Red Hat Satellite 6.2

Installation Guide

Installing Red Hat Satellite Server and Capsule Server

Last Updated: 2018-11-30

Red Hat Satellite 6.2 Installation Guide

Installing Red Hat Satellite Server and Capsule Server

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

Legal Notice

Abstract

This guide describes how to install Red Hat Satellite Server and Capsule Server, perform initial configuration, and configure external services.

Table of Contents

CHAPTER 1. WHAT SATELLITE SERVER AND CAPSULE SERVER DO

Red Hat Satellite is a system management solution that enables you to deploy, configure, and maintain your systems across physical, virtual, and cloud environments. Satellite provides provisioning, remote management and monitoring of multiple Red Hat Enterprise Linux deployments with a single, centralized tool. Red Hat Satellite Server synchronizes the content from Red Hat Customer Portal, and provides functionality including fine-grained life cycle management, user and group role-based access control, integrated subscription management, as well as advanced GUI, CLI, and API access.

Red Hat Satellite Capsule Server mirrors content from Red Hat Satellite Server to facilitate content federation across various geographical locations. Host systems can pull content from the Capsule Server and not from the central Satellite Server. The Capsule Server also provides localized services such as Puppet Master, DHCP, DNS, or TFTP. Capsule Servers assist you in scaling your Satellite environment as the number of your managed systems increases.

Capsule Servers decrease the load on the central server, increase redundancy, and reduce bandwidth usage. For more information see the *Red Hat Satellite Architecture Guide*.

CHAPTER 2. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

2.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base system:

- 64-bit architecture
- The latest version of Red Hat Enterprise Linux 6 Server or 7 Server
- A minimum of 2 CPU cores, 4 CPU cores are recommended
- A minimum of 12 GB memory is required for the Satellite Server to function, 16 GB of memory or more is recommended for each instance of Satellite Server. In addition, a minimum of 4 GB of swap space is also recommended. Satellite running with less memory than the minimum value may not operate correctly.
- A unique host name, which can contain lower-case letters, numbers, dots (.) and hyphens (-)
- A current Red Hat Satellite subscription
- Administrative user (root) access
- A system umask of 0022
- Full forward and reverse DNS resolution using a fully-qualified domain name

Before you install Satellite Server or Capsule Server, you should ensure that your environment meets the requirements for installation.

NOTE

The Red Hat Satellite Server and Capsule Server versions must match. For example, a Satellite 6.1 Server cannot run a 6.2 Capsule Server and a Satellite 6.2 Server cannot run a 6.1 Capsule Server. Mismatching Satellite Server and Capsule Server versions results in the Capsule Server failing silently.

If you have a large number of content hosts, see Appendix A, *Large Deployment Considerations* to ensure that your environment is set up appropriately.

For more information on scaling your Capsule Servers, see Appendix B, *Capsule Server Scalability Considerations*.

2.2. STORAGE REQUIREMENTS AND RECOMMENDATIONS

Ensure that your environment meets the minimum requirements before installing Satellite Server or Capsule Server.

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 in 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.

The **/var/cache/pulp/** directory is used to temporarily store content while it is being synchronized. For content in RPM format, a maximum of 5 RPM files are stored in this directory at any time. After each file is synchronized, it is moved to the **/var/lib/pulp/** directory. Up to eight RPM content synchronization tasks can be running simultaneously by default, with each using up to 1 GB of metadata. For content in ISO format, all ISO files per synchronization task are stored in **/var/cache/pulp/** until the task is complete, after which they are moved to the **/var/lib/pulp/** directory. For example, if you are synchronizing four ISO files, each 4 GB in size, this requires a total of 16 GB in the **/var/cache/pulp/** directory. Take into account the number of ISO files you intend synchronizing because the temporary disk space required for them typically exceeds that of RPM content.

The **/var/lib/qpidd/** directory uses slightly more than 2 MB per Content Host. For example, 10 000 Content Hosts would require 20 GB of disk space in **/var/lib/qpidd/**.

Storage Requirements

The following tables detail recommended storage requirements for specific directories. These values are based on expected use case scenarios and can vary according to individual environments. The Capsule Server table also applies to the Satellite Server as it has an integrated Capsule by default. Pay attention to your specific use case when reading the tables. For example, you could have a Capsule Server without Pulp enabled, in which case you do not need the same level of storage requirements for directories related to Pulp such as **/var/lib/pulp/**.

Table 2.1. Storage Requirements for Satellite Server Installation

Directory	Installation Size	Runtime Size with Red Hat Enterprise Linux 5, 6, and 7 synchronized	Considerations
/var/cache/pulp/	1 MB	10 GB (Minimum)	See the notes in this section's introduction.
/var/lib/pulp/	1 MB	500 GB	• Will continue to grow as content is added to Satellite Server. Plan for expansion over time. • Symbolic links cannot be used.

Directory	Installation Size	Runtime Size with Red Hat Enterprise Linux 5, 6, and 7 synchronized	Considerations
/var/lib/mongodb/	3.5 GB	50 GB	• Will continue to grow as content is added to Satellite Server. Plan for expansion over time. • Symbolic links cannot be used. • NFS is not recommended with MongoDB.
/var/log/	10 MB	250 MB	None
/var/lib/pgsql/	100 MB	10 GB	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. It is recommended not to use NFS with PostgreSQL.
/usr	3 GB	Not Applicable	None
/opt	500 MB (Connected Installations)	Not Applicable	Software collections are installed into the **`/opt/rh/`** and **`/opt/theforeman/`** directories. Write and execute permissions by root are required for installation into to the **`/opt`** directory.

Directory	Installation Size	Runtime Size with Red Hat Enterprise Linux 5, 6, and 7 synchronized	Considerations
/opt	2 GB (Disconnected Installations)	Not Applicable	• Software collections are installed into the `/opt/rh/` and `/opt/theforeman/` directories. Write and execute permissions by root are required for installation into to the `/opt` directory. • A copy of the repositories used for installation is stored in this directory.

Table 2.2. Storage Requirements for Capsule Server Installation

Directory	Installation Size	Runtime Size with Red Hat Enterprise Linux 5, 6, and 7 synchronized	Considerations
/var/cache/pulp/	1 MB	10 GB (Minimum)	See the notes in this section's introduction.
/var/lib/pulp/	1 MB	500 GB	• Will continue to grow as content is added. Plan for expansion over time. • Symbolic links cannot be used.

Directory	Installation Size	Runtime Size with Red Hat Enterprise Linux 5, 6, and 7 synchronized	Considerations
/var/lib/mongodb/	3.5 GB	50 GB	• Will continue to grow as content is added. Plan for expansion over time. • Symbolic links cannot be used. • NFS is not recommended with MongoDB.

Log files are written to **`/var/log/messages/`**, **`/var/log/httpd/`**, and **`/var/lib/foreman-proxy/openscap/content/`**. You can manage the size of these files using **logrotate**. For further information, see Log Rotation in the *System Administrator's Guide*.

Storage Recommendations

- Because most Satellite as well as Capsule Server data is stored within the **`/var`** directory, it is strongly recommended to mount **`/var`** on LVM storage, enabling the system to scale.
- Red Hat recommends the usage of high-bandwidth, low-latency storage for the **`/var/lib/pulp/`** and **`/var/lib/mongodb/`** directories. As Red Hat Satellite has many operations that are I/O intensive, usage of high latency, low-bandwidth storage could potentially have issues with performance degradation. It is recommended not to use NFS with MongoDB as MongoDB does not use conventional I/O to access data files and performance problems will occur when both the data files and the journal files are hosted on NFS. If required to use NFS, mount the volumes with the following option in the **`/etc/fstab`** file: **`bg`**, **`nolock`**, and **`noatime`**.
- Do not use the GFS2 file system as the input-output latency has been found to be too high.
- For improved performance, use solid state drives (SSD) rather than hard disk drives (HDD).
- The XFS file system is recommended for Red Hat Satellite 6 because it does not have the inode limitations that **`ext4`** does. As Satellite uses a lot of symbolic links it is likely that your system will run out of inodes if using **`ext4`** and the default number of inodes.
  If you intend to use Red Hat Enterprise Linux 6 instead, contact your account team to learn about enabling XFS on this system. Also consider that long term support for Satellite 6 on Red Hat Enterprise Linux 6 has a shorter lifespan which might necessitate a migration from version 6 to 7 in the future. Red Hat Enterprise Linux 7 is highly recommended for new installations.
- When **`/var/lib/pulp`** directory is mounted using an NFS share, SELinux will block the synchronization process. To avoid this, specify the SELinux context of the **`/var/lib/pulp`** directory in the file system table by adding the following lines to **`/etc/fstab`**:

```
nfs.example.com:/nfsshare  /var/lib/pulp/content  nfs
context="system_u:object_r:httpd_sys_rw_content_t:s0"  1 2
```

If NFS share is already mounted, remount it using the above recommendation and run the following command:

```
# chcon -R system_u:object_r:httpd_sys_rw_content_t:s0 /var/lib/pulp
```

2.3. SUPPORTED OPERATING SYSTEMS

You can install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Satellite Server and Red Hat Satellite Capsule Server are supported only on the latest versions of Red Hat Enterprise Linux 6 Server or 7 Server that is available at the moment when Satellite 6.2 is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

WARNING

With the release of Red Hat Satellite 6.3, Red Hat will no longer support Red Hat Enterprise Linux 6 as an underlying platform for Red Hat Satellite Server and Red Hat Satellite Capsule Server.

Users who are currently running Satellite Server or Capsule Server on Red Hat Enterprise Linux 6 will be provided a mechanism to migrate from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7.

It is strongly recommended that new installations of Satellite Server and Capsule Server use Red Hat Enterprise Linux 7.

Red Hat Satellite Server and Red Hat Satellite Capsule Server require Red Hat Enterprise Linux installations with the **@Base** package group with no other package-set modifications, and without third-party configurations or software not directly necessary for the direct operation of the server. This restriction includes hardening and other non-Red Hat security software. If you require such software 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.

It is recommended that the Satellite Server be a freshly provisioned system. Using the system for anything other than running Satellite is not supported.

If any of the following exist on the system, they must be removed before installation:

- Java virtual machines
- Puppet RPM files
- Additional yum repositories other than those explicitly required in this guide for installation

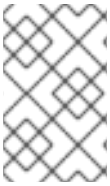
2.4. SUPPORTED BROWSERS

The following web browsers are fully supported:

- Firefox versions 39 and later
- Chrome versions 28 and later

The following web browsers are partially supported. The Satellite web UI interface will function correctly but certain design elements may not align as expected:

- Firefox version 38
- Chrome version 27
- Internet Explorer versions 10 and 11



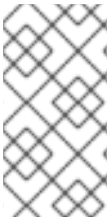
NOTE

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.

2.5. PORTS AND FIREWALLS REQUIREMENTS

Specific network ports must be open and free on the base operating system, as well as open in any network-based firewalls, to enable the components of Satellite architecture to communicate. The tables in this section explain the need for the ports, and the corresponding firewall commands for host-based firewalls are given in the following section. The installation of a Capsule Server will fail if the ports between the Satellite Server and the Capsule Server have not been opened before installation is started.

The tables indicate the destination port and the direction of network traffic, use this information to configure any network-based firewalls. Note that some cloud solutions need to be specifically configured to allow communications between machines as they isolate machines similarly to network-based firewalls.



NOTE

The Satellite Server has an integrated Capsule and any host that is directly connected to the Satellite Server is a Client of the Satellite in the context of these tables. This includes the base system on which a Capsule Server is running. Remember to take this into account when planing any network-based firewall configurations.

Systems which are clients of Capsules, other than the internal Capsule, do not need access to the Satellite Server. See Capsule Networking in the *Red Hat Satellite Architecture Guide* for more information on Satellite Topology.

Required ports can change based on your configuration.

Table 2.3. Ports for Satellite to Red Hat CDN Communication

Port	Protocol	Service	Required For
443	TCP	HTTPS	Subscription Management Services (access.redhat.com) and connecting to the Red Hat CDN (cdn.redhat.com).

Except in the case of a disconnected Satellite, the Satellite Server needs access to the Red Hat CDN.

Table 2.4. Ports for Browser-based User Interface Access to Satellite

Port	Protocol	Service	Required For
443	TCP	HTTPS	Browser-based UI access to Satellite
80	TCP	HTTP	Redirection to HTTPS for web UI access to Satellite (Optional)

Table 2.5. Ports for Client to Satellite Communication

Port	Protocol	Service	Required For
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
8000	TCP	HTTPS	Anaconda to download kickstart templates to hosts, and for downloading iPXE firmware
8140	TCP	HTTPS	Puppet agent to Puppet master connections
9090	TCP	HTTPS	Sending SCAP reports to the Smart Proxy in the integrated Capsule and for the discovery image during provisioning
5000	TCP	HTTPS	Connection to Katello for the Docker registry

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 2.6. Ports for Client to Capsule Communication

Port	Protocol	Service	Required for
80	TCP	HTTP	Anaconda, yum, and for obtaining Katello certificate updates

Port	Protocol	Service	Required for
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
8443	TCP	HTTPS	Subscription Management Services and Telemetry Services
9090	TCP	HTTPS	Sending SCAP reports to the Smart Proxy in the Capsule and for the discovery image during provisioning
5000	TCP	HTTPS	Connection to Katello for the Docker registry
53	TCP and UDP	DNS	Client to Capsule DNS queries to a Capsule's DNS service (Optional)
67	UDP	DHCP	Client to Capsule broadcasts, DHCP broadcasts for Client provisioning from a Capsule (Optional)
69	UDP	TFTP	Clients downloading PXE boot image files from a Capsule for provisioning (Optional)

Table 2.7. Ports for Capsule to Satellite Communication

Port	Protocol	Service	Required For
80	TCP	HTTP	Anaconda, yum, and for obtaining Katello certificate updates
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

Port	Protocol	Service	Required For
5647	TCP	amqp	The Katello agent to communicate with the Satellite's Qpid dispatch router
5000	TCP	HTTPS	Connection to Katello for the Docker registry

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. See the table Ports for Client to Satellite Communication.

Table 2.8. Ports for Satellite to Capsule Communication

Port	Protocol	Service	Required For
443	TCP	HTTPS	Connections to the Pulp server in the Capsule
9090	TCP	HTTPS	Connections to the proxy in the Capsule
80	TCP	HTTP	Downloading a bootdisk (Optional)

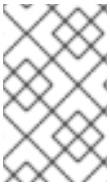
Table 2.9. Ports for Capsule to Client Communication

Port	Protocol	Service	Required For
7	TCP and UDP	ICMP	DHCP Capsule to Client network, ICMP ECHO to verify IP address is free (Optional)
68	UDP	DHCP	Capsule to Client broadcasts, DHCP broadcasts for Client provisioning from a Capsule (Optional)
8443	TCP	HTTP	Capsule to Client "reboot" command to a discovered host during provisioning (Optional)

Table 2.10. Optional Network Ports

Port	Protocol	Service	Required For
22	TCP	SSH	Satellite and Capsule originated communications, for Remote Execution (Rex)

Port	Protocol	Service	Required For
443	TCP	HTTPS	Satellite originated communications, for vCenter compute resource
7911	TCP	DHCP	• Capsule originated commands for orchestration of DHCP records (local or external) • If DHCP is provided by an external service, you must open the port on the external server.
5000	TCP	HTTP	Satellite originated communications, for compute resources in OpenStack or for running Docker containers
22, 16514	TCP	SSH, SSL/TLS	Satellite originated communications, for compute resources in libvirt
389, 636	TCP	LDAP, LDAPS	Satellite originated communications, for LDAP and secured LDAP authentication sources
5900 to 5930	TCP	SSL/TLS	Satellite originated communications, for NoVNC console in web UI to hypervisors



NOTE

A DHCP Capsule sends an ICMP ECHO to confirm an IP address is free, **no response** of any kind is expected. ICMP can be dropped by a networked-based firewall, but **any** response will prevent IP addresses being allocated.

2.6. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER

Systems which are clients of Satellite Server's internal Capsule require access thorough host and networked based firewalls. This section describes configuring the host-based firewall on Satellite Server's base system to enable incoming connections from a Client and to make these rules persistent across system reboots. For more information on the ports used, see Section 2.5, "Ports and Firewalls Requirements".

Configuring the Firewall on Red Hat Enterprise Linux 6

1. Open the ports required for Client to Satellite Communications

```
# iptables -I INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
\
```

```
&& iptables -I INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 67 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 69 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5000 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5647 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 \
-j ACCEPT \
&& iptables -A INPUT -m state --state NEW -p tcp --dport 9090 \
-j ACCEPT \
&& service iptables save
```

2. Verify that the iptables service is started and enabled.

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

Configuring the Firewall on Red Hat Enterprise Linux 7

1. Open the ports required for Client to Satellite Communications.

```
# 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="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="9090/tcp"
```

2. Repeat the command adding the **`--permanent`** option to make the settings persistent.

```
# 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="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="9090/tcp"
```

2.7. ENABLING CONNECTIONS FROM CAPSULE SERVER TO SATELLITE SERVER

Follow this procedure to enable incoming connections from a Capsule Server to a Satellite Server, and make these rules persistent across reboots. If you do not use an external Capsule Server, you do not need to enable this connection.

Prerequisites

A Capsule Server's base system is a client of the Satellite Server, therefore the procedure in Section 2.6, "Enabling Connections from a Client to Satellite Server" should be completed first. This procedure opens the extra ports required by an external Capsule Server.

For more information on the ports used, see Section 2.5, "Ports and Firewalls Requirements".

Configuring the Firewall on Red Hat Enterprise Linux 6

1. Configure iptables service.

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 5000 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5646 -j
ACCEPT \
&& service iptables save
```

2. Start iptables service.

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

Configuring the Firewall on Red Hat Enterprise Linux 7

1. Configure the firewall on Satellite Server.

```
# firewall-cmd --add-port="5000/tcp" --add-port="5646/tcp"
```

2. Repeat the command adding the **`--permanent`** option to make the settings persistent.

```
# firewall-cmd --permanent --add-port="5000/tcp" --add-
port="5646/tcp"
```

2.8. ENABLING CONNECTIONS FROM SATELLITE SERVER AND CLIENTS TO A CAPSULE SERVER

You can enable incoming connections from Satellite Server and clients to Capsule Server and make these rules persistent during reboots. If you do not use an external Capsule Server, you do not need to enable this connection.

For more information on the ports used, see Ports and Firewalls Requirements.

Configuring the Firewall on Red Hat Enterprise Linux 6

1. Configure iptables service.

