puppet-agent packages for clients registered to the Satellite Server. Ensure the required repositories are enabled by following the procedure below to update the subscription manifest. Remove any that are no longer required.

Procedure 8.1. Updating the Subscription Manifest

This procedure describes updating the subscription manifest.

1. Navigate to <https://access.redhat.com> and click **SUBSCRIPTIONS** on the main menu at the top of the page.
2. Scroll down to the **Red Hat Subscription Management** section, and click **Satellite** under **Subscription Management Applications**.
3. Click the name of the system this manifest is associated to, and click **Attach a subscription**.
4. For each subscription that you want to attach, select the check box for that subscription, and specify the quantity of subscriptions to attach.
5. Click **Attach Selected**. It can take several minutes for all the subscriptions to attach. Refresh the screen every few minutes until you receive confirmation that the subscriptions are attached.
6. After the subscriptions have been attached, click **Download Manifest** to generate an archive in **.zip** format containing the manifest for Red Hat Satellite and save the manifest file to a known location.
7. Upload the updated manifest to the Red Hat Satellite Server.
 - a. Log in to the **Satellite** server.
 - b. In the top left corner menu, select the organization that you want to associate with the subscription manifest.
 - c. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest** at the upper right of the page.
 - d. In the **Subscription Manifest** section, click **Actions** and under the **Upload New Manifest** subsection, click **Browse**.
 - e. Select the manifest file to upload, and then click **Upload**.

Procedure 8.2. Upgrading Red Hat Satellite

1. If the Satellite server is running on a virtual machine, take a snapshot prior to upgrading. Otherwise, run **katello-service stop** and create a backup of the relevant databases. See [How to generate database backup for Red Hat Satellite 6.0](#)

for instructions on backing up your databases.

2. Update the operating system:

```
# yum update
```

3. Disable the repositories for the previous version of Satellite.

- If upgrading from Satellite 6.0 on Red Hat Enterprise Linux 7:

```
# subscription-manager repos --disable rhel-7-server-satellite-6.0-rpms
```

- If upgrading from Satellite 6.1 Beta on Red Hat Enterprise Linux 7:

```
# subscription-manager repos --disable rhel-server-7-satellite-6-beta-rpms
```

- If upgrading from Satellite 6.0 on Red Hat Enterprise Linux 6:

```
# subscription-manager repos --disable rhel-6-server-satellite-6.0-rpms
```

- If upgrading from Satellite 6.1 Beta on Red Hat Enterprise Linux 6:

```
# subscription-manager repos --disable rhel-server-6-satellite-6-beta-rpms
```

4. Enable the new repositories.

- On Red Hat Enterprise Linux 7:

```
# subscription-manager repos --enable rhel-7-server-satellite-6.1-rpms
```

- On Red Hat Enterprise Linux 6:

```
# subscription-manager repos --enable rhel-6-server-satellite-6.1-rpms
```

5. If there are discovered hosts available, turn them off and delete all entries under the **Discovered hosts** page.

6. Stop services:

```
# katello-service stop  
output omitted  
Success!
```

Wait for the command to complete. If required, confirm services have stopped:

```
# katello-service status
```

```

mongod is stopped
qdrouterd is stopped
qpidd is stopped
celery init v10.0.
Using configuration: /etc/default/pulp_workers,
/etc/default/pulp_celerybeat
pulp_celerybeat is stopped.
elasticsearch is stopped
celery init v10.0.
Using config script: /etc/default/pulp_resource_manager
node resource_manager is stopped...
foreman-proxy is stopped
tomcat6 is stopped [ OK ]
output truncated

```

Restart the Mongo database as it is required for upgrading the Pulp database:

```
# service-wait mongod start
```

7. Clear the repository cache and update all packages:

```
# yum clean all
# yum update
```

8. Run the installer with the **--upgrade** option:

```
# katello-installer --upgrade
```

If required, add the **--noop** option to the command and review the **/var/log/katello-installer/katello-installer.log** to see what changes would be applied if the **--noop** was omitted.



IMPORTANT

If you have made manual edits to DNS and DHCP configuration files, they will be overwritten during the upgrade process. To avoid this, append the **--capsule-dns-managed=false** and **--capsule-dhcp-managed=false** options to the **--upgrade** installer command.

The **katello-installer** utility will backup files that it changes and log this. For example:

```
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed
/etc/dhcp/dhcpd.conf to puppet with sum
622d9820b8e764ab124367c68f5fa3a1
```

The old file can be restored with this command:

```
# puppet filebucket -l restore /etc/dhcp/dhcpd.conf
622d9820b8e764ab124367c68f5fa3a1
```

9. Restart all services:

```
# katello-service restart
```

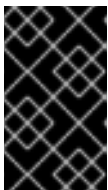
If you are using the Discovery feature, you must also complete [Section 8.3, “Upgrading the Discovery Feature”](#)

Enabling The New Repositories

The Red Hat Satellite manifest file provides access to Red Hat products and repositories. Any new repositories must be enabled and synchronized in Red Hat Satellite Server to prepare them for use by Red Hat Satellite Capsule Servers.

Procedure 8.3. Enable New Red Hat Repositories

1. On the main menu, click **Content** → **Red Hat Repositories** and then click the tab for the type of content that you want to enable.
2. Click the product name for which you want to add repositories. This expands the list of available repository sets.
3. Click each repository set from which you want to select repositories, and select the check box for each required repository. The repository is automatically enabled. After enabling a Red Hat repository, a product for this repository is automatically created. The content from this repository will be downloaded during the next synchronization.



IMPORTANT

Ensure you enable the Satellite Tools repository. This repository provides the `katello-agent` and `puppet-agent` packages for clients registered to the Satellite Server.

4. Start the synchronization process as described in [Section 4.1.3, “Synchronizing Content”](#).

8.1.1. Upgrading Disconnected Satellite

This section shows how to upgrade a disconnected Red Hat Satellite instance.

Prerequisites

- Upgrade to the latest minor version of Red Hat Satellite 6.0 before proceeding. Direct upgrade to 6.1 from earlier minor versions is not supported.
- Run **katello-service start** to restart all services and update the operating system. For instructions on how to update a disconnected system see [Deployment Guide](#)^[11] for Red Hat Enterprise Linux 6 or [System Administrator's Guide](#)^[12] for Red Hat Enterprise Linux 7.