```
# iptables -I INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
\
&& iptables -I INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 67 -j
ACCEPT \
```

```
&& iptables -I INPUT -m state --state NEW -p udp --dport 69 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5000 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5647 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8000 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8443 \
-j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 9090 \
-j ACCEPT \
&& service iptables save
```

2. Start iptables service.

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

Configuring the Firewall on Red Hat Enterprise Linux 7

1. Configure the firewall on Capsule Server.

```
# 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="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

2. Repeat the command adding the **`--permanent`** option to make the settings persistent.

```
# 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="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

2.9. VERIFYING DNS RESOLUTION

Verifying the full forward and reverse DNS resolution using a fully-qualified domain name enables you to prevent issues while installing Satellite.

Ensure that the host name and local host resolve correctly.

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

Successful name resolution results in output similar to the following:

```
# ping -c1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.043 ms

--- localhost ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.043/0.043/0.043/0.000 ms

# ping -c1 `hostname -f`
PING hostname.gateway (XX.XX.XX.XX) 56(84) bytes of data.
64 bytes from hostname.gateway (XX.XX.XX.XX): icmp_seq=1 ttl=64 time=0.019
ms

--- localhost.gateway ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.019/0.019/0.019/0.000 ms
```

To avoid discrepancies with static and transient host names, set all the host names on the system by entering the following command:

```
# hostnamectl set-hostname name
```

For more information, see the Configuring Host Names Using hostnamectl in the *Red Hat Enterprise Linux 7 Networking Guide*.

WARNING

Name resolution is critical to the operation of Satellite 6. If Satellite cannot properly resolve its fully qualified domain name, many options will fail. Among these options are content management, subscription management, and provisioning.

2.10. CHANGING DEFAULT SELINUX PORTS

Red Hat Satellite 6 uses a set of predefined ports. Because Red Hat recommends that SELinux on Satellite 6 systems be set to permissive or 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. You only need to change these ports if you use non-standard ports.

For example, if you change the Satellite web UI ports (HTTP/HTTPS) to 8018/8019, you need to add these port numbers to the httpd_port_t SELinux port type.

This change is also required for target ports. For example, when Satellite 6 connects to an external source, like Red Hat Virtualization or Red Hat OpenStack Platform.

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

Before You Begin

- SELinux must be enabled and running in permissive or enforcing mode before installing Satellite. For more information, see the *Red Hat Enterprise 6 Security-Enhanced Linux User Guide* or the *Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide*.

Changing default ports to user-specified ports

1. To change the port from the default port to a user-specified port, execute the commands using values that are relevant to your environment. These examples use port 99999 for demonstration purposes.

Default Port	SELinux Command
80, 443, 8443	semanage port -a -t http_port_t -p tcp 99999
8080	semanage port -a -t http_cache_port_t -p tcp 99999
8140	semanage port -a -t puppet_port_t -p tcp 99999
9090	semanage port -a -t websm_port_t -p tcp 99999
69	semanage port -a -t tftp_port_t -p udp 99999
53 (TCP)	semanage port -a -t dns_port_t -p tcp 99999
53 (UDP)	semanage port -a -t dns_port_t -p udp 99999
67, 68	semanage port -a -t dhcpd_port_t -p udp 99999
5671	semanage port -a -t amqp_port_t -p tcp 99999
8000	semanage port -a -t soundd_port_t -p tcp 99999
7911	semanage port -a -t dhcpd_port_t -p tcp 99999
5000 on Red Hat Enterprise Linux 6	semanage port -a -t commplex_port_t -p tcp 99999
5000 on Red Hat Enterprise Linux 7	semanage port -a -t commplex_main_port_t -p tcp 99999
22	semanage port -a -t ssh_port_t -p tcp 99999
16514 (libvirt)	semanage port -a -t virt_port_t -p tcp 99999
389, 636	semanage port -a -t ldap_port_t -p tcp 99999
5910 to 5930	semanage port -a -t vnc_port_t -p tcp 99999

2. Disassociate the previously used port number and port type.

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

CHAPTER 3. INSTALLING SATELLITE SERVER

There are two methods of installing Satellite Server, connected and disconnected. A connected installation enables you to obtain the packages necessary to install Satellite Server by installing them directly from the Red Hat Content Delivery Network (CDN). A disconnected installation enables you to download an ISO image of the packages from an external computer and copy it to the Satellite Server for installation.

For hosts that have network connectivity, Red Hat recommends installing 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.

To successfully install Satellite Server, you must have root access.

3.1. INSTALLING SATELLITE SERVER FROM A CONNECTED NETWORK

Installing Satellite Server from a connected network enables you to obtain packages and receive updates directly from the Red Hat Content Delivery Network.

Note that the Satellite 6 installation program is based on Puppet, which means that any manual configuration changes might be overwritten if you run the installation program more than once. If you wish to avoid this use the **`--noop`** argument when you run the installation program to determine what changes would be applied. This argument ensures that no actual changes are made. Potential changes are written to **`/var/log/katello-installer.log`**

Files are always backed up and so you can revert any unwanted changes. For example, in the katello-installer logs you can see an entry similar to the following about Filebucket:

```
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed
/etc/dhcp/dhcpd.conf to puppet with sum 622d9820b8e764ab124367c68f5fa3a1
```

You can restore the previous file as follows:

```
# puppet filebucket -l \
restore /etc/dhcp/dhcpd.conf 622d9820b8e764ab124367c68f5fa3a1
```

3.1.1. Registering to Red Hat Subscription Management

Registering the host to Red Hat Subscription Management enables the host to subscribe to and consume content for any subscriptions available to the user. This includes content such as Red Hat Enterprise Linux, Red Hat Software Collections (RHSCL), and Red Hat Satellite.

Register your system with the Red Hat Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

The command displays output similar to the following:

```
# subscription-manager register
Username: user_name
Password:
```

```
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-
7683431bcf9a
```

3.1.2. Identifying and Attaching the Satellite Subscription to the Host

After you have registered your host, you need to identify and attach an available Satellite subscription. The Satellite subscription provides access to the Satellite content, as well as Red Hat Enterprise Linux, Red Hat Software Collections (RHSCL), and Red Hat Satellite. This is the only subscription required. Every Red Hat subscription is identified by a Pool ID.

1. Identify your Satellite subscription
   On Red Hat Enterprise Linux 6.7 (or higher) or 7.1 (or higher), you can search all available subscriptions containing the string **`Red Hat Satellite`**. On earlier versions of Red Hat Enterprise Linux, you must list **all** available subscriptions and manually check the output for the appropriate subscription.

   a. On Red Hat Enterprise Linux 6.7 (and higher) or 7.1 (and higher), run the following command:

   ```
   # subscription-manager list --available --matches 'Red Hat
   Satellite'
   ```

   This command performs a case-insensitive search of all available subscriptions' fields, including **`Subscription Name`** and **`Provides`**, matching any instances of **`Red Hat Satellite`**. Subscriptions are classified as available if they are not already attached to a system. The search string may also contain the wildcards **`?`** or **`*`** to match a single character or zero or more characters, respectively. The wildcard characters may be escaped with a backslash to represent a literal question mark or asterisk. Likewise, to represent a backslash, it must be escaped with another backslash.

   If you are unable to find an available Satellite subscription, see the Red Hat Knowledgebase solution How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager? to run a script to allow you to see if your subscription is being consumed by another system.

   b. On other versions of Red Hat Enterprise Linux, run the following command:

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

   If the output is too long, pipe it into a pager utility, such as **`less`** or **`more`**, so that you can look over the output one screenful at a time.

   c. Regardless of which form of the **`subscription-manager`** command is run, the output should be similar to the following:

   ```
   Subscription Name: Red Hat Satellite
   Provides:          Red Hat Satellite 6
                      Red Hat Enterprise Linux Server
                      Red Hat Satellite
                      Red Hat Enterprise Linux Load Balancer (for
   RHEL Server)
   SKU:               MCT0370
   Pool ID:           8a85f9874152663c0541943739717d11
   Available:         3
   Suggested:         1
   ```

```
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2014
System Type:       Physical
```

2. Make a note of the Pool ID so that you can attach it to your Satellite host. Your Pool ID will be different than the example provided.

3. To attach your subscription to your Satellite Server, run the following command, using your Pool ID:

```
# subscription-manager attach --pool=pool_id
```

The output should be similar to the following:

```
Successfully attached a subscription for: Red Hat Satellite
```

4. To verify that the subscriptions are successfully attached, run the following command:

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

The outputs displays something similar to the following:

```
+-------------------------------------------+
   Consumed Subscriptions
+-------------------------------------------+
Subscription Name: Red Hat Satellite
Provides:          Red Hat Satellite
                   Red Hat Enterprise Linux Server
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Satellite
                   Red Hat Satellite 6
                   Red Hat Software Collections  (for RHEL Server)
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
                   Red Hat Satellite with Embedded Oracle
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux High Availability (for
RHEL Server)
SKU:               MCT0370
Contract:          10293569
Account:           5361051
Serial:            1653856191250699363
Pool ID:           8a85f9874152663c0541943739717d11
Active:            True
Quantity Used:     1
Service Level:     Premium
Service Type:      L1-L3
Status Details:
Starts:            10/08/2013
Ends:              10/07/2014
System Type:       Physical
```

3.1.3. Configuring Repositories

1. Disable all existing repositories.

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

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

   a. If you are using Red Hat Enterprise Linux 6, run this command.

```
# subscription-manager repos --enable=rhel-6-server-rpms \
--enable=rhel-server-rhscl-6-rpms \
--enable=rhel-6-server-satellite-6.2-rpms
```

   b. If you are using Red Hat Enterprise Linux 7, run this command.

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-6.2-rpms
```

NOTE

If you are installing Red Hat Satellite as a virtual machine hosted on Red Hat Virtualization (RHV), you also need to enable the **Red Hat Common** repository, and install RHV guest agents and drivers. For more information, see Installing the Guest Agents and Drivers on Red Hat Enterprise Linuxin the *Virtual Machine Management Guide* for more information.

3. Ensure that Red Hat Subscription Manager is not set to use a specific operating system release.

```
# subscription-manager release --unset
```

4. Clear out any metadata left from any non-Red Hat yum repositories.

```
# yum clean all
```

5. Verify that the repositories have been enabled.

```
# yum repolist enabled
```

The following output displays:

```
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-6.2-rpms/x86_64    Red Hat Satellite 6.2
(for RHEL 7 Server) (RPMs)                             545
```

```
!rhel-server-rhscl-7-rpms/x86_64              Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server     4,279
repolist: 14,713
```

3.1.4. Installing the Satellite Server Packages

You must update all packages before installing the Satellite Server packages. After installation, you must perform the initial configuration of Satellite Server, including configuring server certificates, setting your user name, password, and the default organization and location.

1. Update all packages.

```
# yum update
```

2. Install the installation package.

```
# yum install satellite
```

3. Go to Section 3.3, “Performing the Initial Configuration” to run the installer program and perform the initial configuration of your Satellite Server.

3.2. DOWNLOADING AND INSTALLING FROM A DISCONNECTED NETWORK

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 might not contain the latest updates, bug fixes, and functionality.

NOTE

If the base system has not been updated from the Red Hat CDN, package dependency errors are possible. The latest version of the required packages will have to be downloaded and installed manually. See Section 3.2.4, “Downloading Packages Manually” for more information.

Before You Begin

- 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.

3.2.1. Downloading the Binary DVD Images

1. Go to Red Hat Customer Portal and log in.
2. Click **DOWNLOADS**.
3. Select **Red Hat Enterprise Linux**.
4. Ensure that you have the correct product and version for your environment.
   - **Product Variant** is set to **Red Hat Enterprise Linux Server**.
   - **Version** is set to the latest minor version of the product you plan to use as the base system.

- **Architecture** is set to the 64 bit version.

5. On the **Product Software** tab, download the Binary DVD image for the latest Red Hat Enterprise Linux Server version.
6. Click **DOWNLOADS** and select **Red Hat Satellite**.
7. Ensure that you have the correct product and version for your environment.
   - **Product Variant** is set to **Red Hat Satellite**.
   - **Version** is set to the latest minor version of the product you plan to use as the base system.
   - **Architecture** is set to the 64 bit version.
8. On the **Product Software** tab, download the Binary DVD image for the latest Red Hat Satellite version.
9. Copy the ISO files to the Satellite base system or other accessible storage device.

```
# scp localfile username@hostname:remotefile
```

3.2.2. Configuring the Base System with Offline Repositories

1. Create a directory to serve as the mount point for the ISO file corresponding to the base system's version.

```
# mkdir /media/rhelX-server
```

   Where *X* is the major version of Red Hat Enterprise Linux you are using.

2. Mount the ISO image for Red Hat Enterprise Linux to the mount point.

```
# mount -o loop rhelX-Server-DVD.iso /media/rhelX-server
```

   The following example shows mounting the ISO using Red Hat Enterprise Linux 7.2:

```
# mount -o loop RHEL-7.2-20151030.0-Server-x86_64-dvd1.iso \
/media/rhel7-server
mount: /dev/loop0 is write-protected, mounting read-only
```

3. Copy the ISO file's repository data file.

```
# cp /media/rhelX-server/media.repo /etc/yum.repos.d/rhelX-
server.repo
```

4. Edit the repository data file and add the **`baseurl`** directive.

```
baseurl=file:///media/rhelX-server/
```

   The following example shows the repository data file using Red Hat Enterprise Linux 7.2:

```
# vi /etc/yum.repos.d/rhel7-server.repo
[InstallMedia]
```

```
name=Red Hat Enterprise Linux 7.2
mediaid=1446216863.790260
metadata_expire=-1
gpgcheck=0
cost=500
baseurl=file:///media/rhel7-server/
enabled=1
```

5. Verify that the repository has been configured.

```
# yum repolist
Loaded plugins: product-id, search-disabled-repos, subscription-
manager
This system is not registered to Red Hat Subscription Management.
You can use subscription-manager to register.
repo id          repo name                        status
InstallMedia     Red Hat Enterprise Linux 7.2    4,620
```

6. Create a directory to serve as the mount point for the ISO file of the Satellite Server.

```
# mkdir /media/sat6
```

7. Mount the ISO image for Red Hat Satellite Server to the mount point.

```
# mount -o loop sat6-DVD.iso /media/sat6
```

The following example shows mounting the ISO using Red Hat Satellite 6.2.1 for Red Hat Enterprise Linux 7:

```
# mount -o loop satellite-6.2.1-rhel-7-x86_64-dvd.iso /media/sat6
mount: /dev/loop1 is write-protected, mounting read-only
```

3.2.3. Installing from the Offline Repositories

1. Import the Red Hat GPG keys.

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

2. Ensure the base system is up to date with the Binary DVD image.

```
# yum update
```

3. Change to the directory where the Satellite ISO is mounted.

```
# cd /media/sat6/
```

4. Run the installer script in the mounted directory.

```
# ./install_packages
 This script will install the foreman packages on the current
machine.
   - Ensuring we are in an expected directory.
   - Copying installation files.
```

```
   - Creating a Repository File
   - Creating RHSCL Repository File
   - Checking to see if Foreman is already installed.
   - Importing the gpg key.
   - Foreman is not yet installed, installing it.
   - Installation repository will remain configured for future
package installs.
   - Installation media can now be safely unmounted.

Install is complete. Please run satellite-installer.
```

If the script fails due to missing or outdated packages, you will need to download and install these separately. See Section 3.2.4, “Downloading Packages Manually” for instructions.

If the script fails due to installed packages being newer than those required, enter **`yum distribution-synchronization`** to downgrade the installed packages to the versions that came from the Red Hat Enterprise Linux ISO, then run the installation script again. This should only occur if you have repositories configured whose source is not the Red Hat Enterprise Linux ISO. Use of such repositories is an unsupported configuration.

5. For a self-registered Satellite, disable the ISO based repositories to avoid conflicts with repositories provided by Satellite Server.

   a. Install **`yum-config-manager`**:

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

   b. Disable the ISO based repositories:

   ```
   # yum-config-manager --disable InstallMedia --disable satellite-
   local --disable scl-local --disable satellite-puppet4
   ```

   c. Confirm **`yum`** repositories are disabled:

   ```
   # yum repolist
   ```

3.2.4. Downloading Packages Manually

If required to download a package manually, proceed as follows:

1. Go to Red Hat Customer Portal and log in.
2. Click **DOWNLOADS**.
3. Select **Red Hat Satellite**.
4. Ensure that you have the correct product and version for your environment.
   - **Product Variant** is set to **Red Hat Satellite**.
   - **Version** is set to the latest minor version of the product you are using as the base system.
   - **Architecture** is set to the 64 bit version.
5. On the **Packages** tab, enter the name of the package required in the Search box.

6. Click **Download Latest** next to the package required.

3.3. PERFORMING THE INITIAL CONFIGURATION

As part of the initial configuration, you can configure a custom server certificate and either manually configure Satellite or automatically configure Satellite using an answer file.

- Manual Configuration - Satellite Server has default initial configuration options that prepare the server for use. You can override these settings depending on your environment's requirements. You can run the command as often as needed to configure any necessary options.
- Automatic Configuration - You can automate most of the installation and configuration by using an answer file.

NOTE

Depending on the options that you use when running the Satellite installer, the configuration can take several minutes to complete.

Before you continue, consider which manifests or packages are relevant for your environment. See the Content Management Guide for more information.

3.3.1. Synchronizing Time

You must start and enable a time synchronizer on the host operating system to minimize the effects of time drift. If a system's time is incorrect, certificate verification can fail.

Two time synchronizers are available: **`NTP`** and **`chrony`**. Each of these has its advantages. **`chrony`** is recommended for systems that are frequently suspended and for systems—such as mobile and virtual systems—that intermittently disconnect from networks and then reestablish network connection. **`NTP`** is recommended for systems that are expected to remain in running states and that are expected to be connected to a network without interruption.

For more information on the differences between **`NTP`** and **`chrony`**, see Differences Between ntpd and chronyd in the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

Synchronizing Time by Using NTP

1. Install ntp.

```
# yum install ntp
```

2. Verify that your NTP server is available.

```
# ntpdate -q ntp_server_address
```

3. Set the system time.

```
# ntpdate ntp_server_address
```

4. Start and enable the ntpd service.

```
# chkconfig ntpd on
```

Synchronizing Time by Using chronyd

1. Install chronyd.

```
# yum install chrony
```

2. Start and enable the chrony service.

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

3.3.2. Installing the SOS Package on the Host Operating System

You should install the **sos** package on the host operating system. The **sos** package enables you to collect configuration and diagnostic information from a Red Hat Enterprise Linux system. You can also use it to provide the initial system analysis, which is required when opening a service request with Red Hat Technical Support. For more information on using sos, see the Knowledgebase solution What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later? on the Red Hat Customer Portal.

Install the **sos** package.

```
# yum install sos
```

3.3.3. Performing the Initial Configuration Manually

The initial configuration creates an organization, location, user name, and password. After the initial configuration, you can create additional organizations and locations if required.

The installation process can take tens of minutes to complete. If you are connecting remotely to the system, consider using a utility such as **`screen`** that allows suspending and reattaching a communication session so that you can check the installation progress in case you become disconnected from the remote system. The Red Hat Knowledgebase article How to use the screen command describes installing **`screen`**; alternately see the **`screen`** manual page for more information. If you lose connection to the shell where the installation command is running, see the log at **`/var/log/foreman-installer/satellite.log`** to determine if the process completed successfully.

Manually configuring Satellite Server

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

It is recommended to specify a meaningful value for the option: **`--foreman-initial-organization`**. This may be your company name. An internal label that matches the value is also created and cannot be changed later on. If you do not specify a value, an organization called **Default Organization** with the label **Default_Organization** is created. You can rename the organization name but not the label.

By default, all configuration files configured by the installer are managed by Puppet. When **`satellite-installer`** is rerun, any manual changes to the Puppet managed files will be overwritten with the initial values. If you want to be able to manage the DNS files and DHCP files manually, use the **`--foreman-proxy-dns-managed=false`** and **`--foreman-proxy-dhcp-managed=false`** options so that the

files related to the respective services will not be managed by Puppet. For more information on how to apply custom configuration on other services, see Appendix C, *Applying Custom Configuration to Red Hat Satellite*.

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-admin-username admin-username \
--foreman-admin-password admin-password \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

When the script completes successfully, the following output is displayed:

```
Installing             Done
   [100%] [.....................................]
   Success!
   * Satellite 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/foreman-installer/satellite.log
```

If you have been installing in a disconnected environment, unmount the ISO images.

```
# umount /media/sat6
# umount /media/rhel7-server
```

3.3.4. 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 and after you run **satellite-installer** the first time, the answer file is populated with the standard parameter values for installation.

You should use the FQDN instead of the IP address where possible in case of network changes.

1. Copy the default answer file **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** to a location on your local file system.

   ```
   # cp /etc/foreman-installer/scenarios.d/satellite-answers.yaml \
   /etc/foreman-installer/scenarios.d/my-answer-file.yaml
   ```

2. To view all of the configurable options, run the **satellite-installer --scenario satellite --help** command.

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

4. Open the **/etc/foreman-installer/scenarios.d/satellite.yaml** file and edit the answer file entry to point to your custom answer file.

```
:answer_file: /etc/foreman-installer/scenarios.d/my-answer-file.yaml
```

5. Run the **satellite-installer** command.

```
# satellite-installer --scenario satellite
```

6. If you have been installing in a disconnected environment, unmount the ISO images.

```
# umount /media/sat6
# umount /media/rhel7-server
```

3.4. CREATING AND INSTALLING MANIFESTS

The Customer Portal page for Satellite Server provides the ability to collect a group of subscriptions and attach them to the Satellite for distribution to managed systems. To do that, create a Subscription Manifest for your Satellite Server.

Creating a Manifest

1. Navigate to the Red Hat Customer Portal and log in.
2. Click **Subscriptions**.
3. In the Red Hat Subscription Management section, click **Satellite Organizations**.

NOTE

You cannot create a new Subscription Manifest if you have no active subscriptions. This can be a Red Hat Enterprise Linux subscription. If you do not have the correct subscription the **Create a Satellite** button will be greyed out.

4. On the **Subscription Management Applications** page, select the **Satellite** tab.
5. Click **Create a Satellite**.
6. In the **Name** field, type the host name of the Satellite Server.
7. Select **Satellite 6.2** as the version and click **Create**.
8. Click **Attach a subscription**.
9. Select the check box for each subscription that you want to attach and specify the number of subscriptions.
10. Click **Attach Selected**.
    It can take several minutes for all the subscriptions to attach.
11. Click **Download Manifest** and save the manifest file to a known location.

Uploading a Manifest to Your Satellite Server

Both the Red Hat Satellite 6 Web UI and CLI provide methods for importing the manifest.

Uploading a Manifest Using the Web UI

1. Verify that you are in the correct Organization.
2. Click **Content** > **Red Hat Subscriptions**.
3. Click **Manage Manifest** to open the Subscriptions page.
4. Click **Choose File**, select the manifest file you created, and click **Open**.
5. Click **Upload** to upload the manifest to the Satellite Server.

Uploading a Manifest Using Hammer CLI

1. Upload a manifest to Satellite Server.

```
# hammer subscription upload --organization-label org_label \
--file path_to_manifest
```

When you have completed this section, you can enable repositories and import Red Hat content. This is a prerequisite for some of the following procedures. See Importing Red Hat Content in the *Red Hat Satellite Content Management Guide* for more information.

3.5. PERFORMING ADDITIONAL CONFIGURATION

3.5.1. 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 Content Delivery Network (CDN). A self-registered Red Hat Satellite Server is registered to itself rather than the Red Hat Customer Portal. The following items are some highlights and limitations of the feature:

- You can subscribe Satellite Server to Content Views and manage updates to the Satellite Server as other managed hosts. A common scenario is applying base operating system updates to all managed Red Hat Enterprise Linux hosts, including the Satellite Server. For example, you can create a Composite Content View including a Red Hat Enterprise Linux 7 Content View and a Satellite 6 Content View and apply it to the Satellite. The Satellite Server Content Views should only contain the required repositories listed in the following procedure. Allowing Satellite Server access to non-required repositories can create potential issues.
- Though a self-registered Satellite allows you to update the Satellite Server through the web UI, you will still need to run **`satellite-installer`** to upgrade it for y-stream releases (for example, Satellite 6.1 to Satellite 6.2) and z-stream releases (for example, Satellite 6.2.7 to Satellite 6.2.8). For more information on upgrading a self-registered Satellite Server, see Section 6.8, “Upgrading a Self-Registered Satellite Server”. For more information on updating a self-registered Satellite for z-stream releases, see Chapter 7, *Updating Satellite Server, Capsule Server, and Content Hosts*.
- If you have a single self-registered Satellite Server, you should always make a full backup before doing an upgrade to untested packages. Upgrading a self-registered Satellite cannot be tested by using life-cycle environments.
- Not all Puppet modules are supported by a self-registered Satellite. When applying Puppet modules to a self-registered Satellite, ensure that they will not create an unsupported configuration.

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.

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 Managing Subscriptions in the *Content Management Guide*.

   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 the button **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 UI or with the command-line interface:

   - **Using the Satellite web UI:**

     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 6.2 for Red Hat Enterprise Linux 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 6.2 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 7Server
        - Red Hat Satellite 6.2 for Red Hat Enterprise Linux 7 Server RPMs x86_64
        - Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64 7Server
        - Red Hat Satellite Tools 6.2 for Red Hat Enterprise Linux 7 Server RPMs x86_64

- **Using the Subscription Manager CLI Tool:**
  You can enable the repositories required for the Satellite Server by using the following command:

    - **For Red Hat Enterprise Linux 6**:

```
# subscription-manager repos --enable=rhel-6-server-satellite-6.2-rpms \
--enable=rhel-6-server-satellite-tools-6.2-rpms \
--enable=rhel-6-server-rpms \
--enable=rhel-server-rhscl-6-rpms
```

    - For **Red Hat Enterprise Linux 7**:

```
# subscription-manager repos --enable=rhel-7-server-satellite-6.2-rpms \
--enable=rhel-7-server-satellite-tools-6.2-rpms \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms
```

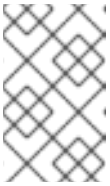
5. Synchronize the Satellite Server:

    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

more information on Content Views see Using Content Views in the *Red Hat Satellite Host Configuration Guide*.

a. To create a Content View:

i. Log into the web UI 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. When registering the Satellite Server, you must specify the organization to which the server belongs, and the life cycle environment. To confirm the available organizations and life cycle environments, in the Satellite web UI navigate to **Hosts** > **New host** and select the drop-down list for these values.

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

Example

```
# 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 Users and Roles in the *Red Hat Satellite Server Administration Guide* for more information.

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

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

10. 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.

11. Install the Katello Agent package to allow errata management and package installation through the Satellite web UI. The **katello-agent** package depends on the gofer package that provides the goferd service. The goferd service must be running 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
```

12. Ensure **goferd** is running:
    - On Red Hat Enterprise Linux 6, run the following command:

      ```
      # service goferd start
      ```

    - On Red Hat Enterprise Linux 7, run the following command:

      ```
      # systemctl start goferd
      ```

3.5.2. Installing the Satellite Tools Repository

The Satellite Tools repository provides the **`katello-agent`** and **`puppet`** packages for clients registered to Satellite Server. Installing the katello agent is recommended to allow remote updates of clients. The base system of a self-registered Satellite Server or of a Capsule Server is a client of Satellite Server and therefore should also have the katello agent installed.

To Install the Satellite Tools Repository:

1. In the Satellite web UI, go to **Content** > **Red Hat Repositories** and select the **RPMs** tab.
2. Find and expand the Red Hat Enterprise Linux Server item.
3. Find and expand the Red Hat Satellite Tools 6.2 (for Red Hat Enterprise Linux *VERSION* Server) (RPMs) item.
   If the Red Hat Satellite Tools 6.2 items are not visible, it may be because they are not included in the Subscription Manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the Subscription Manifest and import it into Satellite.
4. Select the **Enabled** check box next to the Satellite 6.2 Tools repository's name.

Enable the Satellite Tools repository for every supported major version of Red Hat Enterprise Linux running on your hosts. After enabling a Red Hat repository, a Product for this repository is automatically created.

To Synchronize the Satellite Tools Repository:

1. Go to **Content** > **Sync Status**.
   A list of product repositories available for synchronization is displayed.

2. Click the arrow next to the product content to view available content.
3. Select the content you want to synchronize.
4. Click **Synchronize Now**.

3.5.3. Configuring Satellite Server with HTTP Proxy

If your network uses an HTTP Proxy, you can enable it. Use the FQDN instead of the IP address where possible in case of network changes.

1. Verify that the **`http_proxy`**, **`https_proxy`**, and **`no_proxy`** variables are not set.

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

2. Run **`satellite-installer`** with the HTTP proxy options.

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

3. Verify that Satellite Server can connect to the Red Hat Content Delivery Network (CDN) and can synchronize its repositories.

   a. On the network gateway and the HTTP Proxy, enable TCP for the following host names:

Host name	Port	Protocol
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS
cert-api.access.redhat.com (if using Red Hat Insights)	443	HTTPS
api.access.redhat.com (if using Red Hat Insights)	443	HTTPS

   For a list of IP addresses used by the Red Hat CDN (cdn.redhat.com), see the Knowledgebase article Public CIDR Lists for Red Hat on the Red Hat Customer Portal.

   b. On Satellite Server, complete the following details in the **`/etc/rhsm/rhsm.conf`** file:

```
# 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 =
```

NOTE

SELinux ensures access of Red Hat Satellite 6 and Red Hat Subscription Manager to specific ports only. In the case of the HTTP cache, the TCP ports are 8080, 8118, 8123, and 10001 - 10010.

To list the ports permitted by SELinux for the HTTP cache, use a command as follows:

```
# semanage port -l | grep http_cache
http_cache_port_t       tcp    8080, 8118, 8123, 10001-10010
[output truncated]
```

To configure SELinux to permit a port for the HTTP cache, for example 8088, use a command as follows:

```
# semanage port -a -t http_cache_port_t -p tcp 8088
```

For more information on SELinux port settings, see Section 2.10, “Changing Default SELinux ports”.

3.5.4. Enabling Power Management on Managed Hosts

When you enable the baseboard management controller (BMC) module on Satellite Server, you can use power management commands on managed hosts using the intelligent platform management interface (IPMI) or a similar protocol.

The BMC service enables 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).

For more information on the BMC service, see Configuring an Additional Network Interface in *Managing Hosts*.

Before You Begin

- All managed hosts must have a network interface, with type **BMC**. Satellite uses this NIC to pass the appropriate credentials to the host.

Enable Power Management on Managed Hosts

1. Run the installer with the options to enable BMC.

```
# satellite-installer --foreman-proxy-bmc "true" \
--foreman-proxy-bmc-default-provider "freeipmi"
```

3.5.5. Configuring DNS, DHCP, and TFTP on Satellite Server

You can configure DNS, DHCP, and TFTP on Satellite Server.

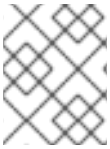
If you want to configure external services, see Chapter 5, *Configuring External Services* for more information.

If you want to disable these services in Satellite in order to manage them manually, see Section 3.5.6, "Disabling DNS, DHCP, and TFTP for Unmanaged Networks" for more information.

To view a complete list of configurable options, run the **`satellite-installer --scenario satellite --help`** command.

Before You Begin

- Contact your network administrator to ensure that you have the correct settings.
- You should have the following information available:
  - DHCP IP address ranges
  - DHCP gateway IP address
  - DHCP nameserver IP address
  - DNS information
  - TFTP server name
- Use the FQDN instead of the IP address where possible in case of network changes.



NOTE

The information in the task is an example. You should use the information relevant to your own environment.

Configure DNS, DHCP, and TFTP on Satellite Server

1. Run **`satellite-installer`** with the options appropriate for your environment.

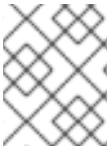
```
# satellite-installer --scenario satellite \
--foreman-proxy-dns true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-servername $(hostname)
```

The status of the installation is displayed. You can view the user name and password in the command output. You can also retrieve the information from the **admin_password** parameter in the **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** file.

```
Success!
  * Satellite 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/foreman-installer/satellite.log
```



NOTE

Any changes to the settings require running **satellite-installer** again. You can run the script multiple times and it updates all configuration files with the changed values.

3.5.6. Disabling DNS, DHCP, and TFTP for Unmanaged Networks

Satellite 6 provides full management capabilities for TFTP, DHCP, and DNS network services running on Satellite's internal or external Capsules. If you want to manage those services manually or use some external method, then Satellite 6 cannot directly integrate with them. While it is possible to develop custom integration scripts via Foreman Hooks (such as creating DNS records after a new host is created), this integration, also known as orchestration, must be disabled in order to prevent DHCP and DNS validation errors.

1. Go to **Infrastructure** > **Subnets** and select a subnet.
2. On the **Capsules** tab, ensure that there is no DHCP Capsule or TFTP Capsule associated by setting the drop-down list to **None**.
3. Disable forward record orchestration.
    a. Go to **Infrastructure** > **Domains** and select a domain.
    b. On the **Domain** tab, setting the **DNS Capsule** drop-down list to **None**.
4. Disable reverse (PTR) record orchestration.
    a. Go to **Infrastructure** > **Subnets** and select a subnet.
    b. On the **Capsules** tab, setting the **Reverse DNS Capsule** drop-down list to **None**.

NOTE

Satellite 6 does not perform orchestration when a Capsule is not set for a given subnet and domain. When enabling or disabling Capsule associations, orchestration commands for existing hosts can fail if the expected records and configuration files are not present. When associating a Capsule in order to turn orchestration on, make sure the required DHCP and DNS records as well as the TFTP files are in place for existing Satellite 6 managed hosts in order to prevent host deletion failures in the future.

3.5.7. Configuring Satellite Server for Outgoing Emails

To send email messages from Satellite Server, you can use either an **SMTP** server, or the **`sendmail`** command.

1. Edit the configuration file ***/etc/foreman/email.yaml*** to match your preferred delivery method. The following example shows the contents of the configuration file for using an SMTP server:

```
production:
   delivery_method: :smtp
   smtp_settings:
      address: smtp.example.com
      port: 25
      domain: example.com
      authentication: :login
      user_name: satellite@example.com
      password: satellite
```

   Where the **`user_name`** and **`password`** directives specify the login credentials for the SMTP server. The default ***/etc/foreman/email.yaml*** contains **`authentication: :none`**.

   The following example uses **gmail.com** as an SMTP server:

```
production:
   delivery_method: :smtp
   smtp_settings:
      enable_starttls_auto: true
      address: smtp.gmail.com
      port: 587
      domain: smtp.gmail.com
      authentication: :plain
      user_name: user@gmail.com
      password: password
```

   The following example uses the **`sendmail`** command as a delivery method:

```
production:
   delivery_method: :sendmail
   sendmail_settings:
      arguments: "-i -t -G"
```

   Where the **`arguments`** directive is used to pass command-line options to **`sendmail`**. The default value of **`arguments`** is "-i -t". For more information see the **sendmail 1** man page.

2. If you decide to send email via an SMTP server which uses TLS authentication, also perform one of the following steps:

   - Mark the CA certificate of the SMTP server as trusted. To do so, execute the following commands on Satellite Server:

```
# cp mailca.crt /etc/pki/ca-trust/source/anchors/
# update-ca-trust enable
# update-ca-trust
```

     Where ***`mailca.crt`*** is the CA certificate of the SMTP server.

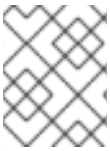
- Alternatively, add the following directive to ***/etc/foreman/email.yaml*** under smtp_settings:

```
enable_starttls_auto: false
```

3. After updating the ***/etc/foreman/email.yaml*** file, restart Katello services to apply the changes.

```
# katello-service restart
```

4. Additional email settings, such as the reply address or subject prefix, can be set up in the Satellite web UI at **Administer** > **Settings** under the **General** tab.



NOTE

For information on configuring email notifications for individual users or user groups, see Configuring Email Notifications in the *Red Hat Satellite Server Administration Guide*.

3.5.8. Configuring Satellite Server with a Custom Server Certificate

SSL certificates are used to protect information and enable secure communication. Red Hat Satellite 6 creates self-signed SSL certificates to enable encrypted communications between the Satellite Server, external Capsule Servers, and all hosts. Instead of using these self-signed certificates, you can install custom SSL certificates issued by a Certificate Authority which is an external, trusted company. For example, your company might have a security policy stating that SSL certificates must be obtained from a Certificate Authority. To obtain the certificate, create a Certificate Signing Request and send it to the Certificate Authority, as described in Section 3.5.8.1, “Obtain an SSL Certificate for the Satellite Server”. In return, you receive a signed SSL certificate.

NOTE

Obtain custom SSL certificates for the Satellite Server and all external Capsule Servers **before** starting this procedure.

To use a custom certificate on Satellite Server, complete these steps:

1. Section 3.5.8.1, “Obtain an SSL Certificate for the Satellite Server”
2. Section 3.5.8.2, “Validate the Satellite Server’s SSL Certificate”
3. Section 3.5.8.3, “Run the Satellite Installer with Custom Certificate Parameters”
4. Section 3.5.8.4, “Install the New Certificate on all Hosts Connected to the Satellite Server”

If you have external Capsule Servers, you must also complete the steps in Section 4.7.6, “Configuring Capsule Server with a Custom Server Certificate”.