Procedure 8.4. Upgrading Disconnected Satellite

1. If there are discovered hosts available, turn them off and delete all entries under the **Discovered hosts** page.
2. Stop services:

```
# katello-service stop
output omitted
Success!
```

Wait for the command to complete. If required, confirm services have stopped:

```
# katello-service status
mongod is stopped
qdrouterd is stopped
qpidd is stopped
celery init v10.0.
Using configuration: /etc/default/pulp_workers,
/etc/default/pulp_celerybeat
pulp_celerybeat is stopped.
elasticsearch is stopped
celery init v10.0.
Using config script: /etc/default/pulp_resource_manager
node resource_manager is stopped...
foreman-proxy is stopped
tomcat6 is stopped [ OK ]
output truncated
```

Restart the Mongo database as it is required for upgrading the Pulp database:

```
# service-wait mongod start
```

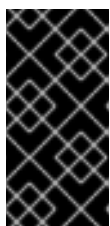
3. Obtain the ISO file, mount it, and run the `install_packages` script as described in [Section 2.1.2, “Downloading from a Disconnected Network”](#). After executing successfully, the script returns the following message:

```
Upgrade is complete. Please backup your data and run katello-
installer.
```

4. Create a backup of the relevant databases. See [How to generate database backup for Red Hat Satellite 6.0](#) for instructions on backing up your databases.
5. Run the installer with the **--upgrade** option:

```
# katello-installer --upgrade
```

If required, add the **--noop** option to the command and review the `/var/log/katello-installer/katello-installer.log` to see what changes would be applied if the **--noop** was omitted.



IMPORTANT

If you have made manual edits to DNS and DHCP configuration files, they will be overwritten during the upgrade process. To avoid this, append the **--capsule-dns-managed=false** and **--capsule-dhcp-managed=false** options to the **--upgrade** installer command.

The **katello-installer** utility will backup files that it changes and log this. For example:

```
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed
/etc/dhcp/dhcpd.conf to puppet with sum
622d9820b8e764ab124367c68f5fa3a1
```

The old file can be restored with this command:

```
# puppet filebucket -l restore /etc/dhcp/dhcpd.conf
622d9820b8e764ab124367c68f5fa3a1
```

6. Restart all services:

```
# katello-service restart
```

7. Update the Discovery template:

- a. At the **Hosts** tab, select **Provisioning templates**.
- b. Select **PXELinux global default**.
- c. At the **Template editor** dialog box, in the **Provisioning Template** tab, modify the **PXELinux global default** template discovery menu entry. Insert the following text at the end of the template:

```
LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2
```

The **proxy.type** option can be either **proxy** or **foreman**. For **proxy**, all communication goes through the Capsule. For **foreman**, the communication goes directly to Satellite Server, which was the behavior in Satellite 6.0.

The **proxy.url** specifies the URL of the Satellite Capsule or Server. Both HTTP and HTTPS protocols are supported.

8.2. UPGRADING RED HAT SATELLITE CAPSULE

Procedure 8.5. To Upgrade Red Hat Satellite Capsule:

1. Update the operating system:

```
# yum update
```

2. Disable the repositories for the previous version of Satellite.

- If upgrading from Satellite 6.0 on Red Hat Enterprise Linux 7:

```
# subscription-manager repos --disable rhel-7-server-satellite-  
capsule-6.0-rpms
```

- If upgrading from Satellite 6.1 Beta on Red Hat Enterprise Linux 7:

```
# subscription-manager repos --disable rhel-server-7-satellite-  
capsule-6-beta-rpms
```

- If upgrading from Satellite 6.0 on Red Hat Enterprise Linux 6:

```
# subscription-manager repos --disable rhel-6-server-satellite-  
capsule-6.0-rpms
```

- If upgrading from Satellite 6.1 Beta on Red Hat Enterprise Linux 6:

```
# subscription-manager repos --disable rhel-server-6-satellite-  
capsule-6-beta-rpms
```

3. Enable the new repositories.

- On Red Hat Enterprise Linux 7:

```
# subscription-manager repos --enable rhel-7-server-satellite-  
capsule-6.1-rpms
```

- On Red Hat Enterprise Linux 6:

```
# subscription-manager repos --enable rhel-6-server-satellite-  
capsule-6.1-rpms
```

4. If there are discovered hosts available, turn them off and delete all entries under the **Discovered hosts** page.

5. Stop the following services to prevent dependency errors during the database migration:

```
# for i in qpid pulp_workers pulp_celerybeat pulp_resource_manager  
httpd; do service $i stop; done
```

6. Clear the repository cache and update all packages:

```
# yum clean all  
# yum update
```

7. The following steps are required only if you upgrade from Satellite 6.0:

- a. Install the capsule-installer package:

```
# yum install capsule-installer
```

**NOTE**

In Red Hat Satellite 6.0, the **katello-installer** script provided the Satellite Capsule Server installer. In Satellite 6.1, the **capsule-installer** script has its own package.

Installing capsule-installer automatically removes the katello-installer package and saves the previous Capsule configuration and answer files.

- b. Copy the previous answer file to the new **capsule-installer** directory:

```
# cp /etc/katello-installer/answers.capsule-  
installer.yaml.rpmsave /etc/capsule-installer/answers.capsule-  
installer.yaml
```

8. On the Satellite Server, generate an archive with new certificates:

```
# capsule-certs-generate --capsule-fqdn "capsule.example.com" --  
certs-tar "capsule.example.com-certs.tar"
```

Replace *capsule.example.com* with the fully qualified domain name of the Capsule. Copy the archive file to the Capsule.

9. Install the Discovery plug-in if you plan to use the Capsule as a proxy for discovered hosts:

```
# yum install rubygem-smart_proxy_discovery.noarch
```

10. Verify if the `foreman_url` setting refers to the Satellite Server correctly. On the Capsule execute:

```
# grep foreman_url /etc/foreman-proxy/settings.yml
```

The above command should return the fully qualified domain name (FQDN) of the Satellite server, for example:

```
:foreman_url: https://satellite.example.com
```

11. Restart the foreman-proxy component on the Satellite Capsule server:

```
# service foreman-proxy restart
```

12. Run the installer with the **--upgrade** option:

```
# capsule-installer --upgrade --certs-tar capsule.example.com-  
certs.tar
```

Replace *capsule.example.com-certs.tar* with the path to the certificate archive on the Capsule.