3.5.8.1. Obtain an SSL Certificate for the Satellite Server

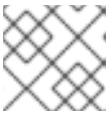
NOTE

If you already have a custom SSL Certificate for the Satellite Server, skip this procedure.

1. Create a directory to contain all the source certificate files, accessible to only the **`root`** user. In these examples, the directory is **`/root/sat_cert`**.

```
# mkdir /root/sat_cert
# cd /root/sat_cert
```

2. Create a private key with which to sign the Certificate Signing Request (CSR).

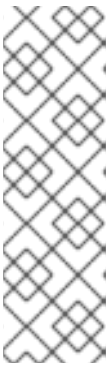


NOTE

If you already have a private key for the Satellite Server, skip this step.

```
# openssl genrsa -out /root/sat_cert/satellite_cert_key.pem 4096
```

3. Create a Certificate Signing Request (CSR)
A Certificate Signing Request is a text file containing details of the server for which you are requesting a certificate. For this command, you provide the private key (output by the previous step), answer some questions about the Satellite Server, and the Certificate Signing Request is created.



NOTE

The certificate's Common Name (CN) must match the fully-qualified domain name (FQDN) of the server on which it is used. If you are requesting a certificate for a Satellite Server, this is the FQDN of the Satellite Server. If you are requesting a certificate for a Capsule Server, this is the FQDN of the Capsule Server.

To confirm a server's FQDN, run the following command on that server: **`hostname -f`**.

```
# openssl req -new \
  -key /root/sat_cert/satellite_cert_key.pem \ 1
  -out /root/sat_cert/satellite_cert_csr.pem   2
```

1. Satellite Server's private key, used to sign the certificate
2. Certificate Signing Request file

Example Certificate Signing Request session

```
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:Queensland
Locality Name (eg, city) [Default City]:Brisbane
Organization Name (eg, company) [Default Company Ltd]:Example
```

```
Organizational Unit Name (eg, section) []:Sales
Common Name (eg, your name or your server's hostname)
[]:satellite.example.com
Email Address []:example@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:Example
```

4. Send the certificate request to the Certificate Authority.
   When you submit the request, be sure to specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the Certificate Authority for the preferred method. In response to the request you can expect to receive a Certificate Authority bundle, and a signed certificate, in separate files.

3.5.8.2. Validate the Satellite Server's SSL Certificate

Run the **katello-certs-check** command with the required parameters as per the following example. This validates the input files required for custom certificates and outputs the commands necessary to install them on the Satellite Server, all Capsule Servers, and hosts under management with Satellite.

1. Validate the custom SSL certificate input files. Change the files' names to match your files.

```
# katello-certs-check \
   -c /root/sat_cert/satellite_cert.pem \        (1)
   -k /root/sat_cert/satellite_cert_key.pem \    (2)
   -r /root/sat_cert/satellite_cert_csr.pem \    (3)
   -b /root/sat_cert/ca_cert_bundle.pem          (4)
```

(1) Certificate file for the Satellite Server, signed by your Certificate Authority

(2) Satellite Server's private key, used to sign the certificate

(3) Certificate signing request file for the Satellite Server

(4) Certificate Authority bundle

If you do not have a request file, see the following Red Hat Knowledgebase article We do not have certificate request (CSR) file for the custom certificate, how can we complete the satellite v 6.2 installation using satellite-installer command?

Example output of `katello-certs-check`

```
Validating the certificate subject=
/C=AU/ST=Queensland/L=Brisbane/O=Example/OU=Sales/CN=satellite.example.com
/emailAddress=example@example.com
Check private key matches the certificate: [OK]
Check ca bundle verifies the cert file: [OK]

Validation succeeded.

To install the Satellite main server with the custom certificates, run:
```

```
satellite-installer --scenario satellite \
--certs-server-cert "/root/sat_cert/satellite_cert.pem" \
--certs-server-cert-req "/root/sat_cert/satellite_cert_csr.pem" \
--certs-server-key "/root/sat_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/sat_cert/ca_cert_bundle.pem"

To update the certificates on a currently running Satellite installation,
run:

satellite-installer --scenario satellite \
--certs-server-cert "/root/sat_cert/satellite_cert.pem" \
--certs-server-cert-req "/root/sat_cert/satellite_cert_csr.pem" \
--certs-server-key "/root/sat_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/sat_cert/ca_cert_bundle.pem" \
--certs-update-server --certs-update-server-ca

To use them inside a $CAPSULE, run this command INSTEAD:

capsule-certs-generate --capsule-fqdn "" \
--certs-tar  "/root/certs.tar" \
--server-cert "/root/sat_cert/satellite_cert.pem" \
--server-cert-req "/root/sat_cert/satellite_cert_csr.pem" \
--server-key "/root/sat_cert/satellite_cert_key.pem" \
--server-ca-cert "/root/sat_cert/ca_cert_bundle.pem" \
--certs-update-server
```

3.5.8.3. Run the Satellite Installer with Custom Certificate Parameters

Now that you have created an SSL certificate and verified it is valid for use with Red Hat Satellite 6, the next step is to install the custom SSL certificate on the Satellite Server and all its hosts.

There is a minor variation to this step, depending on whether or not the Satellite Server is already installed. If it is **already** installed, the existing certificates must be *updated* with those in the certificates archive.

The commands in this section are output by the **`katello-certs-check`** command, as detailed in Section 3.5.8.2, “Validate the Satellite Server’s SSL Certificate”, and can be copied and pasted into a terminal.

1. Run the **`satellite-installer`** command, depending on your situation:

    a. If Satellite is already installed, run the following command on the Satellite Server:

    ```
    # satellite-installer --scenario satellite \
    --certs-server-cert "/root/sat_cert/satellite_cert.pem" \
    --certs-server-cert-req "/root/sat_cert/satellite_cert_csr.pem" \
    --certs-server-key "/root/sat_cert/satellite_cert_key.pem" \
    --certs-server-ca-cert "/root/sat_cert/ca_cert_bundle.pem" \
    --certs-update-server --certs-update-server-ca
    ```

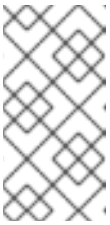
    Important parameters in this command include **`--certs-update-server`** and **`--certs-update-server-ca`**, which specify that the server’s SSL certificate and certificate authority are to be updated. For a brief description of all the installer’s parameters, run the command: **`satellite-installer --scenario satellite --help`**.

NOTE

For all files in the **satellite-installer** command, use full path names, not relative path names. The installer records all files' paths and names, and if you run the installer again, but from a different directory, it may fail as it is unable to find the original files.

b. If Satellite is **not** already installed, run the following command on the Satellite Server:

```
# satellite-installer --scenario satellite \
--certs-server-cert "/root/sat_cert/satellite_cert.pem" \
--certs-server-cert-req "/root/sat_cert/satellite_cert_csr.pem" \
--certs-server-key "/root/sat_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/sat_cert/ca_cert_bundle.pem"
```



NOTE

For all files in the **satellite-installer** command, use full path names, not relative path names. The installer records all files' paths and names, and if you run the installer again, but from a different directory, it may fail as it is unable to find the original files.

2. Verify the certificate has been successfully installed on the Satellite Server before installing it on hosts. On a computer with network access to the Satellite Server, start a web browser, navigate to the URL **https://*satellite.example.com*** and view the certificate's details.

3.5.8.4. Install the New Certificate on all Hosts Connected to the Satellite Server

Now that the custom SSL certificate has been installed on the Satellite Server, it must also be installed on every host registered to the Satellite Server. Run the following commands on all applicable hosts.

1. Delete the current **katello-ca-consumer** package on the host.

```
# yum remove 'katello-ca-consumer*'
```

2. Install the custom SSL certificate on the host.

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

3.5.9. Restricting Access to mongod

Only the **apache** and **root** users should be allowed access to the MongoDB database daemon, **mongod**, to reduce the risk of data loss.

Restrict access to **mongod** on Satellite and Capsule Servers using the following commands.

Configuring the Firewall on Red Hat Enterprise Linux 6

1. Configure **iptables** service on Satellite and Capsule Servers.

```
# iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner \
```

```
--uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -j DROP \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -j DROP \
&& service iptables save
```

Configuring the Firewall on Red Hat Enterprise Linux 7

1. Configure the firewall on Satellite and Capsule Servers.

```
# firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP
```

2. Repeat the command adding the **`--permanent`** option to make the settings persistent.

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
```

```
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 28017 -j DROP
```

CHAPTER 4. INSTALLING CAPSULE SERVER

Before you install Capsule Server, you should ensure that your environment meets the requirements for installation. Capsule Server has the same requirements for installation as Satellite Server. For more information, see Section 2.1, “System Requirements”.

4.1. REGISTERING CAPSULE SERVER TO SATELLITE SERVER

Before You Begin

- The Satellite Server must have a manifest installed with the appropriate repositories for the organization you intend to subscribe to. The manifest must contain repositories for the Capsule's base system as well as any clients connected to the Capsule. The repositories must be synchronized. See the Content Management Guide for more information on manifests and repositories.
- The Satellite Server's base system must be able to resolve the host name of the Capsule Server's base system and vice versa.
- You must have a Satellite Server user name and password. For more information, see the *Red Hat Satellite 6.2 Server Administration Guide*.

Register Capsule Server to Satellite Server

1. Install the 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.

```
# subscription-manager register --org organization_name
```

4.2. IDENTIFYING AND ATTACHING THE CAPSULE SERVER SUBSCRIPTION

After you have registered the Capsule Server, you need to identify your Capsule Server subscription Pool ID. The Pool ID enables you to attach the required subscription to your Capsule Server. The Capsule Server subscription provides access to the Capsule Server content, as well as Red Hat Enterprise Linux, Red Hat Software Collections (RHSCL), and Red Hat Satellite. This is the only subscription required.

1. Identify your Capsule Server subscription.

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

The command displays output similar to the following:

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

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:               10/07/2022
System Type:        Physical
```

2. Make a note of the Pool ID so that you can attach it to your Satellite host. Your Pool ID will be different than the example provided.

3. Attach your subscription to your Capsule Server, using your Pool ID:

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

The outputs displays something similar to the following:

```
Successfully attached a subscription for: Red Hat Capsule Server
```

4. To verify that the subscriptions are successfully attached, run the following command:

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

4.3. CONFIGURING REPOSITORIES

1. Disable all existing repositories.

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

2. Enable the Red Hat Satellite, Red Hat Enterprise Linux, and Red Hat Software Collections repositories.
The Red Hat Software Collections repository provides a later version of Ruby required by some Red Hat Satellite features, including the Remote Execution feature.

Ensure the Red Hat Enterprise Linux repository matches the specific version you are using.

a. If you are using Red Hat Enterprise Linux 6, run this command.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-6-server-satellite-capsule-6.2-rpms \
--enable rhel-server-rhscl-6-rpms
```

b. If you are using Red Hat Enterprise Linux 7, run this command.

```
# subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-7-server-satellite-capsule-6.2-rpms \
--enable rhel-server-rhscl-7-rpms
```

3. Clear out any metadata left from any non-Red Hat **yum** repositories.

```
# yum clean all
```

4. Verify that the repositories have been enabled.

```
# yum repolist enabled
```

The following output displays:

```
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-6.2-rpms/x86_64       Red Hat
Satellite Capsule 6.2(for RHEL 7 Server) (RPMs)    176
repolist: 7,793
```

4.4. SYNCHRONIZING TIME

You must start and enable a time synchronizer on the host operating system to minimize the effects of time drift. If a system's time is incorrect, certificate verification can fail.

Two time synchronizers are available: **NTP** and **chrony**. Each of these has its advantages. **chrony** is recommended for systems that are frequently suspended and for systems—such as mobile and virtual systems—that intermittently disconnect from networks and then reestablish network connection. **NTP** is recommended for systems that are expected to remain in running states and that are expected to be connected to a network without interruption.

For more information on the differences between **NTP** and **chrony**, see Differences Between ntpd and chronyd in the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

Synchronizing Time by Using NTP

1. Install ntp.

```
# yum install ntp
```

2. Verify that your NTP server is available.

```
# ntpdate -q ntp_server_address
```

3. Set the system time.

```
# ntpdate ntp_server_address
```

4. Start and enable the ntpd service.

```
# chkconfig ntpd on
```

Synchronizing Time by Using chronyd

1. Install chronyd.

```
# yum install chrony
```

2. Start and enable the chrony service.

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

4.5. INSTALLING CAPSULE SERVER

1. Install the installation package.

```
# yum install satellite-capsule
```

4.6. PERFORMING INITIAL CONFIGURATION OF CAPSULE SERVER

This section demonstrates a default installation of Capsule Server, including use of default certificates, DNS, and DHCP configuration. For details of more advanced configuration options, see Section 4.7, “Performing Additional Configuration on Capsule Server”.

4.6.1. Configuring Capsule Server with a Default Server Certificate

You can use the default certificate authority (CA) that comes with Capsule Server, which is used by both the server and the client SSL certificates for the authentication of subservices.

Before You Begin

- You must have attached the required subscription to the Capsule Server.
- You must have installed the **`katello-ca-consumer-latest`** package.
- You must have registered your Capsule Server to the Satellite Server.

Configure Capsule Server with a Default Server Certificate

1. Create the certificates archive on Satellite Server.

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

2. Ensure that the **`satellite-installer`** package is available on the Capsule Server.

3. Copy the generated archive .tar file from Satellite Server to Capsule Server.

4. Enable the certificate based on the needs of your environment. For more information, see **`satellite-installer --scenario capsule --help`**.

```
# satellite-installer --scenario capsule \
--capsule-parent-fqdn "satellite.example.com" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "mycapsule.example.com" \
--foreman-proxy-oauth-consumer-key
"UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV" \
--foreman-proxy-oauth-consumer-secret \
"ZhH8p7M577ttNU3WmUGWASag3JeXKgUX" \
--capsule-pulp-oauth-secret "TPk42MYZ42nAE3rZvyLBh7Lxob3nEUi8" \
--capsule-certs-tar "~/mycapsule.example.com-certs.tar"
```

4.7. PERFORMING ADDITIONAL CONFIGURATION ON CAPSULE SERVER

4.7.1. Installing the katello Agent

Installing the katello agent is recommended to allow remote updates of clients. The base system of a self-registered Satellite Server or of a Capsule Server is a client of Satellite Server and therefore should also have the katello agent installed.

Before You Begin

- You must have enabled the Satellite Tools repositories in Satellite Server.
- You must have synchronized the Satellite Tools repositories in Satellite Server.

To Install katello-agent:

1. Log into the system.
2. Enable the Satellite tools repository for this version of Satellite.
   - On Red Hat Enterprise Linux 7, run the following command:

     ```
     # subscription-manager repos \
     --enable=rhel-7-server-satellite-tools-6.2-rpms
     ```

   - On Red Hat Enterprise Linux 6, run the following command:

     ```
     # subscription-manager repos \
     --enable=rhel-6-server-satellite-tools-6.2-rpms
     ```

3. Install the package.

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

4. Ensure the **goferd** service is running:

- On Red Hat Enterprise Linux 6, enter the following command:

```
# service goferd start
```

- On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start goferd
```

4.7.2. Enabling Remote Execution on Capsule Server

If you want to run commands on a Capsule Server's hosts, you must ensure that remote execution is enabled.

NOTE

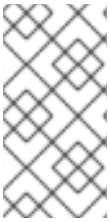
Remote execution on external Capsules is disabled by default. To use remote execution on a Capsule Server you need to enable it by running the following command:

```
# satellite-installer --scenario capsule \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4.7.3. Adding Life Cycle Environments to Capsule Servers

If your Capsule Server has content functionality enabled, you must add one or more life cycle environments to it. Adding an environment enables Capsule Server to synchronize content from Satellite Server and provide content to host systems.

Red Hat recommends that you create one or multiple life cycle environments and assign them to your Capsule Server. This ensures that Capsule receives only the repositories contained in Content Views that are promoted to the respective life cycle environments, and results in optimizing the usage of system resources.



NOTE

Avoid assigning the Library Lifecycle Environment to your Capsule Server as it triggers an automated Capsule sync every time a repository is updated from the CDN. This may consume multiple system resources on Capsules, network bandwidth between Satellite and Capsules, and available disk space on Capsules.

Capsule Server is configured using Satellite Server's Hammer CLI. You must execute all commands on Satellite Server.

1. Log in to the Satellite Server CLI as root.
2. Display a list of all Capsule Servers and note the ID.

```
# hammer capsule list
```

3. Using the ID, verify the details of your Capsule Server.

```
# hammer capsule info --id capsule_id_number
```

4. Verify the life cycle environments available and note the environment ID.

```
# hammer capsule content available-lifecycle-environments \
--id capsule_id_number
```

Available life cycle environments are available for Capsule Server, but not currently attached.

5. Add the life cycle environment to your Capsule Server.

```
# hammer capsule content add-lifecycle-environment \
--id capsule_id_number --environment-id environment_id_number
```

6. Repeat for each life cycle environment you want to add to Capsule Server.

7. To synchronize all content from your Satellite Server environment to Capsule Server, run the following command:

```
# hammer capsule content synchronize --id capsule_id_number
```

8. To synchronize a specific life cycle environment from your Satellite Server to Capsule Server, run the following command:

```
# hammer capsule content synchronize --id external_capsule_id_number
\
--environment-id environment_id_number
```

For more information on working with Life Cycle Environments, see Life Cycle Environments in the *Red Hat Satellite Server Administration Guide*.

4.7.4. Enabling Power Management 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 enables 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).

For more information on the BMC service, see Configuring an Additional Network Interface in the *Red Hat Satellite Host Configuration Guide*.

Before You Begin

- All managed hosts must have a network interface, with type **BMC**. Satellite uses this NIC to pass the appropriate credentials to the host.

Enable Power Management on Managed Hosts

1. Run the installer with the options to enable BMC.

```
# satellite-installer --scenario capsule \
--foreman-proxy-bmc "true" \
--foreman-proxy-bmc-default-provider "freeipmi"
```

4.7.5. Configuring DNS and DHCP on Capsule Server

You can configure DNS, DHCP, and TFTP on Capsule Server.

You can also configure Capsule Server to use external DNS and DHCP services. See Chapter 5, *Configuring External Services* for more information.

To view a complete list of configurable options, run the **`satellite-installer --scenario capsule --help`** command.

Before You Begin

- You must have the correct network name (**`dns-interface`**) for the DNS server.
- You must have the correct interface name (**`dhcp-interface`**) for the DHCP server.

Configure DNS, DHCP, and TFTP on Capsule Server

1. Run capsule installer with the options applicable to your environment.
   The following example shows full provisioning services:

```
# satellite-installer --scenario capsule \
--foreman-proxy-tftp=true \
--foreman-proxy-foreman-oauth-key "your_organization_key" \
--foreman-proxy-foreman-oauth-secret "your_organization_secret" \
--capsule-certs-tar "~/capsule.example.com-certs.tar" \
--foreman-proxy-templates=true \
--foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-gateway=192.168.122.1 \
--foreman-proxy-dhcp-nameservers=192.168.122.1 \
--foreman-proxy-dhcp-range="192.168.122.100 192.168.122.200" \
--foreman-proxy-dhcp-interface=eth0 \
--foreman-proxy-dns=true \
--foreman-proxy-dns-forwarders=8.8.8.8 \
--foreman-proxy-dns-interface=eth0 \
--foreman-proxy-dns-zone=example.com

# satellite-installer --scenario capsule \
--foreman-proxy-dns true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
```

```
--foreman-proxy-tftp-servername $(hostname) \
--capsule-puppet true \
--foreman-proxy-puppetca true
```

4.7.6. Configuring Capsule Server with a Custom Server Certificate

Red Hat Satellite 6 comes with default SSL certificate to enable encrypted communications between the Satellite Server, Capsule Servers, and all hosts. You can replace the default certificate with a custom certificates if required. For example, your company's security policy might dictate that SSL certificates must be obtained from a specific Certificate Authority.

Prerequisites

- Satellite Server configured with a custom certificate. For more information, see Section 3.5.8, "Configuring Satellite Server with a Custom Server Certificate".
- Capsule Server installed and registered to the Satellite Server. For more information, see Chapter 4, *Installing Capsule Server*.

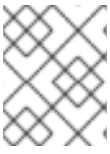
To use a custom certificate on each Capsule Server, complete these procedures:

1. Section 4.7.6.1, "Obtain an SSL Certificate for the Capsule Server"
2. Section 4.7.6.2, "Validate the Capsule Server's SSL Certificate"
3. Section 4.7.6.3, "Create the Capsule Server's Certificate Archive File"
4. Section 4.7.6.4, "Install the Capsule Server's Custom Certificate"
5. Section 4.7.6.5, "Install the Capsule Server's New Certificate on All Hosts"

4.7.6.1. Obtain an SSL Certificate for the Capsule Server

IMPORTANT

This procedure generates PEM encoded certificates. Only PEM encoding must be used for the SSL Certificates.



NOTE

- Do **not** use the Satellite Server's certificate on any Capsule Server as each server's certificate is unique.

1. Create a directory to contain all the source certificate files, accessible to only the **`root`** user.

```
# mkdir /root/capsule_cert
# cd /root/capsule_cert
```

In these examples, the directory is **`/root/capsule_cert`**. If you have multiple Capsule Servers, name the directory to match. For example, if you have Capsule Servers named **`capsule_apac`** and **`capsule_emea`**, you might create directories named *capsule_apac* and *capsule_emea* respectively. This is not *required*, but reduces the risk of using files from one Capsule Server on another Capsule Server.

2. Create a private key with which to sign the Certificate Signing Request (CSR).

NOTE

If you already have a private key for the Capsule Server, skip this step.

```
# openssl genrsa -out /root/capsule_cert/capsule_cert_key.pem 4096
```

3. Create a Certificate Signing Request (CSR).
A Certificate Signing Request is a text file containing details of the server for which you are requesting a certificate. For this command, you provide the private key (output by the previous step), answer some questions about the Capsule Server, and the Certificate Signing Request is stored in a file.

NOTE

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

To confirm a server's FQDN, run the command **`hostname -f`** on the server.

```
# openssl req -new \
  -key /root/capsule_cert/capsule_cert_key.pem \ 1
  -out /root/capsule_cert/capsule_cert_csr.pem   2
```

1. Capsule Server's private key, used to sign the certificate
2. Certificate Signing Request file

Example Certificate Signing Request session

```
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:Queensland
Locality Name (eg, city) [Default City]:Brisbane
Organization Name (eg, company) [Default Company Ltd]:Example
Organizational Unit Name (eg, section) []:Sales
Common Name (eg, your name or your server's hostname)
[]:capsule.example.com
Email Address []:example@example.com

Please enter the following 'extra' attributes
```

```
to be sent with your certificate request
A challenge password []:password
An optional company name []:Example
```

4. Send the certificate signing request to the Certificate Authority.
When you submit the request, be sure to specify the lifespan of the certificate. The method for sending the certificate signing request varies, so consult the Certificate Authority for the preferred method. In response to the request you can expect to receive a Certificate Authority bundle, and a signed certificate, in separate files.

4.7.6.2. Validate the Capsule Server's SSL Certificate

On the Satellite Server, validate the Capsule Server's certificate input files with the **katello-certs-check** command. This process requires that you have copied the Capsule Server key, CSR, and SSL certificate from Capsule Server to Satellite Server.

```
# katello-certs-check \
   -c /root/capsule_cert/capsule_cert.pem \        1
   -k /root/capsule_cert/capsule_cert_key.pem \    2
   -r /root/capsule_cert/capsule_cert_csr.pem \    3
   -b /root/capsule_cert/ca_cert_bundle.pem        4
```

1. Capsule Server certificate file, provided by your Certificate Authority
2. Capsule Server's private key, used to sign the certificate
3. Capsule Server's certificate signing request file
4. Certificate Authority bundle, provided by your Certificate Authority

If the certificate is successfully validated, the output will contain the following.

```
Check private key matches the certificate: [OK]
Check ca bundle verifies the cert file: [OK]
```

Proceed to Section 4.7.6.3, "Create the Capsule Server's Certificate Archive File".

4.7.6.3. Create the Capsule Server's Certificate Archive File

The Capsule Server's installer requires the server's certificate be provided in an archive file. To create this file, use the **capsule-certs-generate** command on the Satellite Server.

The **capsule-certs-generate** command must be run once for every external Capsule Server. In these examples, **capsule.example.com** is the example FQDN and **capsule_certs.tar** the example archive file's name. Replace these with values appropriate to your environment, taking care not to overwrite an existing certificate archive file. For example, if you have Capsule Servers named **capsule1** and **capsule2**, you could name the certificate archive files **capsule1_certs.tar** and **capsule2_certs.tar**.

1. Copy and paste into a terminal the **capsule-certs-generate** command, as output by the **katello-certs-check** command in Section 3.5.8.2, "Validate the Satellite Server's SSL Certificate".

2. Edit the values for **--capsule-fqdn** to match the Capsule Server's FQDN, and **--certs-tar** to the file path and name for the certificate archive file.

3. If the Capsule Server has not already been installed, remove the **--certs-update-server** parameter. This is used only to **update** an existing Capsule Server's certificate.

4. On the Satellite Server, run the resulting command.

   Example `capsule-certs-generate` command

   ```
   # capsule-certs-generate --capsule-fqdn "capsule.example.com" \
   --certs-tar  "/root/capsule_cert/capsule_certs.tar" \
   --server-cert "/root/capsule_cert/capsule_cert.pem" \
   --server-cert-req "/root/capsule_cert/capsule_cert_csr.pem" \
   --server-key "/root/capsule_cert/capsule_cert_key.pem" \
   --server-ca-cert "/root/sat_cert/ca_cert_bundle.pem" \
   --certs-update-server
   ```

5. On the Satellite Server, copy the certificate archive file to the Capsule Server, providing the **root** user's password when prompted.
   In this example the archive file is copied to the **root** user's home directory, but you may prefer to copy it elsewhere.

   ```
   # scp /root/capsule_cert/capsule_certs.tar root@capsule.example.com:
   ```

Proceed to Section 4.7.6.4, "Install the Capsule Server's Custom Certificate".

4.7.6.4. Install the Capsule Server's Custom Certificate

WARNING

Complete this procedure on the Capsule Server.

To install the Capsule Server's custom certificates, run the Satellite installer. The command, including parameters, is output by the the **capsule-certs-generate** command in Section 4.7.6.3, "Create the Capsule Server's Certificate Archive File".

1. Copy and paste the custom **capsule-certs-generate** command but do **NOT** run it.

2. Edit the value for **--capsule-certs-tar** to match the location of the certificates archive file.

3. If you want to enable additional features on the Capsule Server, append their parameters to the **satellite-installer** command. For a description of all the installer's parameters, run the command **satellite-installer --scenario capsule --help**.

Example custom `satellite-installer` command

```
# satellite-installer --scenario capsule \
```

```
--capsule-parent-fqdn "satellite.example.com" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com" \
--foreman-proxy-oauth-consumer-key "FeQsbASvCjvvaqE6duKH6SoYZWg4jwjg" \
--foreman-proxy-oauth-consumer-secret "7UhPXFPDBongvdTbNixbsWR5WFZsKEgF" \
--capsule-pulp-oauth-secret "VpQ9587tVmYeuY4Du6VitmZpZE5vy9ac" \
--capsule-certs-tar "/root/capsule_certs.tar"
```

NOTE

The **satellite-installer** command, as output by the **capsule-certs-generate** command, is unique to each Capsule Server. Do **not** use the same command on more than one Capsule Server.

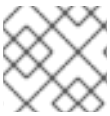
Do **NOT** delete the certificates archive file (the .tar file) even after the certificates have been deployed to all relevant hosts. It is required, for example, when upgrading the Capsule Server. If the certificates archive file is not found by the installer, it will fail with a message similar to the following:

```
[ERROR YYYY-MM-DD hh:mm:ss main]  tar -xzf
/var/tmp/srvcapsule01.tar returned 2 instead of one of [0]
```

Proceed to Section 4.7.6.5, “Install the Capsule Server’s New Certificate on All Hosts”.

4.7.6.5. Install the Capsule Server’s New Certificate on All Hosts

Hosts which connect to an external Capsule Server require that server’s custom certificate. Run the following command on all the Capsule Server’s hosts.



NOTE

Use the Capsule Server’s host name, **not** that of the Satellite Server.

```
# yum -y localinstall \
http://capsule.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

4.7.7. Restricting Access to mongod

Only the **apache** and **root** users should be allowed access to the MongoDB database daemon, **mongod**, to reduce the risk of data loss.

Restrict access to **mongod** on Satellite and Capsule Servers using the following commands.

Configuring the Firewall on Red Hat Enterprise Linux 6

1. Configure **iptables** service on Satellite and Capsule Servers.

```
# iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner \
```

```
--uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -j DROP \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -j DROP \
&& service iptables save
```

Configuring the Firewall on Red Hat Enterprise Linux 7

1. Configure the firewall on Satellite and Capsule Servers.

```
# firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP
```

2. Repeat the command adding the **`--permanent`** option to make the settings persistent.

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 27017 -m owner \
```

```
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner apache -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
\
-o lo -p tcp -m tcp --dport 28017 -m owner \
--uid-owner root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
\
-o lo -p tcp -m tcp --dport 28017 -j DROP
```

CHAPTER 5. CONFIGURING EXTERNAL SERVICES

Some environments have existing DNS, DHCP, and TFTP services and do not need to use the Satellite Server to provide these services. If you want to use external servers to provide DNS, DHCP, or TFTP, you can configure them for use with Satellite Server.

If you want to disable these services in Satellite in order to manage them manually, see Section 3.5.6, “Disabling DNS, DHCP, and TFTP for Unmanaged Networks” for more information.

5.1. CONFIGURING SATELLITE WITH EXTERNAL DNS

You can configure Satellite to use an external server to provide DNS service.

1. Deploy a Red Hat Enterprise Linux Server and install the ISC DNS Service.

```
# yum install bind bind-utils
```

2. Create the configuration for the domain.
   The following example 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).

```
# 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;
    };
};
```

   The **`inet`** line must be entered as one line in the configuration file.

3. Create a key file.

```
# ddns-confgen -k capsule
```

This command can take a long time to complete.

4. Copy and paste the output from the key section into a separate file called **/etc/rndc.key**.

```
# cat /etc/rndc.key
key "capsule" {
        algorithm hmac-sha256;
        secret "GeBbgGoLedEAAwNQPtPh3zP56MJbkwM84UJDtaUS9mw=";
};
```

IMPORTANT

This is the key used to change DNS server configuration. Only the root user should read and write to it.

5. Create zone files.

```
# 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
```

6. 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.
```

There should be no extra non-ASCII characters.

5.2. VERIFYING AND STARTING THE DNS SERVICE

1. Validate the syntax.

```
# named-checkconf -z /etc/named.conf
```

2. Start the server.

   a. If you are using Red Hat Enterprise Linux 6, run this command.

```
# service named restart
```

   b. If you are using Red Hat Enterprise Linux 7, run this command.

```
# systemctl restart named
```

3. Add a new host.
   The following uses the example host 192.168.38.2. You should change this to suit your environment.

```
# 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.

```
# nslookup aaa.virtual.lan 192.168.38.2
```

5. If necessary, 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).

   - For Satellite Server running Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
&& firewall-cmd --permanent --add-port="53/udp" --add-
port="53/tcp"
```

   - For Satellite Server running Red Hat Enterprise Linux 6:

```
# iptables -I INPUT -m state --state NEW -p udp \
--dport 53 -j ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp \
```

```
--dport 53 -j ACCEPT \
&& service iptables save
```

Verify that the iptables service is started and enabled.

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

5.3. CONFIGURING CAPSULE SERVER WITH EXTERNAL DNS

1. On the Red Hat Enterprise Linux Server, install the ISC DNS Service.

```
# yum install bind bind-utils
```

Ensure that the **nsupdate** utility was installed. The Capsule uses the **nsupdate** utility to update DNS records on the remote server.

2. Copy the **/etc/rndc.key** file from the services server to the Capsule Server.

```
# scp localfile username@hostname:remotefile
```

3. Verify that the key file has the correct owner, permissions, and SELinux label.

```
# ls /etc/rndc.key -Zla
-rw-r-----. root named system_u:object_r:dnssec_t:s0
/etc/rndc.key
```

4. Test the **nsupdate** utility by adding a host remotely.

```
# 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 **satellite-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/dns.yml** file.

```
# satellite-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 the foreman-proxy service.

   a. If you are using Red Hat Enterprise Linux 6, run this command.

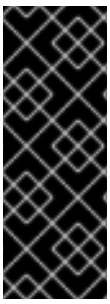
```
# service foreman-proxy restart
```

b. If you are using Red Hat Enterprise Linux 7, run this command.

```
# systemctl restart foreman-proxy
```

7. Log in to the Satellite Server web UI.
8. Go to **Infrastructure** > **Capsules**. Locate the appropriate Capsule Server and from the **Actions** drop-down list, select **Refresh**. The DNS feature should appear.
9. Associate the DNS service with the appropriate subnets and domain.

5.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP



IMPORTANT

From Satellite 6.3 onwards, the foreman-proxy DHCP **`isc`** provider does not support remote DHCP lease files. You must follow the procedures in the Satellite 6.3 Installation guide to change to the new remote ISC DHCP provider **`remote_isc`** when you upgrade to Satellite 6.3. For more information about using **`remote_isc`** in Satellite 6.3, see Configuring Satellite Server with External DHCP in the *Red Hat Satellite 6.3 Installation Guide*.

1. Deploy a Red Hat Enterprise Linux Server and install the ISC DHCP Service.

```
# yum install dhcp
```

2. Generate a security token in an empty directory.

```
# 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.

3. Copy the secret hash from the key.

```
# cat Komapi_key.+*.private |grep ^Key|cut -d ' ' -f2
```

4. Edit the **dhcpd** configuration file for all of the subnets and add the key.

```
# 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 "jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

5. Delete the two key files from the directory where you created them.

6. Define each subnet on the Satellite Server.
   It is recommended to set up a lease range and reservation range separately to prevent conflicts. For example, the lease range is 192.168.38.10 to 192.168.38.100 so the reservation range (defined in the Satellite web UI) is 192.168.38.101 to 192.168.38.250. Do not set DHCP Capsule for the defined Subnet yet.

   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.

7. Configure the firewall for external access to the DHCP server.

   - For Satellite Server running Red Hat Enterprise Linux 7:

     ```
     # firewall-cmd --add-service dhcp \
     && firewall-cmd --permanent --add-service dhcp
     ```

   - For Satellite Server running Red Hat Enterprise Linux 6:

     ```
     # iptables -I INPUT -m state --state NEW -p tcp \
     --dport 67 -j ACCEPT \
     && service iptables save
     ```

     Verify that the iptables service is started and enabled.

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

8. Determine the UID and GID numbers of the foreman-proxy user on the Capsule Server. Create the same user and group with the same IDs on the DHCP server.

   ```
   # groupadd -g 990 foreman-proxy
   # useradd -u 992 -g 990 -s /sbin/nologin foreman-proxy
   ```

9. To make the configuration files readable, restore the read and execute flags.

   ```
   # chmod o+rx /etc/dhcp/
   # chmod o+r /etc/dhcp/dhcpd.conf
   # chattr +i /etc/dhcp/ /etc/dhcp/dhcpd.conf
   ```

10. Start the DHCP service.

    a. If you are using Red Hat Enterprise Linux 6, run this command.

a. If you are using Red Hat Enterprise Linux 6, run this command.

```
# service dhcpd start
```

b. If you are using Red Hat Enterprise Linux 7, run this command.

```
# systemctl start dhcpd
```

11. Export the DHCP configuration and leases files using NFS.

```
# yum install nfs-utils
# systemctl enable rpcbind nfs-server
# systemctl start rpcbind nfs-server nfs-lock nfs-idmapd
```

12. Create the DHCP configuration and leases files to be exported using NFS.

```
# mkdir -p /exports/var/lib/dhcpd /exports/etc/dhcp
```

13. 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
```

14. Mount the file systems in /etc/fstab.

```
# mount -a
```

15. 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)
```

16. Reload the NFS server.

```
# exportfs -rva
```

17. Configure the firewall for the DHCP omapi port 7911 for the Capsule Server.

- On Red Hat Enterprise Linux 7, run the following command:

```
# firewall-cmd --add-port="7911/tcp" \
&& firewall-cmd --permanent --add-port="7911/tcp"
```

- On Red Hat Enterprise Linux 6, run the following commands:

```
# iptables -I INPUT -m state --state NEW -p tcp \
--dport 7911 -j ACCEPT \
&& service iptables save
```

Ensure that the iptables service is started and enabled.

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

18. If required, configure the firewall for external access to NFS.
Clients are configured using NFSv3.

- On Red Hat Enterprise Linux 7, use the **firewalld** daemon's NFS service to configure the firewall.

```
# 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
```

- On Red Hat Enterprise Linux 6, configure the ports for NFSv3 in the **/etc/sysconfig/nfs** file.