IMPORTANT

If you have made manual edits to DNS and DHCP configuration files, they will be overwritten during the upgrade process. To avoid this, append the **--dns-managed=false** and **--dhcp-managed=false** options to the **--upgrade** installer command.

13. Upgrade the `foreman-discovery-image` package on the Satellite server and turn on the hosts that were shut down prior the upgrade.

8.3. UPGRADING THE DISCOVERY FEATURE

The following steps describe how to upgrade the Discovery feature of Red Hat Satellite 6.

Procedure 8.6. How to Upgrade the Discovery Feature of Satellite 6

1. Verify that all relevant packages are up-to-date on the Satellite server:

```
# yum upgrade ruby193-rubygem-foreman_discovery
```

Restart the Satellite server if any packages were updated.

2. Upgrade the Discovery image on the Satellite Capsule that is either connected to the provisioning network with discovered hosts or provides TFTP services for discovered hosts.

```
# yum upgrade foreman-discovery-image
```

3. On the same instance, install the package which provides the Proxy service, and then restart **foreman-proxy** service. Discovered hosts in Satellite 6.1 are no longer required to have direct connection to Satellite Server.

```
# yum install rubygem-smart_proxy_discovery
# service foreman-proxy restart
```

4. All subnets with discovered nodes need this specified in Satellite Server so it connects via the Foreman Proxy. In the web UI, navigate to **Infrastructure** → **Capsules** and verify that the desired proxy lists the Discovery feature. If it does not, click **Refresh features**.
5. Navigate to **Infrastructure** → **Subnets** and select the required Smart Proxy for each subnet that you want to use discovery, and verify that it is connected to the Discovery Proxy.
6. Navigate to **Provisioning Templates**, edit the PXELinux global default template and modify it according to the example below.



NOTE

Different options appear on the APPEND line compared to the Satellite 6.0 release.

```

LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2

```

The **proxy.type** option can be either **proxy** or **foreman**. If you specify **proxy** then all communication goes through the Satellite Capsule. This is the preferred method. If you specify **foreman** then all communication goes directly to the Satellite Server. This is the method used by Satellite 6.0.



NOTE

When using proxy type, the default port on Satellite Capsule is 9090, but for direct communication with Satellite Server, you need to use port 80.

The **proxy.url** option specifies the URL of the Satellite Capsule or Server depending on the previous setting. Both HTTP and HTTPS schemes are supported.

It is possible to omit the **proxy.url** option to determine the Capsule DNS name from its SRV record. This might be useful when there are multiple discovery subnets. Review the global settings and permissions in the Satellite Server user interface. See the [Red Hat Satellite 6.1 User Guide](#) for more information.

8.4. UPGRADING RED HAT SATELLITE CLIENTS

The `katello-agent` package from Satellite 6.0 is not compatible with Red Hat Satellite 6.1. You need to manually upgrade to the new version of `katello-agent` on all Satellite clients.

Procedure 8.7. To Upgrade the `katello-agent` Package:

1. Log in to the client system and enable the Satellite tools repository.

```
# subscription-manager repos --enable=rhel-version-server-satellite-
tools-6.1-rpms
```

Replace *version* with **6** or **7** depending on the Red Hat Enterprise Linux version you are using.

2. Synchronize the repository. Replace *ID* with the ID of the tools repository.

```
# hammer repository synchronize --id ID
```

3. Upgrade the `katello-agent` package.

```
# yum upgrade katello-agent
```

**IMPORTANT**

It is currently not possible to upgrade the agent using Red Hat Satellite before upgrading Satellite itself.

8.5. UPGRADING BETWEEN MINOR VERSIONS OF SATELLITE

This procedure must be followed to upgrade between minor versions, for example, from 6.1.8 to 6.1.9.

Prerequisites

- Ensure you have synchronized Satellite Server repositories for Satellite, Capsule, and Satellite Tools.
- Ensure each external Capsule and Content Host can be upgraded by promoting the updated repositories to all relevant content views.

Procedure 8.8. Upgrading the Satellite Server to the Next Minor Version

1. Check that only the correct repositories are enabled:

- a. List the enabled repositories:

```
subscription-manager repos --list-enabled
```

- b. Ensure you only have the following repositories enabled:

```
rhel-X-server-rpms
rhel-X-server-satellite-6.1-rpms
rhel-server-rhsc1-X-rpms
```

Where *X* is the major version of Red Hat Enterprise Linux you are using. If you have a self-registered Satellite, the Red Hat Satellite Tools repository (rhel-6-server-satellite-tools-6.1-rpms or rhel-7-server-satellite-tools-6.1-rpms), which provides Katello Agent, can also be present.

2. If you are on a self-registered Satellite, download all packages **before** stopping Satellite Server:

```
# yum update --downloadonly
```

This step is optional for Satellites which are **not** self-registered.

3. Stop services:

```
# katello-service stop
output omitted
Success!
```

4. Update all packages:

```
# yum update
```

5. If a kernel update occurs, reboot the system:

```
# reboot
```

6. Perform the upgrade:

```
# katello-installer --upgrade
```

7. On a self-registered Satellite, restart **goferd**:

- On Red Hat Enterprise Linux 6:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart goferd
```

Procedure 8.9. Upgrading a Capsule Server to the Next Minor Version

1. Check that only the correct repositories are enabled:

- a. List the enabled repositories:

```
subscription-manager repos --list-enabled
```

- b. Ensure you only have the following repositories enabled:

```
rhel-X-server-rpms  
rhel-X-server-satellite-capsule-6.1-rpms  
rhel-server-rhsc1-X-rpms  
rhel-X-server-satellite-tools-6.1-rpms
```

Where *X* is the major version of Red Hat Enterprise Linux you are using. The Red Hat Satellite Tools repository (rhel-6-server-satellite-tools-6.1-rpms or rhel-7-server-satellite-tools-6.1-rpms), provides Katello Agent.

2. Stop services:

```
# katello-service stop  
output omitted  
Success!
```

3. Update all packages:

```
# yum update
```

4. If a kernel update occurs, reboot the system:

```
# reboot
```

5. Perform the upgrade:

```
# capsule-installer --upgrade
```

6. Restart **goferd**:

- On Red Hat Enterprise Linux 6:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart goferd
```

Procedure 8.10. Upgrading a Content Host to the Next Minor Version

1. Update all packages:

```
# yum update
```

2. If a kernel update occurs, reboot the system:

```
# reboot
```

3. Restart **goferd**:

- On Red Hat Enterprise Linux 6:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7:

```
# systemctl restart goferd
```

[10] https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.1/html/User_Guide/chap-Red_Hat_Satellite-User_Guide-Configuring_Activation_Keys.html

[11] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-yum.html#s1-yum-upgrade-system

[12] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7-Beta/html/System_Administrators_Guide/ch-yum.html#s1-yum-upgrade-system

CHAPTER 9. NEXT STEPS

The content of the Installation Guide takes you through installing Red Hat Satellite Server, Capsule Server, and setting up the repositories so that client host systems can update from the Satellite Server. There are other configuration steps you will need to take to take full advantage of your Red Hat Satellite Server and Capsule Server. The [Red Hat Satellite 6.1 User Guide](#) can assist in configuring life cycle environments, products, organizations, locations, and other components while the [Red Hat Satellite Provisioning Guide](#) can assist with setting up a working provisioning environment for your Red Hat Satellite Server.

CHAPTER 10. UNINSTALLING RED HAT SATELLITE SERVER AND CAPSULE SERVER



WARNING

The following procedures will erase all applications that are used with Red Hat Satellite Server or Red Hat Satellite Capsule Server on the target system. If you are using any of these applications or application data for any other purposes than Red Hat Satellite, back up the information before running these scripts.

REMOVING SATELLITE SERVER

The command to uninstall Red Hat Satellite Server is **katello-remove**. The uninstall script will issue a warning twice, requiring confirmation before it removes all packages and configuration files in the system. Below is a sample output of the command:

```
# katello-remove
WARNING: This script will erase many packages and config files.
Important packages such as the following will be removed:
* elasticsearch
* httpd (apache)
* mongodb
* tomcat6
* puppet
* ruby
* rubygems
* All Katello and Foreman Packages
Once these packages and configuration files are removed there is no going
back.
If you use this system for anything other than Katello and Foreman you
probably
do not want to execute this script.
Read the source for a list of what is removed.  Are you sure(Y/N)? y
ARE YOU SURE?: This script permanently deletes data and configuration.
Read the source for a list of what is removed.  Type [remove] to continue?
remove
Shutting down Katello services...
...
```

REMOVING CAPSULE SERVER

The command to uninstall Red Hat Satellite Capsule Server is **capsule-remove** from the capsule-installer package. Same as **katello-remove**, **capsule-remove** will issue a warning twice, requiring confirmation before removing the content.

APPENDIX A. GLOSSARY OF TERMS

The following terms are used throughout this document. Familiarize yourself with these terms to help your understanding of Red Hat Satellite 6.

Activation Key

A registration token used in a Kickstart file to control actions at registration. These are similar to Activation Keys in Red Hat Satellite 5, but provide a subset of features because Puppet controls package and configuration management after registration.

Application Life Cycle Environment

An *Application Life Cycle Environment* represents a step, or stage, in a promotion path through the Software Development Life Cycle (SDLC). Promotion paths are also known as development paths. Content such as packages and Puppet modules move through life cycle environments by publishing and promoting Content Views. All Content Views have versions, which means you can promote a specific version through a typical promotion path; for example, from development to test to production. Channel cloning implements this concept in Red Hat Satellite 5.

Attach

The process of associating a Subscription to a Host that provides access to RPM content.

Capsule

A *Capsule* is an additional server that can be used in a Red Hat Satellite 6 deployment to facilitate content federation and distribution in addition to other localized services (Puppet Master, **DHCP**, **DNS**, **TFTP**, and more).

Catalog

A *Catalog* is a document that describes the desired system state for one specific computer. It lists all of the resources that need to be managed, as well as any dependencies between those resources.

Compute Profile

Compute Profiles specify default attributes for new virtual machines on a compute resource.

Compute Resource

A *Compute Resource* is virtual or cloud infrastructure, which Red Hat Satellite 6 uses for deployment of hosts and systems. Examples include Red Hat Enterprise Virtualization Manager, OpenStack, EC2, and VMWare.

Content

Content includes software packages (RPM files) and Puppet modules. These are synchronized into the Library and then promoted into Life Cycle Environments using Content Views so that they can be consumed by Hosts.

Content Delivery Network (CDN)

The *Content Delivery Network (CDN)* is the mechanism used to deliver Red Hat content in a geographically co-located fashion. For example, content that is synchronized by a Satellite in Europe pulls content from a source in Europe.

Content Host

A *Content Host* is the part of a host that manages tasks related to content and subscriptions.

Content View

A *Content View* is a definition of content that combines products, packages, and Puppet modules with capabilities for intelligent filtering and creating snapshots. Content Views are a refinement of the combination of channels and cloning from Red Hat Satellite 5.

External Node Classifier

An *External Node Classifier* is a Puppet construct that provides additional data for a Puppet Master to use when configuring Hosts. Red Hat Satellite 6 acts as an External Node Classifier to Puppet Masters in a Satellite deployment.

Facter

Facter is a program that provides information (facts) about the system on which it is run; for example, Facter can report total memory, operating system version, architecture, and more. Puppet modules enable specific configurations based on host data gathered by Facter.

Hammer

Hammer is a command line tool for Red Hat Satellite 6. Use Hammer to manage Red Hat Satellite 6 as a standard CLI, for scripts, and also through an interactive shell.

Hiera

Hiera is a key/value look-up tool for configuration data which allows keeping site-specific data out of puppet manifests.

Host

A *Host* refers to any system, either physical or virtual, that Red Hat Satellite 6 manages.

Host Collection

A *Host Collection* is equivalent to a Satellite 5 *System Group*, that is, a user defined group of one or more Hosts.

Host Group

A *Host Group* is a template for building a Host. This includes the content view (which defines the available RPM files and Puppet modules) and the Puppet classes to apply (which ultimately determines the software and configuration).

Location

A *Location* is collection of default settings that represent a physical place. These can be nested so that you can set up an hierarchical collection of locations. For example, you can set up defaults for "Middle East", which are refined by "Tel Aviv", which are further refined by "Data Center East", and then finally by "Rack 22".

Library

The *Library* contains every version, including the latest synchronized version, of the software that the user will ever deploy. For an Information Technology Infrastructure

Library (ITIL) ^[13] organization or department, this is the Definitive Media Library^[14] (previously named the Definitive Software Library).