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Restart the service.

```
# service nfs restart
```

Add rules to the **/etc/sysconfig/iptables** file.

```
# iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p udp
\
--dport 111 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 111 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp \
--dport 2049 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 2049 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 32803 -j ACCEPT \
```

```
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp \
--dport 32769 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp \
--dport 892 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 892 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp \
--dport 875 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 875 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp \
--dport 662 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp \
--dport 662 -j ACCEPT \
&& service iptables save
```

Restart the firewall.

```
# service iptables restart
```

For more information on using NFSv3 behind a firewall on Red Hat Enterprise Linux 6, see Running NFS Behind a Firewall in the *Red Hat Enterprise Linux 6 Storage Administration Guide*.

5.5. CONFIGURING CAPSULE SERVER WITH EXTERNAL DHCP

1. Install the NFS client.

```
# yum install nfs-utils
```

2. Create the DHCP directories for NFS.

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

3. Change the file owner.

```
# chown -R foreman-proxy /mnt/nfs
```

4. Verify communication with the NFS server and RPC communication paths.

```
# showmount -e 192.168.38.2
# rpcinfo -p 192.168.38.2
```

5. Add the following 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 on **/etc/fstab**.

```
# mount -a
```

7. 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
```

8. Run the **satellite-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/dhcp.yml** file.

```
# satellite-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=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-dhcp-server dhcp.example.com
```

9. Restart the foreman-proxy service.

   a. If you are using Red Hat Enterprise Linux 6, run this command.

   ```
   # service foreman-proxy restart
   ```

   b. If you are using Red Hat Enterprise Linux 7, run this command.

   ```
   # systemctl restart foreman-proxy
   ```

10. Log in to the Satellite Server web UI.

11. Go to **Infrastructure** > **Capsules**. Locate the appropriate Capsule Server and from the **Actions** drop-down list, select **Refresh**. The DHCP feature should appear.

12. Associate the DHCP service with the appropriate subnets and domain.

5.6. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

Before You Begin

- You should have already configured NFS and the firewall for external access to NFS. See Section 5.4, “Configuring Satellite Server with External DHCP”.

Configure Satellite Server with External TFTP

1. Install and enable the TFTP server.

```
# yum install tftp-server syslinux
```

   a. On Red Hat Enterprise 7, enable and activate the **tftp.socket** unit.

```
# systemctl enable tftp.socket
# systemctl start tftp.socket
```

   b. On Red Hat Enterprise Linux 6, enable and start the **xinetd** service.

```
# service xinetd enable
# service xinetd start
```

2. Configure the PXELinux environment.

```
# 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
```

5.6.1. Configuring the Firewall for External Access to TFTP

Configuring the Firewall for External Access to the TFTP Service Using Red Hat Enterprise Linux 7

1. Configure the firewall (UDP on port 69).

```
# firewall-cmd --add-port="69/udp" \
&& firewall-cmd --permanent --add-port="69/udp"
```

Configuring the Firewall for External Access to the TFTP Service Using Red Hat Enterprise Linux 6

1. Configure the firewall.

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 69 -j ACCEPT
\
&& service iptables save
```

2. Ensure the iptables service is started and enabled.

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

5.7. CONFIGURING CAPSULE SERVER WITH EXTERNAL TFTP

1. Create the TFTP directory to prepare for NFS.

```
# mkdir -p /mnt/nfs/var/lib/tftpboot
```

2. Add the following line in the **/etc/fstab** file:

```
192.168.38.2:/exports/var/lib/tftpboot /mnt/nfs/var/lib/tftpboot nfs
rw,vers=3,auto,nosharecache,context="system_u:object_r:tftpdir_rw_t:
s0" 0 0
```

3. Mount the file systems in **/etc/fstab**.

```
# mount -a
```

4. Run the **satellite-installer** script to make the following persistent changes to the **/etc/foreman-proxy/settings.d/tftp.yml** file.

```
# satellite-installer --foreman-proxy-tftp=true \
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot
```

5. 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.

```
# satellite-installer --foreman-proxy-tftp-servername=new_FQDN
```

This updates all configuration files with the new value.

6. Log in to the Satellite Server web UI.
7. Go to **Infrastructure** > **Capsules**. Locate the appropriate Capsule Server and from the **Actions** drop-down list, select **Refresh**. The TFTP feature should appear.
8. Associate the TFTP service with the appropriate subnets and domain.

5.8. CONFIGURING SATELLITE WITH EXTERNAL IDM DNS

Red Hat Satellite can be configured to use a Red Hat Identity Management (IdM) server to provide the DNS service. Two methods are described here to achieve this, both using a transaction key. For more information on Red Hat Identity Management, see the Linux Domain Identity, Authentication, and Policy Guide.

The first method is to install the IdM client which will handle the process automatically using the *generic security service algorithm for secret key transaction* (GSS-TSIG) technology defined in RFC3645. This method requires installing the IdM client on the Satellite Server or Capsule's base system and having an account created by the IdM server administrator for use by the Satellite administrator. See Section 5.8.1, "Configuring Dynamic DNS Update with GSS-TSIG Authentication" to use this method.

The second method, *secret key transaction authentication for DNS* (TSIG), uses an **`rndc.key`** for authentication. It requires root access to the IdM server to edit the BIND configuration file, installing the **`BIND`** utility on the Satellite Server's base system, and coping the **`rndc.key`** to between the systems. This technology is defined in RFC2845. See Section 5.8.2, "Configuring Dynamic DNS Update with TSIG Authentication" to use this method.

NOTE

You are not required to use Satellite to manage DNS. If you are already using the Realm enrollment feature of Satellite, where provisioned hosts are enrolled automatically to IdM, then the **`ipa-client-install`** script will create DNS records for the client. The following procedure and Realm enrollment are therefore mutually exclusive. See External Authentication for Provisioned Hosts in the *Server Administration Guide* for more information on configuring Realm enrollment.

Determining where to install the IdM Client

When Satellite Server wants to add a DNS record for a host, it first determines which Capsule is providing DNS for that domain. It then communicates with the Capsule and adds the record. The hosts themselves are not involved in this process. This means you should install and configure the IdM client on the Satellite or Capsule that is currently configured to provide a DNS service for the domain you want to manage using the IdM server.

5.8.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication

In this example, Satellite Server has the following settings.

Host name	**satellite.example.com**
Network	**192.168.55.0/24**

The IdM server has the following settings.

Host name	**idm1.example.com**
Domain name	**example.com**

Before you Begin.

1. Confirm the IdM server is deployed and the host-based firewall has been configured correctly. See Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide* for more information.
2. Obtain an account on the IdM server with permissions to create zones on the IdM server.
3. Confirm if the Satellite or an external Capsule is managing DNS for a domain.
4. Confirm that the Satellite or external Capsule are currently working as expected.
5. In the case of a newly installed system, complete the installation procedures in this guide first. In particular, DNS and DHCP configuration should have been completed.
6. Optionally, make a backup of the answer file. This can make it easier to revert to using the internal DNS service. See Section 3.3.4, “Configuring Red Hat Satellite with an Answer File” for more information.

Create a Kerberos Principal on the IdM Server.

1. Ensure you have a Kerberos ticket.

```
# kinit idm_user
```

   Where *idm_user* is the account created for you by the IdM administrator.

2. Create a new Kerberos principal for the Satellite or Capsule to use to authenticate to the IdM server.

```
# ipa service-add capsule/satellite.example.com
```

Install and Configure the IdM Client.

Do this on the Satellite or Capsule Server that is managing the DNS service for a domain.

1. Install the IdM client package.

```
# yum install ipa-client
```

2. Configure the IdM client by running the installation script and following the on-screen prompts.

```
# ipa-client-install
```

3. Ensure you have a Kerberos ticket.

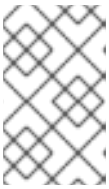
```
# kinit admin
```

4. Remove any preexisting keytab.

```
# rm /etc/foreman-proxy/dns.keytab
```

5. Get the keytab created for this system.

```
# ipa-getkeytab -p capsule/satellite.example.com@EXAMPLE.COM \
-s idm1.example.com -k /etc/foreman-proxy/dns.keytab
```



NOTE

When adding a keytab to a standby system with the same host name as the original system in service, add the **r** option to prevent generating new credentials and rendering the credentials on the original system invalid.

6. Set the group and owner for the keytab file to **foreman-proxy** as follows.

```
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/dns.keytab
```

7. If required, check the keytab is valid.

```
# kinit -kt /etc/foreman-proxy/dns.keytab \
capsule/satellite.example.com@EXAMPLE.COM
```

Configure DNS Zones in the IdM web UI.

1. Create and configure the zone to be managed:
   a. Navigate to **Network Services** > **DNS** > **DNS Zones**.
   b. Select **Add** and enter the zone name. In this example, **example.com**.
   c. Click **Add and Edit**.
   d. On the Settings tab, in the **BIND update policy** box, add an entry as follows to the semi-colon separated list.

```
grant capsule\047satellite.example.com@EXAMPLE.COM wildcard *
ANY;
```

   e. Ensure **Dynamic update** is set to **True**.
   f. Enable **Allow PTR sync**.
   g. Select **Save** to save the changes.

2. Create and Configure the reverse zone.

    a. Navigate to **Network Services** > **DNS** > **DNS Zones**.

    b. Select **Add**.

    c. Select **Reverse zone IP network** and add the network address in CIDR format to enable reverse lookups.

    d. Click **Add and Edit**.

    e. On the **Settings** tab, in the **BIND update policy** box, add an entry as follows to the semi-colon separated list:

    ```
    grant capsule\047satellite.example.com@EXAMPLE.COM wildcard *
    ANY;
    ```

    f. Ensure **Dynamic update** is set to **True**.

    g. Select **Save** to save the changes.

Run the Installation Script on the Satellite or Capsule Server that is Managing the DNS Service for the Domain.

- On a Satellite Server's Base System.

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-
principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

- On a Capsule Server's Base System.

```
satellite-installer --scenario capsule \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-
principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

Restart the Satellite or Capsule's Proxy Service.

- On Red Hat Enterprise Linux 7.

```
# systemctl restart foreman-proxy
```

- On Red Hat Enterprise Linux 6.

```
# service foreman-proxy restart
```

Update the Configuration in Satellite web UI.

After you have run the installation script to make any changes to a Capsule, instruct Satellite to scan the configuration on each affected Capsule as follows:

1. Navigate to **Infrastructure** > **Capsules**.
2. For each Capsule to be updated, from the **Actions** drop-down menu, select **Refresh**.
3. Configure the domain:
   a. Go to **Infrastructure** > **Domains** and select the domain name.
   b. On the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.
4. Configure the subnet:
   a. Go to **Infrastructure** > **Subnets** and select the subnet name.
   b. On the **Subnet** tab, set **IPAM** to **None**.
   c. On the **Domains** tab, ensure the domain to be managed by the IdM server is selected.
   d. On the **Capsules** tab, ensure **Reverse DNS Capsule** is set to the Capsule where the subnet is connected.
   e. Click **Submit** to save the changes.

5.8.2. Configuring Dynamic DNS Update with TSIG Authentication

In this example, Satellite Server has the following settings.

IP address	**192.168.25.1**
Host name	**satellite.example.com**

The IdM server has the following settings.

Host name	**idm1.example.com**
IP address	**192.168.25.2**
Domain name	**example.com**

Before you Begin

1. Confirm the IdM Server is deployed and the host-based firewall has been configured correctly. See Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide* for more information.

2. Obtain **root** user privileges on the IdM server.

3. Confirm if the Satellite or an external Capsule is managing DNS for a domain.

4. Confirm that the Satellite or external Capsule are currently working as expected.

5. In the case of a newly installed system, complete the installation procedures in this guide first. In particular, DNS and DHCP configuration should have been completed.

6. Optionally, make a backup of the answer file. This can make it easier to revert to using the internal DNS service. See Section 3.3.4, “Configuring Red Hat Satellite with an Answer File” for more information.

Enabling External Updates to the DNS Zone in the IdM Server

1. On the IdM Server, add the following to the top of the **/etc/named.conf** file.

```
// This was added to allow Satellite Server at 192.168.25.1 to make
DNS updates.
############################################################################
####
include "/etc/rndc.key";
controls  {
inet 192.168.25.2 port 953 allow { 192.168.25.1; } keys { "rndc-
key"; };
};
############################################################################
####
```

2. In the IdM web UI, go to **Network Services** > **DNS** > **DNS Zones**. Select the name of the zone. On the **Settings** tab:

   a. Add the following in the **BIND update policy** box.

   ```
   grant "rndc-key" zonesub ANY;
   ```

   b. Ensure **Dynamic update** is set to **True**.

   c. Click **Update** to save the changes.

3. Copy the **/etc/rndc.key** file from the IdM server to a secure location for later use. Alternatively, copy it directly to Satellite's base system as follows.

```
# scp /etc/rndc.key root@satellite.example.com:/etc/rndc.key
```

4. On Satellite Server, run the installation script as follows to use the external DNS server.

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
```

```
--foreman-proxy-dns-server="192.168.25.2" \
--foreman-proxy-keyfile=/etc/rndc.key \
--foreman-proxy-dns-ttl=86400
```

Testing External Updates to the DNS Zone in the IdM Server

1. Install **bind-utils** for testing with **nsupdate**.

```
# yum install bind-utils
```

2. Ensure the key in the **/etc/rndc.key** file on Satellite Server is the same one as used on the IdM server.

```
key "rndc-key" {
        algorithm hmac-md5;
        secret "secret-key==";
};
```

3. On Satellite Server, create a test DNS entry for a host. For example, host ***test.example.com*** with an A record of **192.168.25.20** on the IdM server at **192.168.25.1**.

```
# echo -e "server 192.168.25.1\n \
update add test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. On Satellite Server, test the DNS entry.

```
# nslookup test.example.com 192.168.25.1
Server:  192.168.25.1
Address: 192.168.25.1#53

Name: test.example.com
Address: 192.168.25.20
```

5. To view the entry in the IdM web UI, go to **Network Services** > **DNS** > **DNS Zones**. Select the name of the zone and search for the host by name.

6. If resolved successfully, remove the test DNS entry.

```
# echo -e "server 192.168.25.1\n \
update delete test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

7. Confirm that the DNS entry was removed.

```
# nslookup test.example.com 192.168.25.1
```

The above **nslookup** command will fail and output the SERVFAIL error message if the record was successfully deleted.

CHAPTER 6. UPGRADING SATELLITE SERVER AND CAPSULE SERVER

Upgrading is the process of migrating your Satellite and Capsule Server installations from a y-stream release to the next, for example Satellite 6.1 to Satellite 6.2; while updating is the process of migrating your Satellite and Capsule Server installations from a z-stream release to the next, for example Satellite 6.2.7 to Satellite 6.2.8. This section covers upgrading instructions. For more information on updating, see Chapter 7, *Updating Satellite Server, Capsule Server, and Content Hosts*. Upgrades are usually done in order to take advantage of significant new features or capabilities. Because upgrades can involve deprecation of installed code, they can require a significant amount of time. You should plan your workflow to avoid disruption to your operating environment when performing upgrades. Before upgrading, check the Red Hat Satellite Release Notes for potential conflicts.

The Satellite Server and Capsule Servers are upgraded independently. Upgrade the Satellite Server first, and then upgrade any Capsule Servers. Satellite 6.1 Capsule Servers are not compatible with Satellite 6.2, 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 Satellite Server and Capsule Servers. See Section 6.7, “Upgrading Satellite Clients” for more information.

IMPORTANT

The Red Hat Satellite Server and Capsule Server y-stream versions must match. For example, a 6.1 Satellite Server cannot run a 6.2 Capsule Server and a 6.2 Satellite Server cannot run a 6.1 Capsule Server. Mismatching Satellite Server and Capsule Server versions will result in the Capsule Server failing silently. However, a Capsule Server using one z-stream version older than the Satellite Server is supported. For example, a Satellite 6.2.8 Satellite Server can run a 6.2.7 Capsule Server.

Supported upgrade paths for Satellite 6.2:

- Satellite 6.0.X GA to 6.1.X GA.
- Satellite 6.1.9 GA or later to 6.2.X GA or later.

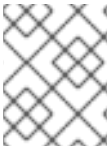
You must perform the upgrade from each version to the next. Upgrading from the Beta to GA version is not supported.

Storage Requirements for Satellite 6.2

The storage requirements for Satellite have changed from the previous version. Before upgrading, review the storage requirements as detailed in Section 2.2, “Storage Requirements and Recommendations” and ensure that the requirements are met before attempting an upgrade.

I/O Speed Requirements for Satellite 6.2

Upgrading from Satellite or Capsule 6.1 to 6.2 is a lengthy operation that can take many hours depending on the size of the repositories on the Satellite or Capsule. The upgrade processing rate for the content of the **`/var/lib/pulp/`** directory is estimated at between 50 GB and 100 GB per hour. If you have 500 GB of content, the upgrade could take from 5 to 10 hours.



NOTE

If the Pulp directory is stored on an NFS device, the upgrade takes significantly more time. For example, testing with 600 GB of Pulp storage took 30 hours to migrate over NFS.

To check the size of the **/var/lib/pulp/** directory, enter the following command:

```
# df -h /var/lib/pulp
```

Checking I/O speed is important because it affects the duration of the upgrade. Testing with the **hdparm** tool, and examining the reported **timing buffered disk reads** element, can identify a potential performance problem. You need to check where the location of the **/var/lib/pulp/** directory is mounted, and with this information you are then able to determine the I/O speed.

1. First of all, install **hdparm** in order to be able to measure I/O speed.

   ```
   # yum install hdparm
   ```

2. Display information about the **/var** directory.

   ```
   # df /var
   ```

   Examine the output and identify the logical device used by the **var** directory. In this example, it is **/dev/mapper/rhel_vm37–118-root**.

   ```
   Filesystem                     1K-blocks     Used Available Use%
   Mounted on
   /dev/mapper/rhel_vm37--118-root 200303044 41739160 158563884  21% /
   ```

3. Display information about the logical volume.

   ```
   # lvs -a -o +devices
   ```

   Examine the output and identify the device associated with the logical volume. In this example, it is the **root** device, which located at **/dev/vda2**.

   ```
   LV   VG            Attr       LSize   Pool Origin Data%  Meta%
   Move Log Cpy%Sync Convert Devices
   root rhel_vm37-118 -wi-ao---- 191.12g
   /dev/vda2(0)
   swap rhel_vm37-118 -wi-ao----   7.88g
   /dev/vda2(48926)
   ```

4. Measure the I/O speed to the relevant device. Use the location of the device that you identified in the previous step. In this example, it is **/dev/vda2**. The result is reported under the **timing buffered disk reads** element.