Manifest

A *Manifest* transfers subscriptions from the Customer Portal to Red Hat Satellite 6. This is similar in function to certificates used with Red Hat Satellite 5.

For more information about certificates and subscription types, see:

- [RHN Classic, Red Hat Satellite, and Channel Entitlements](#)^[15]
- [The Structure of Satellite Certificates \(Classic Style of Certificates\)](#)^[16]

Organization

An *Organization* is an isolated collection of systems, content, and other functionality within a Satellite 6 deployment.

Product

A collection of content repositories. Products can be Red Hat products or newly-created products made up of software and configuration content.

Promote

The act of moving a content view comprised of software and configuration content from one Application Life Cycle Environment to another, such as moving from development to QA to production.

Provisioning Template

A *Provisioning Template* is a user-defined template for Kickstart files, snippets, and other provisioning actions. In Satellite 6 they provide similar functionality to Kickstart Profiles and cobbler Snippets in Red Hat Satellite 5.

Pulp Node

A *Pulp Node* is a Capsule Server component that mirrors content. This is similar to the Red Hat Satellite 5 Proxy. The main difference is that content can be staged on the Pulp Node before it is used by a Host.

Puppet Agent

The *Puppet Agent* is an agent that runs on a Host and applies configuration changes to that Host.

Puppet Master

A *Puppet Master* is a Capsule Server component that provides Puppet manifests to Hosts for execution by the Puppet Agent.

Puppet Module

A *Puppet Module* is a self-contained bundle of code and data that you can use to manage resources such as users, files, and services.

Repository

A *Repository* provides storage for a collection of content. For example, a YUM repository

or a Puppet repository.

Role

A *Role* specifies a collection of permissions that are applied to a set of resources, such as Hosts.

Smart Proxy

A *Smart Proxy* is a Capsule Server component that can integrate with external services, such as **DNS** or **DHCP**.

Smart Variable

A *Smart Variable* is a configuration value that controls how a Puppet Class behaves. This can be set on a Host, a Host Group, an Organization, or a Location.

Standard Operating Environment (SOE)

A *Standard Operating Environment (SOE)* is a controlled version of the operating system on which applications are deployed.

Subscription

Subscriptions are the means by which you receive content and service from Red Hat.

Synchronizing

Synchronizing refers to mirroring content from external resources into the Red Hat Satellite 6 Library.

Synchronization Plans

Synchronization Plans provide scheduled execution of content synchronization.

User Group

A *User Group* is a collection of roles which can be assigned to a collection of users. This is similar to a Role in Red Hat Satellite 5.

User

A user is anyone registered to use Red Hat Satellite. Authentication and authorization is possible through built-in logic, through external LDAP resources, or with Kerberos.

[13] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[14] http://en.wikipedia.org/wiki/Definitive_Media_Library

[15] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/MigratingRHN/sat-certs.html

[16] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/Subscription_Concepts_and_Workflows/index.html#subscription-legacy

APPENDIX B. REVISION HISTORY

Revision 1-70	Wed 30 Nov 2016	Stephen Wadeley
Bug 1329930 - [RFE] Add Satellite Communication Matrix details to Satellite-6 product documentation.		
Revision 1-69	Fri 21 Oct 2016	Stephen Wadeley
Bug 1367674 - Review upgrade procedures.		
Revision 1-68	Tue 13 Sept 2016	Brandi Munilla
Bug 1364249 - Incorrect commands in satellite 6.1 installation document.		
Revision 1-67	Tue 18 Aug 2016	Stephen Wadeley
Bug 1362527 - Provide instructions on how to perform upgrades between minor releases.		
Revision 1-66	Tue 02 Aug 2016	Stephen Wadeley
Bug 1346928 - Missing port in Satellite to Capsule communication in chapter 7.2.3.		
Revision 1-65	Wed Apr 27 2016	Russell Dickenson
BZ#1322207 - Amended subscription manager example command.		
Revision 1-64	Tue Dec 15 2015	Russell Dickenson
BZ#1249288 - Updated the section on setting a custom server certificate.		
Revision 1-63	Mon Nov 16 2015	Hayley Hudgeons
Building for async 2.		
Revision 1-62	Mon Oct 12 2015	Hayley Hudgeons
Building for async 1.		
Revision 1-61	Thu Sept 24 2015	Megan Lewis
BZ#1146946 - Integrated peer review feedback.		
Revision 1-60	Mon Sept 21 2015	Megan Lewis
BZ#1146946 - Added chapter on Configuring a Self-Registered Satellite.		
Revision 1-59	Tues August 25 2015	Hayley Hudgeons
Re-building GA docs to include html-single format.		
Revision 1-58	Fri August 7 2015	Ella Deon Ballard
Building for GA.		
Revision 1-57	Mon August 3 2015	Jo Somers
BZ#1243608 Added supported upgrade paths to chapter 7 Upgrading Red Hat Satellite Server and Capsule Server		
Revision 1-56	Fri July 24 2015	Megan Lewis
BZ#1209249 Reverted changes to 2.2.3.2. Configuring Red Hat Satellite with a Custom Server Certificate.		
Revision 1-55	Thu July 23 2015	Jo Somers
BZ 1206788: Section 4.2 Disconnected Satellite, updated to match Satellite 6.1 User Guide, section Disconnected Satellite.		
Revision 1-54	Thu July 23 2015	David O'Brien
BZ 1205469: Review section on obtaining packages for connected and disconnected environments. Minor updates.		
Revision 1-53	Tue July 21 2015	Jo Somers

BZ#1234016 Changed subscription-manager --env option in section Obtaining the Required Packages for the Capsule Server, Prerequisites, Step 2.

Revision 1-52	Mon July 20 2015	Megan Lewis
BZ#1243608 Added important note to 1.4. Prerequisites, 6.2. Red Hat Satellite Capsule Server Prerequisites, and Chapter 7. Upgrading Red Hat Satellite Server and Capsule Server stating that Satellite server and Capsule versions must match.		
BZ#1209249 Updated 2.2.3.2. Configuring Red Hat Satellite with a Custom Server Certificate with --foreman-server flags.		
Revision 1-51	Fri July 17 2015	Jo Somers
Fix BZ1242526 section 7.1 Upgrading RH Satellite Capsule Servers, Added warning		
Fix BZ1241273 section 7.1 Upgrading RH Satellite Capsule Servers, Added new step 9 Restart all services'		
Fix BZ1235777 Section 6.4 Running the Installation and Configuration Program for Capsule Server, Prerequisites: Added two commands after note and rewrote Section 6.4.1 Adding a Capsule Server		
Revision 1-50	Thu July 16 2015	Jo Somers
Fix BZ1241461 section 7.1 Upgrading Red Hat Satellite Server and Capsule server,step 6 changed		
Fix BZ1230332 section 2.2.1 Configuring Red Hat Satellite Manually,step 2a and step 2b changed uid-owner katello to uid-owner foreman		
Revision 1-49	Tue July 14 2015	David O'Brien
Remove draft status.		
Rebuild for technical review.		
Revision 1-48	Mon July 13 2015	Megan Lewis
BZ#1200617 Corrected the directory used throughout the procedure.		
Revision 1-47	Sat July 11 2015	David O'Brien
BZ #1241581 Clarify requirement to manually upgrade katello-agent on all clients.		
Revision 1-46	Wed July 8 2015	Jo Somers
BZ#1200617 Corrected Step 4 and combined steps 7 and 8 in section 4.2 Disconnected Satellite.		
Revision 1-45	Thu July 2 2015	Jo Somers
BZ#1171611 Corrected error in Step 2 of section 6.7 Registering Host Systems to a Red Hat Satellite Capsule Server.		
Revision 1-44	Wed July 1 2015	Megan Lewis
BZ#1206788 Corrected error in Step 2 of 4.2.4. Importing Content to a Disconnected Satellite Server.		
Revision 1-43	Thu Jun 25 2015	Jo Somers
BZ 1234705 Changed channel to repository in sections Red Hat Satellite 6 Supported Usage, Prerequisites, Synchronization Status, Base Operating Systems, Downloading from a Disconnected Network		
Revision 1-42	Wed Jun 24 2015	Jo Somers
BZ 1200617 Deleted sections:Configuring the Synchronization Host, Synchronizing Content, Exporting Content		
Revision 1-41	Mon Jun 15 2015	David O'Brien
6.1 Public Beta release.		
Add edition number.		
Revision 1-40	Thu June 8 2015	Jo Somers
BZ 1180277 Changed firewall-cmd --reload in section Required Network Ports.		
BZ 1180277 In section Application Specifications, added Red Hat Enterprise Linux 7 chronyd command.		
Revision 1-39	Thu June 4 2015	Jo Somers

BZ 1180277 Added firewall-cmd --reload to section Configuring Red Hat Satellite Manually.

Revision 1-38 **Wed May 27 2015** **David O'Brien**
BZ 1216072 Cleaned up instances of "beta" in GA release.

Revision 1-37 **Wed May 13 2015** **David O'Brien**
Add chapter on virt-who for tech review.

Revision 1-36 **Mon May 11 2015** **David O'Brien**
Tech review version.

Revision 1-35 **Mon May 4 2015** **Athene Chan**
BZ#1195556 Updated the "Registering Host Systems to the Red Hat Capsule Server" section.
Updated the instructions to installing a Capsule Server and the types of Capsule Servers. Rearranged configuration options.
BZ#1209761 Removed the option "-v" in the "katello-installer" command from the "Configuring DNS, DHCP and TFTP" section.
BZ#1212974 Added UDP firewall rules to port 53 in the "Required Network Ports" section.
BZ#1129498 Removed the "#" at the beginning of the commands in the "Required Network Ports" section of the Capsule Server section.
BZ#1167898 Removed unsupported DNS support from the "Red Hat Satellite Capsule Server" section.
BZ#1188300 Added port 8443 as a required free port for subscription management services in the Prerequisites section.

Revision 1-34 **Thu April 30 2015** **Megan Lewis**
BZ#1175924 Updated note in 4.2.4. Importing Content to a Disconnected Satellite Server.

Revision 1-33 **Wed April 29 2015** **Megan Lewis**
BZ#1175835 Updated example in 4.2.2 Synchronizing Content.
BZ#1175924 Updated Step 7 of 4.2.4. Importing Content to a Disconnected Satellite Server.
Fixed typos in 4.2.4. Importing Content to a Disconnected Satellite Server.

Revision 1-32 **Tue April 28 2015** **Athene Chan**
BZ#1202055 Changed the instructions as per comment #8 in the bug to reflect the correct procedure for upgrading Capsule Servers.

Revision 1-31 **Mon April 27 2015** **Jo Somers**
BZ#1171697 In section Prerequisites, added hostname requirements

Revision 1-30 **Fri April 24 2015** **Jo Somers**
BZ#1171611 In section Registering Host Systems to a Red Hat Satellite Capsule, changed subscription-manager command from org name to org label

Revision 1-29 **Thu April 23 2015** **Megan Lewis**
BZ#1192272 Corrected error in Configuring Red Hat Satellite with an Answer File.

Revision 1-28 **Wed April 22 2015** **Athene Chan**
Updated all prerequisites from Red Hat Enterprise Linux 6.5 to 6.6.

Revision 1-29 **Wed April 22 2015** **Jo Somers**
In section Prerequisites, updated ports in table 1.26.5 to 6.6.

Revision 1-27 **Wed April 15 2015** **Athene Chan**

BZ#1180715 Added a table to the "Storage" prerequisites.

BZ#1174453 Changed "katello-installer" to "capsule-installer" in the "Obtaining the Required Packages for the Capsule Server" section.

BZ#1205493 Updated procedure for creating a manifest.

Revision 1-26 **Wed April 8 2015** **Megan Lewis**
Updated brand.

Revision 1-25 **Fri April 1 2015** **Athene Chan**
Restructured the installation guide's table of contents.

Revision 1-24 **Fri April 1 2015** **Athene Chan**
BZ#1166191 Added a note about chained certificates.
Changed the procedure to "Setting Up a Manifest" in accordance to the changes in the Customer Portal.
BZ#1145823 Changed a step to make sure that organization names are used for the "Satellite Name" when registering a Satellite for manifests.
BZ#1194392 Clarified that the Satellite subscription should not be attached to the manifest.
BZ#1185849 Changed the output if the subscription SKU and changed the second step in the procedure "To Install a Satellite Capsule Server on a Certificate-managed System"
BZ#1185836 Added "Capsule" to the note in the "Red Hat Satellite Capsule Server Prerequisites" section.
BZ#1174578 Removed duplicated capsule registration steps in "Installing a Red Hat Satellite Capsule Server" and "Configuring a Red Hat Satellite Capsule Server".
BZ#1173816 Removed the firewall rules on elasticsearch in the "Configuring a Red Hat Satellite Capsule Server" section as the Capsule server does not use elasticsearch.
Changed the repository names to correct Beta repositories for both the Satellite Server and Capsule.
BZ#1173680 Added a note on the Storage prerequisites section about latency and networked storage.
BZ#1176479 Added information on configuring DNS, DHCP, and TFTP to the Configuration Options.
Added firewall port 5674 for amqp connections and SELinux considerations for amqp in the prerequisites section.