   ```
   # hdparm -tT /dev/vda2
   ```

   Remember **/dev/vda2** will vary depending on where the **/var/lib/pulp/** directory is mounted on your system, which is why it is vital to correctly identify the location by following the above steps.

A minimum of 80 MB/s **`Buffered Read Time`** is recommended for read access to the device used in the **`/var`** partition. Throughput lower than this can cause extreme performance issues with your Satellite installation, upgrade, and day-to-day operations. In extreme situations, slow I/O (in the order of 10–20 MB/s) to the **`/var`** partition has caused Satellite 6.1 to 6.2 upgrades to take in excess of 24 hours.

Following the Progress of the Upgrade

Due to the lengthy upgrade time, consider using a utility such as **`screen`** that allows suspending and reattaching a communication session so that you can check the upgrade progress without having to stay connected to the command shell continuously. The Red Hat Knowledgebase article How do I use the screen command? describes installing **`screen`**; alternately see the **`screen`** manual page for more information. If you lose connection to the shell where the installation command is running you can see the logs in **`/var/log/foreman-installer/satellite.log`** to determine if the process completed successfully.

Foreman Hooks' Report Class Renamed to ConfigReport

In Satellite 6.2 the Report class has been changed to ConfigReport. This means the hooks triggered by Rails events no longer look for a script stored in the **`/hooks/report/`** directory. Upgrading requires removing all hooks until the upgrade has completed successfully. After completing the upgrade and verifying that Satellite is working as expected, restore the Foreman hooks. Create a directory **`/usr/share/foreman/config/hooks/config_report/`** and move hooks such as **`after_create`** and **`before_create`** to the new directory. Rails events and Foreman hooks are described in Using Foreman Hooks in the *Red Hat Satellite Server Administration Guide*.

Docker Support in Satellite 6.2

Satellite 6.2 has upgraded Docker support from version 1 to version 2. With that change, Docker has altered its data model removing support of Docker images and introducing support for manifests. As a result of this fundamental change, Docker version 1 support is completely removed; therefore, any existing Docker version 1 repositories are deleted as part of the upgrade.

The previously created containers will still be visible in Satellite Server and can still be started, however you will not be able to create new containers as the images will no longer be present.

To assist in creating Docker version 2 repositories post upgrade, obtain and save the details of any Docker version 1 repositories prior to the upgrade.

This can be accomplished using the Satellite web UI or the Hammer CLI.

To view existing Docker repositories using the Satellite web UI:

1. Go to **Content** > **Products**
2. Select the repositories to display the settings.

To view existing Docker repositories using the CLI:

1. List all docker repositories for a specific organization:

```
# hammer repository list --organization-id 1 --content-type docker
```

2. Obtain docker repository information for a specific repository:

```
# hammer repository info --id 3
```

Once the upgrade has been completed, if the above repositories are available from a Docker version 2 registry, they can be recreated with the appropriate details (such as name, docker-upstream-name, and URL). The process to manage Docker version 2 content is similar to the process in Satellite 6.1 with Docker version 1 and other content types: create and synchronize repositories, then create, publish, and promote Content Views.

6.1. MIGRATING FROM RED HAT ENTERPRISE LINUX 6 TO RED HAT ENTERPRISE LINUX 7

Red Hat Satellite 6.2 is the last y-stream version that Red Hat Enterprise Linux 6 supports. To upgrade to a later version of Red Hat Satellite 6, you must first migrate to Red Hat Enterprise Linux 7. You might also want to migrate your existing Satellite installation to Red Hat Enterprise Linux 7 and later upgrade Red Hat Satellite.

Red Hat Satellite 6.2.13 introduces **satellite-clone**. This tool copies an existing Satellite installation from an existing Red Hat Enterprise Linux server to a new Red Hat Enterprise Linux 7 server.

The Satellite clone tool does not support migrating a Capsule Server to Red Hat Enterprise Linux 7. Instead you must backup the existing Capsule Server, restore it on Red Hat Enterprise Linux 7, then reconfigure the Capsule Server.

6.1.1. Exclusions

The Satellite migration process includes only the Red Hat Satellite environment. Any customizations made outside the Satellite environment, for example manually configured cron jobs, are outside scope.

6.1.2. Before You Begin

Terminology

Throughout this procedure, ensure that you understand the following terminology:

- Source server - existing Satellite Server or Capsule Server
- Target server - new server, to which Satellite Server or Capsule Server is being migrated

In migrating a Satellite Server or Capsule Server, the following requirements apply:

- Install only a minimal Red Hat Enterprise Linux 7 instance. Do not install any Red Hat Enterprise Linux 7 software groups, or third-party applications. Complete the prerequisites in the Preparing your environment for installation section of the *Installation Guide*.
- The target server must have capacity to store the backup files, which the source server transfers to the target server, and the backup files when they are restored.
- If you are registering the target server using an activation key, obtain it from the Red Hat Customer Portal.
  For more information, see Subscription Asset Manager: Activation Key in the *Subscription Concepts and Workflows* Guide.

6.1.3. Satellite Server Migration Overview

The high-level steps of migrating Satellite Server or a Capsule Server are as follows:

1. Backup the source Satellite Server or Capsule Server.

2. Stop all Satellite services.
3. Copy the backup files to the target server.
4. Migrate the source Satellite Server or Capsule Server.
5. Shut down the source Satellite Server or Capsule Server.
6. Finalize the target Satellite Server or Capsule Server configuration.
7. Decommission the source Satellite Server or Capsule Server.

WARNING

Isolate the target server from internal networks to avoid unwanted communication with Capsule Servers and hosts. Reconnect the target server after the source server is decommissioned.

A connection to the Red Hat Content Delivery Network (CDN) is required for subscription activation and Satellite installation, unless installing from ISO files.

6.1.4. Migrating a Satellite Server

The following procedure outlines how to migrate an existing Satellite Server installation to the target server.

1. On the source server, identify the current Red Hat Satellite subscription:

```
# subscription-manager list --consumed \
--matches 'Red Hat Satellite'|grep "Pool ID:"|awk '{print $3}'
```

   Note the Red Hat Satellite subscription pool ID because you need this in a later step.

2. On the source server, remove the Red Hat Satellite subscription:

```
# subscription-manager remove --serial=$(subscription-manager list \
--consumed \
--matches 'Red Hat Satellite'|grep "Serial:"|awk '{print $2}')
```

3. On the source server, perform an online Satellite backup with only the databases active. The backup files are output to the **backup** directory:

```
# katello-service stop
# service postgresql start
# service mongod start
# katello-backup --online-backup /backup/
```

4. On the source server, stop and disable Katello services.

   a. To stop all Katello services, enter the following command:

```
# katello-service stop
```

b. Disable all Katello services.
On Red Hat Enterprise Linux 7, run the following script:

```
# for i in $(katello-service list | cut -d' ' -f1);do systemctl
disable $i;done
```

On Red Hat Enterprise Linux 6, run the following script:

```
# for i in $(katello-service list | cut -d' ' -f1);do chkconfig
$i off;done
```

5. Shut down the Red Hat Enterprise Linux server hosting the source server.

6. Register the target server to the Customer Portal, add the necessary subscriptions, and enable the Satellite repositories. For more information, see Installing Satellite Server from a Connected Network in the *Installation Guide*. Optionally you can choose to register using an activation key in the Customer Portal which is explained in later steps.

a. Register the target server to the Customer Portal, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

b. Attach the subscription to the target server:

```
# subscription-manager attach --pool=pool_ID
```

The Red Hat Satellite subscription pool ID is identified earlier in this procedure.

c. Ensure only the required Red Hat Enterprise Linux 7 repositories are enabled:

```
# subscription-manager repos --disable=*
# subscription-manager repos \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-6.2-rpms
```

7. On the target server, either mount the shared storage or copy the following files and folder from the source Satellite Server. The default destination directory is **/backup**. You can choose another directory if there is not enough storage space.

- **candlepin.dump**
- **config_files.tar.gz**
- **foreman.dump**
- **mongo_dump** folder
- **pulp_data.tar**

8. On the target server, install the **satellite-clone** package:

```
# yum install satellite-clone
```

9. On the target server, customize the **satellite-clone** configuration file.
Edit the configuration file **/etc/satellite-clone/satellite-clone-vars.yml**. To change an item from the default, remove the comment character (#), and edit its value.

   a. If the backup files were not copied to the directory **/backup** on the target server, change the **backup_dir** to the path containing the backup files.

   b. Change the **rhel_migration** to **yes**.

   c. If you did not register the target server to the Customer Portal, you can use an activation key in the Customer Portal to register it:

      i. Change the **register_to_portal** to **true**.

      ii. Change the **activationkey** to the name of your activation key in the Customer Portal.

      iii. Change the **orgname** to your Satellite organization in the Customer Portal.

   d. Change other values to suit your requirements.

10. Start the **satellite-clone** process:

```
# satellite-clone
```

11. On the target server, reconfigure DHCP, DNS, TFTP and remote execution services. The cloning process disables these services on the target Satellite Server to avoid conflict with the source Satellite Server.
Reconfigure and enable DHCP, DNS, TFTP in the Satellite web UI. For more information, see Configuring DNS, DHCP, and TFTP on Satellite Server in the *Installation Guide*.

    Enable remote execution:

```
# satellite-installer --scenario satellite --enable-foreman-plugin-
remote-execution --enable-foreman-proxy-plugin-remote-execution-ssh
```

12. Log in to the Satellite web UI as user **admin** and password **changeme**. Change the **admin** user's password.

13. Refresh the Satellite's manifest.

    a. Log in to the Satellite web UI.

    b. Ensure that the correct organization is selected.

    c. Navigate to **Content > Red Hat Subscriptions**, then click **Manage Manifest**.

    d. Click **Refresh Manifest**.

    e. Navigate to **Content > Red Hat Subscriptions** and verify the available subscriptions are correct.

14. Edit Capsules' association with lifecycles.
    The cloning process breaks the association between Capsule Servers and their lifecycle environments to avoid interference with existing infrastructure. On the target Satellite Server, follow the instructions in file **`/usr/share/satellite-clone/logs/reassociate_capsules.txt`** to reverse these changes.

15. Update your network configuration, for example, DNS, to match the target server's IP address with its new host name.

16. If the source server uses the **`virt-who`** daemon, install and configure it on the target server.

    a. Complete the prerequisite steps for the virt-who daemon. For more information, see Prerequisites in the *Virtual Instances Guide*.

    b. Copy all the virt-who configuration files in directory **`/etc/virt-who.d/`** from the source server to the same directory on the target server.

    c. Configure and start the virt-who daemon. For more information, see Configuring and Starting virt-who Service in the *Virtual Instances Guide*.

17. Restart **goferd** on all registered content hosts and Capsules.

    ```
    # systemctl restart goferd
    ```

18. Decommission the source server.

6.1.5. Migrating a Capsule Server

The following procedure outlines how to migrate an existing Capsule Server installation to the target server.

1. Install Red Hat Enterprise Linux 7 server. This is the new Capsule Server.

2. Install the new Capsule Server, but do not proceed to run the **`satellite-installer`**. For more information, see Installing Capsule Server in the *Installation Guide*, but stop after completing the *Installing Capsule Server* section.

3. Backup the source Capsule Server. For more information, see Backing up Red Hat Satellite Server in the *Server Administration Guide*.

4. Stop all Satellite services on the source Capsule Server.

    ```
    # katello-service stop
    ```

5. Copy all backup files from the source server to the target server.

6. Restore the Capsule Server on the target server.

7. Shut down the source Capsule Server.

8. Update your network configuration, for example, DNS, to match the target server's IP address with its new host name.

9. Decommission the source Capsule Server.

6.2. UPGRADING TO SATELLITE SERVER 6.2

IMPORTANT

The Satellite 6 upgrading program is based on Puppet, which means that any manual configuration changes might be overwritten to the installation defaults. To avoid this, use the **`--noop`** argument when you run the upgrading program to determine what changes would be applied after you perform the upgrade. This argument ensures that no actual changes are made. Potential changes are written to **`/var/log/foreman-installer/satellite.log`**.

6.3. UPGRADING A CONNECTED SATELLITE SERVER

Before You Begin

- You have upgraded to the minor version 6.1.9 or later of Red Hat Satellite Server 6.1. The requirement is to be at least on 6.1.9, upgrading to a higher minor version past 6.1.9 is not needed for the upgrade to Red Hat Satellite 6.2. Direct upgrades from earlier minor versions are not supported. For more information, see Upgrading Between Minor Versions of Satellite in the *Red Hat Satellite 6.1 Installation Guide*.
- Review and update your firewall configuration prior to upgrading your Satellite Server. For additional information, see Section 2.5, “Ports and Firewalls Requirements”.
- Make sure you **DO NOT** delete the manifest from the Customer Portal or in the Satellite Web UI as this will unregister all of your content hosts.
- Backup and remove all Foreman hooks before upgrading. Put any hooks back only after Satellite is known to be working after the upgrade is complete.

WARNING

You must retain the content of both the **`root/ssl-build`** directory and the directory in which you created any source files associated with your custom certificates. Even if you choose to implement custom certificates, you must retain the content of the **`/root/ssl-build`** directory when performing upgrades. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrade Satellite Server

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.
2. A pre-upgrade script is available to detect conflicts and list hosts which have duplicate entries in Satellite Server that can be unregistered and deleted after upgrade. In addition, it will detect

hosts which are not assigned to an organization. If a host is listed under **Hosts** > **All hosts** without an organization association and if a content host with same name has an organization already associated with it then the content host will automatically be unregistered. This can be avoided by associating such hosts to an organization before upgrading.
Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.

a. To use the pre-upgrade script, ensure you have **`ruby193-rubygem-katello-2.2.0.90-1-sat`** or later.

```
# yum update ruby193-rubygem-katello
```

b. Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.

```
# foreman-rake katello:upgrade_check
```

If the upgrade check reports a failure due to running tasks, then 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.

3. Backup the DNS and DHCP configuration files **`/etc/zones.conf`** and **`/etc/dhcp/dhcpd.conf`** as the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.

4. If you have made manual edits to DNS or DHCP configuration files and do not want the changes overwritten, run the following command.

```
# katello-installer --capsule-dns-managed=false \
--capsule-dhcp-managed=false
```

5. Disable the repository for Red Hat Satellite 6.1.

- On Red Hat Enterprise Linux 6, run the following command:

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

- On Red Hat Enterprise Linux 7, run the following command:

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

6. If required, to verify that the Satellite 6.1 repositories are now disabled, enter a command as follows:

```
# subscription-manager repos --list-enabled
```

7. Enable the repositories for Red Hat Satellite 6.2.

- On Red Hat Enterprise Linux 6, run the following command:

```
# subscription-manager repos --enable=rhel-6-server-rpms \
--enable=rhel-server-rhscl-6-rpms \
--enable=rhel-6-server-satellite-6.2-rpms
```

- On Red Hat Enterprise Linux 7, run the following command:

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-6.2-rpms
```

8. Clear out any metadata left from any non-Red Hat yum repositories.

```
# yum clean all
```

9. Verify that the repositories have been enabled.

```
# yum repolist enabled
```

Output similar to the following displays:

```
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-6.2-rpms/x86_64         Red Hat Satellite 6.2
(for RHEL 7 Server) (RPMs)                              545
!rhel-server-rhscl-7-rpms/x86_64                 Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server     4,279
repolist: 14,713
```

10. In the Satellite web UI, go to **Hosts** > **Discovered hosts**. If there are discovered hosts available, turn them off and then delete all entries under the **`Discovered hosts`** page. Select all other organizations in turn using the organization setting menu and repeat this action as required. Reboot these hosts after the upgrade has completed.

11. Make sure all external Capsule Servers are assigned to an organization, otherwise they might get unregistered due to host-unification changes.

12. Configure the repositories in the Satellite web UI.

    a. In the Satellite web UI, go to **Content** > **Red Hat Repositories** and select the **RPM** tab.

    b. Find and expand the Red Hat Enterprise Linux Server **Product**.

    c. Find and expand Red Hat Satellite Tools 6.2 (for Red Hat Enterprise Linux X Server) (RPMs).

    d. Select Red Hat Satellite Tools 6.2 for Red Hat Enterprise Linux X Server RPMs x86_64.

13. Synchronize the newly enabled repositories.

    a. In the Satellite web UI, go to **Content** > **Sync Status**.

    b. Click the arrow next to the product to view available repositories.

c. Select the repositories for 6.2.

d. Click **Synchronize Now**.
If you get an error when trying to update the Satellite Tools repository, make sure you **DO NOT** delete the manifest from the Customer Portal or in the Satellite Web UI as this will unregister all of your content hosts. See the Red Hat Knowledgebase solution Cannot enable Red Hat Satellite Tools Repo on Satellite 6.2 for more information.

14. Update any pre-existing Content Views that utilize 6.1 version repositories with the new version for 6.2. Publish and promote updated versions of any Content Views that now have the new 6.2 version repositories.

15. Clear the repository cache.

```
# yum clean all
```

16. Run the upgrade check again to ensure that the steps that we have already undertaken have not resulted in a situation in which tasks might get stuck during the upgrade:

```
# foreman-rake katello:upgrade_check
```

17. Stop Katello services.

```
# katello-service stop
```

18. Update all packages.

```
# yum update
```

19. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **`--noop`** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

a. Add the following line to the **`/etc/httpd/conf/httpd.conf`** configuration file.

```
Include /etc/httpd/conf.modules.d/*.conf
```

b. Restart the **`httpd`** service.

- On Red Hat Enterprise Linux 6, enter the following command:

```
# service httpd restart
```

- On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl restart httpd
```

c. Start the **`postgresql`** and **`mongod`** database services.

- On Red Hat Enterprise Linux 6, enter the following command:

```
# service postgresql start
```

```
# service mongod start
```

- On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start postgresql
# systemctl start mongod
```

d. Enter the command with the **--noop** option as follows.

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

Review the **/var/log/foreman-installer/satellite.log** to see what changes **would be applied** if the **--noop** option was omitted. Look for the **+++** and **---** symbols indicating changes to configurations files. Because the above "no operation" command does not actually create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

e. Stop Katello services.

```
# katello-service stop
```

20. Perform the upgrade by running the installer script with the **--upgrade** option.

```
# satellite-installer --scenario satellite --upgrade
```

WARNING

If you run the command from a directory containing a ***config*** subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

In such a case, change directory, for example to the ***root*** user's home directory, and run the command again.

21. Check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

22. If you made changes in the previous step, restart Katello services.

```
# katello-service restart
```

23. Update the Discovery template in the Satellite web UI.

a. Go to **Hosts** > **Provisioning templates**.

b. Select **PXELinux global default**.

c. In the **Template editor** dialog box, edit the **PXELinux global default** template discovery menu entry, updating the stanza that begins with **LABEL discovery** to match the following text:

```
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.
- The **`proxy.url`** specifies the URL of the Satellite Capsule or Server. Both HTTP and HTTPS protocols are supported.