Revision 1-23 **Mon Mar 30 2015** **David O'Brien**
BZ 1203878: Update RH Common repository name to Satellite Tools.

Revision 1-22 **Wed Mar 23 2015** **Jo Somers**
BZ#1201194 In section Prerequisites, added Red Hat Enterprise Linux 6.6 or later

Revision 1-21 **Wed Mar 23 2015** **Jo Somers**
BZ#1201193 Added Red Hat Enterprise Linux 6.6 or later and reference to solution article in section Installing Red Hat Satellite with an ISO Image-Prerequisites

Revision 1-20 **Wed Mar 18 2015** **Jo Somers**
BZ#1200617 Added new steps 1-6 in section Importing Content to a Disconnected Satellite Server.

Revision 1-19 **Tue Mar 17 2015** **Athene Chan**
BZ#1170334 Added network ports to be opened as a prerequisite to installation.
BZ#1193153 sentence structure change to procedure statement.

Revision 1-18 **Thu Mar 12 2015** **Jo Somers**
BZ#1119934 In section Configuring Red Hat Satellite Manually, Procedure 2.2 Running the Installer Script: changed Step 1 katello-installer command

Revision 1-17 **Mon Mar 09 2015** **David O'Brien**
BZ#1166642 Add comment to enable SELinux and relabel files after installation if SELinux was disabled during installation.

Revision 1-16 **Wed Mar 03 2015** **Jo Somers**

Fix BZ 1170713 In section Installing Red Hat Satellite, Procedure 2.1, for Red Hat Enterprise Linux 7, added repo names before yum install

Revision 1-15**Fri Feb 27 2015****David O'Brien**

BZ#1183657 Add "puppet module" and "catalog" to Glossary

Revision 1-14**Wed Feb 25 2015****Athene Chan**

BZ#1180191 Corrected the required RPMs to install for synchronizing hosts in a disconnected Satellite Server.

Revision 1-13**Tue Feb 18 2015****Jo Somers**

BZ#1180277 Corrected firewall command from complete reload to reload in section Red Hat Satellite Capsule Server Prerequisites.

BZ#1180277 Added firewall reload command in section Configuring a Red Hat Satellite Capsule Server.

Revision 1-12**Mon Feb 9 2015****Megan Lewis**

BZ#1178176 Further corrections in 4.2.4. Importing Content to a Disconnected Satellite Server.

BZ#1177574 Added line breaks to Procedure 2.5 in 2.3.2. Configuring Red Hat Satellite with a Custom Server Certificate.

Revision 1-11**Fri Jan 23 2015****Athene Chan**

BZ#1184589 Emphasize what base operating system variants is required for Red Hat Satellite.

Revision 1-10**Fri Jan 23 2015****Megan Lewis**

BZ#1178176 Corrected 40G to 40GB in 4.2.4. Importing Content to a Disconnected Satellite Server.

BZ#1179022 Corrected errors in examples in 5.4. Configuring a Red Hat Satellite Capsule Server.

Revision 1-9**Fri Jan 23 2015****Athene Chan**

BZ#1177568 Replaced the "service" and "chkconfig" command for chronyd to the recommended "systemctl" command instead.

Revision 1-8**Wed Jan 21 2015****David O'Brien**

BZ 1184306 - Make the requirement for a Base install more obvious.

Revision 1-7**Thu Dec 18 2014****Megan Lewis**

BZ#1168273 Corrected package name for installing puppet agent.

BZ#1169499 Clarified supported Red Hat Enterprise Linux variants in Prerequisites.

BZ#1164251 Corrected example in Adding Lifecycle Environments to a Red Hat Satellite Capsule Server.

BZ#1167904 Added chrony and sos into the prerequisites for install.

Revision 1-6.2**Thu Nov 19 2014****Athene Chan**

Added additional admin and password options to the katello-installer.

Removed hashes on the firewall requirements.

Included references to support for scripting frameworks in the Puppet Supported Usage paragraph.

Revision 1-6.1**Friday Nov 14 2014****Athene Chan**

BZ#1153567 Added a "Capsule Scalability" section.

Revision 1-6**Thu Nov 13 2014****Athene Chan**

BZ#1153564 Added a "Next Steps" chapter.

BZ#1153772 Added firewall configuration and additional steps to ensure that the Satellite Server can go through the HTTP Proxy without issues.

BZ#1146574 Changed the gpg filename.

Revision 1-5**Tue Nov 11 2014****Athene Chan**

BZ#1132840 Added two advanced firewall consideration tables in the prerequisites.

BZ#1152630 Edited incorrect reference to Red Hat Enterprise Linux 7.

BZ#1150412 Added "--complete-reload" to the firewall-cmd firewall commands.

BZ#1143746 Changed incorrect certs-server-key in procedure 2.4.

Revision 1-4 **Mon Nov 10 2014** **Athene Chan**

BZ#1152630 Added RHEL7 firewall-cmd command examples for the firewall requirements.

Revision 1-3 **Fri Nov 7 2014** **Athene Chan**

BZ#1161254 Added a new firewall rule to the list of firewall rules to allow katello-installer to run after initial install. Moved the firewall rules to the "Configuring Red Hat Satellite" sections to prevent errors.

Revision 1-2.02 **Fri Oct 3 2014** **Athene Chan**

Various edits from translators' feedback.

BZ#1147673 Removed MS DHCP from supported DHCP features.

BZ#1140520 Changed all "ACME_Corporation" entries to the correct default organization entry "Default Organization".

BZ#1139806 Added a note in the Prerequisites sections for Red Hat Satellite Server and Red Hat Satellite Capsule Server that the host system has to be updated before installing Red Hat Satellite. BZ#1138430 Changed "yum-config-manager" to "subscription-manager" to match the procedure description to the command block.

BZ#1141954 Added example repositories to the "Enabling Red Hat Repositories" section and a note to enable RH Common repositories for client systems.

BZ#1140722 Added note to highlight that the command needs to change if the repository is different from the example command.

Revision 1-2.01 **Fri Sep 12 2014** **Athene Chan**

BZ#1140875 Added firewall rules after the Satellite Server and Capsule Server installation.

Revision 1-2 **Thu Sep 11 2014** **Athene Chan**

BZ#1140422 Changed the repository names for Red Hat Satellite Server and Red Hat Satellite Capsule Server.

Revision 1-1 **Wed Sep 10 2014** **Athene Chan**

Added additional ports in the Prerequisites section.

Revision 1-0 **Tue Sep 9 2014** **Athene Chan**

Red Hat Satellite 6.0 GA Release

Revision 0-34 **Thu Aug 21 2014** **Athene Chan**

BZ#1131360 Replaced an option on the command to reflect the correct one.

Revision 0-33 **Tue Aug 12 2014** **Athene Chan**

BZ#1130208 Added "Red Hat Software Collections" as a channel to enable.

BZ#1129104 Add requirement to make port 8080 available for katello installation. Update how to configure iptables accordingly.

BZ#1125241 Added a note that default location and default organization can be changed after initial configuration.

BZ#1044558 Added chapter on http proxy configuration options in katello-installer.

BZ#1120492 Added a note in "Red Hat Satellite Server Supported Usage" about embedded tomcat deployments.

BZ#1125299 Added references to "next steps" sections in the "Installing Red Hat Satellite" chapter.

BZ#1125357 Removed the deprecated repository directories.

BZ#1121814 Corrected the Satellite Capsule Server installer option.

BZ#1089086 Included filesize recommendations in the Prerequisites.

BZ#1119866 Added the Red Hat Software Collections package as a required package for the Satellite Capsule Server installation.

BZ#1118406 Added a table of ports, protocols and services in the Prerequisites section.

BZ#1120855 Various corrections on filenames and commands.

BZ#1121676 Added a note that all hammer commands are ran on the Satellite Server.

BZ#1113811 Created the section "Red Hat Satellite 6 Supported Usage".

BZ#1128922 Added a "Results" subsection.

BZ#754728 Added sections "Configuring Red Hat Satellite with a Custom Server Certificate" and "Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate"

BZ#1122183 Changed the entry on Account Username and added an example for Base DN.

BZ#1129498 Group iptables commands for better readability.

Revision 0-32

Fri Jul 11 2014

Athene Chan

BZ#1157545, BZ#115047, BZ#1116471, BZ#1117052, BZ#1117052, BZ#1115065 Minor edits, spelling errors and revisions to text.

Revision 0-31

Mon Jun 30 2014

Athene Chan

Book published for Beta Release.

Revision 0-30

Tue Jun 24 2014

Dan Macpherson

Second test brewing for Beta.

Revision 0-29

Tue Jun 24 2014

Dan Macpherson

Test brewing for Beta.

Revision 0-28

Mon Nov 11 2013

Dan Macpherson

Fixing minor error.

Revision 0-27

Mon 11 Nov 2013

Dan Macpherson

Preparation for MDP2.

Revision 0-26

Mon 11 Nov 2013

Athene Chan

BZ#1024530, 1027466 Additional edits to steps for Satellite nodes.

Revision 0-25

Thu 7 Nov 2013

Megan Lewis

BZ#1027461 Added steps to create activation key and retrieve oauth secret. Added note to verify nodes exist.

Revision 0-24

Thu 7 Nov 2013

Athene Chan

BZ#1027466 Added a small seciton on using Satellite nodes. Added synchronization step.

Revision 0-23

Wed 30 Oct 2013

Athene Chan

BZ#1024438 changed procedures to accommodate yum-utils installation.
 BZ#1024529 removed katello.yml instructions as this is not preferred way of LDAP configuration.
 BZ#1024559 Added foreman-libvirt to the yum install command.
 BZ#1024530 Added new information to the section on Satellite Nodes.

Revision 0-22	Tue 29 Oct 2013	Athene Chan
BZ#1024094 yum-utils command updated.		
Revision 0-21	Wed 09 Oct 2013	Dan Macpherson
Finalizing QE review implementation		
Revision 0-20	Wed 2 Oct 2013	Athene Chan
BZ#1014402 Installation requirements updated.		
Revision 0-19	Wed 2 Oct 2013	Athene Chan
BZ#1014402 Prerequisites for installation updated.		
Revision 0-18	Tue 1 Oct 2013	Athene Chan
BZ#1009719, 971944 Minor spelling and grammar edits.		
Revision 0-17	Thu 19 Sep 2013	Athene Chan
BZ#1009719 Updated the Prerequisites and the install instructions.		
Revision 0-16	Tue 17 Sep 2013	Athene Chan
BZ#971944 Added storage requirements for Satellite.		
Revision 0-15	Wed 11 Sep 2013	Megan Lewis
Integrating QE feedback.		
Revision 0-14	Mon 12 Aug 2013	Dan Macpherson
Removing draft watermark.		
Revision 0-13	Mon 12 Aug 2013	Dan Macpherson
Preparing documentation for technical review.		
Revision 0-09	Thu 20 June 2013	Dan Macpherson
Correction to repo label for installation.		
Revision 0-08	Thu 20 June 2013	Dan Macpherson
Added MDP1 status.		
Revision 0-07	Wed 19 June 2013	Athene Chan
Revised channel for installation.		
Revision 0-06	Thu 13 June 2013	Athene Chan
Edited book for grammatical errors and sentence structure.		
Revision 0-05	Tue 11 June 2013	Athene Chan
Added Chapters for manifests and for synchronization. Edited sections based on technical review feedback.		
Revision 0-04	Fri 31 May 2013	Athene Chan
Changed field names in the Satellite:Provisioning LDAP section.		
Revision 0-03	Thu 30 May 2013	Athene Chan
Renamed all web application components to the rebranded names of "Red Hat Satellite: Content and Entitlement" and "Red Hat Satellite: Provisioning and Configuration".		

Revision 0-02	Tue 28 May 2013	Athene Chan
Incorporated technical review edits.		
Updated commands for installing Red Hat Satellite.		
Standardized tagging of components.		

Revision 0-01	Fri 17 May 2013	Athene Chan
Initial book creation		