24. If you have the OpenSCAP plug-in installed, but do not have the default OpenSCAP content available, run the following command.

```
# foreman-rake foreman_openscap:bulk_upload:default
```

25. In the Satellite web UI, go to **Configure** > **Discovery Rules** and associate selected organizations and locations with discovery rules.

26. After upgrading, add the **`--assumeyes`** parameter to **`katello-backup`** commands in your backup script.
From Satellite 6.2.13, you must add the **`--assumeyes`** parameter because the **`katello-backup`** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

6.4. UPGRADING A DISCONNECTED SATELLITE SERVER

Before You Begin

- You should have upgraded to the latest minor release of Red Hat Satellite Server 6.1. Direct upgrade from earlier minor versions is not supported. For more information, see Upgrading Between Minor Versions of Satellite in the *Red Hat Satellite 6.1 Installation Guide*.
- Review and update your firewall configuration prior to upgrading your Satellite Server. For additional information, see Section 2.5, “Ports and Firewalls Requirements”.
- Make sure you **DO NOT** delete the manifest from the Customer Portal or in the Satellite Web UI as this will unregister all of your content hosts.

- Backup and remove all Foreman hooks before upgrading. Put any hooks back only after Satellite is known to be working after the upgrade is complete.

WARNING

You must retain the content of both the **`root/ssl-build`** directory and the directory in which you created any source files associated with your custom certificates. Even if you choose to implement custom certificates, you must retain the content of the **`/root/ssl-build`** directory when performing upgrades. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrade Disconnected Satellite Server

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.
2. A pre-upgrade script is available to detect conflicts and list hosts which have duplicate entries in Satellite Server that can be unregistered and deleted after upgrade. In addition, it will detect hosts which are not assigned to an organization. If a host is listed under **Hosts** > **All hosts** without an organization association and if a content host with same name has an organization already associated with it then the content host will automatically be unregistered. This can be avoided by associating such hosts to an organization before upgrading.
   Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.
   a. To use the pre-upgrade script, ensure you have **`ruby193-rubygem-katello-2.2.0.90-1-sat`** or later.

   ```
   # yum update ruby193-rubygem-katello
   ```

   b. Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.

   ```
   # foreman-rake katello:upgrade_check
   ```

   If the upgrade check reports a failure due to running tasks, then 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.
3. Backup the DNS and DHCP configuration files **`/etc/zones.conf`** and **`/etc/dhcp/dhcpd.conf`** as the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.

4. If you have made manual edits to DNS or DHCP configuration files and do not want the changes overwritten, run the following command.

```
# katello-installer --capsule-dns-managed=false \
--capsule-dhcp-managed=false
```

5. In the Satellite web UI, go to **Hosts** > **Discovered hosts**. If there are discovered hosts available, turn them off and then delete all entries under the **`Discovered hosts`** page. Select all other organizations in turn using the organization setting menu and repeat this action as required. Reboot these hosts after the upgrade has completed.

6. Make sure all external Capsule Servers are assigned to an organization, otherwise they might get unregistered due to host-unification changes.

7. Stop Katello services.

```
# katello-service stop
```

8. Obtain the ISO file, mount it, and install the packages. For more information, see Section 3.2, “Downloading and Installing from a Disconnected Network”.

9. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **`--noop`** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

   a. Add the following line to the **`/etc/httpd/conf/httpd.conf`** configuration file.

```
Include /etc/httpd/conf.modules.d/*.conf
```

   b. Restart the **`httpd`** service.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service httpd restart
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl restart httpd
```

   c. Start the **`postgresql`** and **`mongod`** database services.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service postgresql start
# service mongod start
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start postgresql
# systemctl start mongod
```

   d. Enter the command with the **`--noop`** option as follows.

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

Review the **/var/log/foreman-installer/satellite.log** to see what changes **would be applied** if the **--noop** option was omitted. Look for the **+++** and **---** symbols indicating changes to configurations files. Because the above "no operation" command does not actually create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

e. Stop Katello services.

```
# katello-service stop
```

10. Perform the upgrade by running the installer script with the **--upgrade** option.

```
# satellite-installer --scenario satellite --upgrade
```

WARNING

If you run the command from a directory containing a ***config*** subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

In such a case, change directory, for example to the ***root*** user's home directory, and run the command again.

11. Check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

12. If you made changes in the previous step, restart Katello services.

```
# katello-service restart
```

13. Update the Discovery template in the Satellite web UI.

a. Go to **Hosts** > **Provisioning templates**.

b. Select **PXELinux global default**.

c. In the **Template editor** dialog box, edit the **PXELinux global default** template discovery menu entry, updating the stanza that begins with **LABEL discovery** to match the following text:

```
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.
- The **`proxy.url`** specifies the URL of the Satellite Capsule or Server. Both HTTP and HTTPS protocols are supported.

14. If you have the OpenSCAP plug-in installed, but do not have the default OpenSCAP content available, run the following command.

```
# foreman-rake foreman_openscap:bulk_upload:default
```

15. In the Satellite web UI, go to **Configure** > **Discovery Rules** and associate selected organizations and locations with discovery rules.

16. After upgrading, add the **`--assumeyes`** parameter to **`katello-backup`** commands in your backup script.
From Satellite 6.2.13, you must add the **`--assumeyes`** parameter because the **`katello-backup`** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

6.5. UPGRADING CAPSULE SERVERS

Before You Begin

- You should have upgraded the Satellite Server before upgrading any Capsule Servers.
- The capsule should be on at least the 6.1.9 minor version of Red Hat Satellite Server 6.1. The requirement is to be at least on 6.1.9, upgrading to a higher minor version past 6.1.9 is not needed for the upgrade to Red Hat Satellite 6.2. Direct upgrades from earlier minor versions are not supported. For more information, see Upgrading Between Minor Versions of Satellite in the *Red Hat Satellite 6.1 Installation Guide*.
- Ensure the Capsule's base system is registered to the newly upgraded Satellite Server.
- Ensure the Capsule has the correct organization and location settings in the newly upgraded Satellite Server.
- Review and update your firewall configuration prior to upgrading your Capsule Server. For additional information, see Section 2.5, "Ports and Firewalls Requirements".

WARNING

You must retain the content of both the **root/ssl-build** directory and the directory in which you created any source files associated with your custom certificates. Even if you choose to implement custom certificates, you must retain the content of the **/root/ssl-build** directory when performing upgrades. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrading Capsule Servers

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.
2. Backup the DNS and DHCP configuration files **/etc/zones.conf** and **/etc/dhcp/dhcpd.conf** as the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.
3. If you have made manual edits to DNS or DHCP configuration files and do not want the changes overwritten, enter the following command.

```
# capsule-installer --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

4. Disable the repository for Red Hat Satellite 6.1.
   - On Red Hat Enterprise Linux 6, run the following command:

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

   - On Red Hat Enterprise Linux 7, run the following command:

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

5. Enable the new repositories.
   The Red Hat Software Collections repository provides a later version of Ruby required by some Red Hat Satellite features, including the Remote Execution feature.
   - On Red Hat Enterprise Linux 7, run the following command:

```
# subscription-manager repos \
--enable rhel-7-server-satellite-capsule-6.2-rpms \
--enable rhel-server-rhscl-7-rpms
```

- On Red Hat Enterprise Linux 6, run the following command:

```
# subscription-manager repos \
--enable rhel-6-server-satellite-capsule-6.2-rpms \
--enable rhel-server-rhscl-6-rpms
```

6. In the Satellite web UI, go to **Hosts** > **Discovered hosts**. If there are discovered hosts available, turn them off and then delete all entries under the **`Discovered hosts`** page. Select all other organizations in turn using the organization setting menu and repeat this action as required. Reboot these hosts after the upgrade has completed.

7. Clear the repository cache.

```
# yum clean all
```

8. Stop Katello services:

```
# katello-service stop
```

9. Update all packages.

```
# yum update
```

10. On the Satellite Server, generate an archive with new certificates.

```
# capsule-certs-generate --capsule-fqdn "mycapsule.example.com" \
--certs-tar "mycapsule.example.com-certs.tar"
```

You should replace ***mycapsule.example.com*** with the fully qualified domain name of the Capsule Server.

11. Copy the archive file to the Capsule Server.

```
# scp mycapsule.example.com-certs.tar mycapsule.example.com:~/
```

WARNING

Do not remove the certificate archive file after the upgrade, it is required for future updates.

12. If you plan to use Capsule Server as a proxy for discovered hosts, install the Discovery plug-in.

```
# yum install rubygem-smart_proxy_discovery.noarch
```

13. On the Capsule Server, verify that the **`foreman_url`** setting is correct.

```
# grep foreman_url /etc/foreman-proxy/settings.yml
```

The fully qualified domain name of the Satellite Server should display.

14. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **`--noop`** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

    a. Add the following line to the **`/etc/httpd/conf/httpd.conf`** configuration file.

    ```
    Include /etc/httpd/conf.modules.d/*.conf
    ```

    b. Restart the **`httpd`** service.

    - On Red Hat Enterprise Linux 6, enter the following command:

    ```
    # service httpd restart
    ```

    - On Red Hat Enterprise Linux 7, enter the following command:

    ```
    # systemctl restart httpd
    ```

    c. Start the **`mongod`** database service.

    - On Red Hat Enterprise Linux 6, enter the following command:

    ```
    # service mongod start
    ```

    - On Red Hat Enterprise Linux 7, enter the following command:

    ```
    # systemctl start mongod
    ```

    d. Enter the command with the **`--noop`** option as follows.

    ```
    # satellite-installer --scenario capsule --upgrade --verbose --
    noop
    ```

    Review the **`/var/log/foreman-installer/capsule.log`** to see what changes **would be applied** if the **`--noop`** option was omitted. Look for the **`+++`** and **`---`** symbols indicating changes to configurations files. Because the above "no operation" command does not actually create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

    e. Stop Katello services.

    ```
    # katello-service stop
    ```

15. Perform the upgrade by running the installer script with the **`--upgrade`** option, and specify the path to the certificate archive previously created on the Satellite Server.

    ```
    # satellite-installer --scenario capsule --upgrade \
    --capsule-certs-tar mycapsule.example.com-certs.tar
    ```

WARNING

If you run the command from a directory containing a ***config*** subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/capsule.yaml) was not found,
can not continue.
```

In such a case, change directory, for example to the ***root*** user's home directory, and run the command again.

16. Check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

17. Upgrade the foreman-discovery package on Satellite Server and turn on the hosts that were shut down prior to the upgrade.

18. After upgrading, add the **`--assumeyes`** parameter to **`katello-backup`** commands in your backup script.
    From Satellite 6.2.13, you must add the **`--assumeyes`** parameter because the **`katello-backup`** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

6.6. UPGRADING DISCOVERY ON CAPSULE SERVERS

1. Verify that all relevant packages are current on the Satellite Server.

```
# yum upgrade tfm-rubygem-foreman_discovery
```

2. Restart Katello services, if applicable.

```
# katello-service restart
```

3. 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
```

4. On the same instance, install the package which provides the Proxy service, and then restart foreman-proxy service.

```
# yum install rubygem-smart_proxy_discovery
# service foreman-proxy restart
```

5. In the Satellite web UI, go to **Infrastructure** > **Capsules** and verify that the relevant proxy lists the Discovery feature. Click **Refresh features** if necessary.

6. Go to **Infrastructure** > **Subnets** and select the required Smart Proxy for each subnet on which you want to use discovery, and verify that it is connected to the Discovery Proxy.

Update the Discovery template in the Satellite web UI.

a. Go to **Hosts** > **Provisioning Templates**.

b. Select **PXELinux global default**.

c. In the **Template editor** dialog box, edit the **PXELinux global default** template discovery menu entry, updating the stanza that begins with **LABEL discovery** to match the following text:

```
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.
- The **`proxy.url`** specifies the URL of the Satellite Capsule or Server. Both HTTP and HTTPS protocols are supported.
- You can omit the proxy.url option and determine the Capsule DNS name from its SRV record. This is useful if there are multiple discovery subnets. For more information, see Configuring PXE-booting in the *Red Hat Satellite Host Configuration Guide*.

6.7. UPGRADING SATELLITE CLIENTS

You must upgrade all clients to the new version of **`katello-agent`** so that your clients are compatible with Satellite Server. This requires changing the Satellite Tools repository from 6.1 to 6.2, which can be done manually or by installing the **`satellite-tools-upgrade`** package. This package only contains a post installation script to change the Satellite Tools repository version.

Before You Begin

- You must have upgraded Satellite Server.
- You must have enabled the new Satellite Tools repositories on the Satellite.
- You must have synchronized the new repositories in the Satellite.
- If you have not previously installed **`katello-agent`** on your clients, use the manual method.

WARNING

You must retain the content of both the **`root/ssl-build`** directory and the directory in which you created any source files associated with your custom certificates. Even if you choose to implement custom certificates, you must retain the content of the **`/root/ssl-build`** directory when performing upgrades. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrade Satellite Clients Using the satellite-tools-upgrade Package

1. In the Satellite web UI, go to **Hosts** > **Content Hosts** or **Host Collections** and select the Content Hosts to be upgraded.
2. On the **Packages** tab, enter the package name **`satellite-tools-upgrade`** in the search field.
    a. If upgrading a single host using the Content Host view, select **Perform** to install the package.
    b. If upgrading a collection of hosts using the Bulk Actions view, select **Install** to install the package.
3. On the **Packages** tab, enter the package name **`katello-agent`** in the search field.
    a. If upgrading a single host using the Content Host view, select **Package Update** and then **Perform** to update the package.
    b. If upgrading a collection of hosts using the Bulk Actions view, select **Update** to update the package.

NOTE

Until Red Hat Bugzilla 1291960 is resolved, you will see duplicate package versions installed on your systems after attempting to upgrade **`katello-agent`** using the web UI or the **hammer** CLI. See the bug for more details.

Upgrade Satellite Clients Manually

1. Log into the client system.
2. Disable the repositories for the previous version of Satellite.
    - On Red Hat Enterprise Linux 7, run the following command:

```
# subscription-manager repos \
--disable rhel-7-server-satellite-tools-6.1-rpms
```

    - On Red Hat Enterprise Linux 6, run the following command:

```
# subscription-manager repos \
--disable rhel-6-server-satellite-tools-6.1-rpms
```

3. Enable the Satellite tools repository for this version of Satellite.
   - On Red Hat Enterprise Linux 7, run the following command:

     ```
     # subscription-manager repos \
     --enable=rhel-7-server-satellite-tools-6.2-rpms
     ```

   - On Red Hat Enterprise Linux 6, run the following command:

     ```
     # subscription-manager repos \
     --enable=rhel-6-server-satellite-tools-6.2-rpms
     ```

4. Upgrade the Katello Agent package.

   ```
   # yum upgrade katello-agent
   ```

6.8. UPGRADING A SELF-REGISTERED SATELLITE SERVER

You can synchronize your self-registered Satellite Server with the Red Hat Customer Portal, publish and promote Content Views, and upgrade your self-registered Satellite Server.

Before You Begin

- You have upgraded to at least the minor version 6.1.9 of Red Hat Satellite Server 6.1. The requirement is to be at least on 6.1.9, upgrading to a higher minor version past 6.1.9 is not needed for the upgrade to Red Hat Satellite 6.2. Direct upgrades from earlier minor versions are not supported. For more information, see Upgrading Between Minor Versions of Satellite in the *Red Hat Satellite 6.1 Installation Guide*.
- Review and update your firewall configuration prior to upgrading your Satellite Server. For additional information, see Section 2.5, “Ports and Firewalls Requirements”.
- Make sure you **DO NOT** delete the manifest from the Customer Portal or in the Satellite Web UI as this will unregister all of your content hosts.
- Backup and remove all Foreman hooks before upgrading. Put any hooks back only after Satellite is known to be working after the upgrade is complete.

WARNING

You must retain the content of both the **`root/ssl-build`** directory and the directory in which you created any source files associated with your custom certificates. Even if you choose to implement custom certificates, you must retain the content of the **`/root/ssl-build`** directory when performing upgrades. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrade Self-Registered Satellite Server

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.
2. A pre-upgrade script is available to detect conflicts and list hosts which have duplicate entries in Satellite Server that can be unregistered and deleted after upgrade. In addition, it will detect hosts which are not assigned to an organization. If a host is listed under **Hosts** > **All hosts** without an organization association and if a content host with same name has an organization already associated with it then the content host will automatically be unregistered. This can be avoided by associating such hosts to an organization before upgrading.
   Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.
   a. To use the pre-upgrade script, ensure you have **`ruby193-rubygem-katello-2.2.0.90-1-sat`** or later.

   ```
   # yum update ruby193-rubygem-katello
   ```

   b. Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.

   ```
   # foreman-rake katello:upgrade_check
   ```

   If the upgrade check reports a failure due to running tasks, then 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.
3. Backup the DNS and DHCP configuration files **`/etc/zones.conf`** and **`/etc/dhcp/dhcpd.conf`** as the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.
4. If you have made manual edits to DNS or DHCP configuration files and do not want the changes overwritten, run the following command.

```
# katello-installer --capsule-dns-managed=false \
--capsule-dhcp-managed=false
```

5. List the enabled repositories:

```
# subscription-manager repos --list-enabled
```

6. Ensure you have the following repositories enabled and no others:

```
rhel-X-server-satellite-tools-6.1-rpms
rhel-server-rhscl-X-rpms
rhel-X-server-satellite-6.1-rpms
rhel-X-server-rpms
```

Where X is the major version of the base system. If other repositories are found, follow the Section 3.1.3, “Configuring Repositories” procedure to remove them.

7. Disable the repositories for the previous version of Satellite on the base system.

- On Red Hat Enterprise Linux 6, enter the following commands:

```
# subscription-manager repos \
--disable rhel-6-server-satellite-6.1-rpms
# subscription-manager repos \
--disable rhel-6-server-satellite-tools-6.1-rpms
```

- On Red Hat Enterprise Linux 7, enter the following commands:

```
# subscription-manager repos \
--disable rhel-7-server-satellite-6.1-rpms
# subscription-manager repos \
--disable rhel-7-server-satellite-tools-6.1-rpms
```

8. If required, to verify that the Satellite 6.1 repositories are now disabled, enter a command as follows:

```
# subscription-manager repos --list-enabled
```

9. Configure the repositories in the Satellite web UI.

a. In the Satellite web UI, go to **Content** > **Red Hat Repositories** and select the **RPM** tab.

b. Find and expand the Red Hat Satellite **Product**.

c. Find and expand the **Repository Set** Red Hat Satellite 6.1 (for Red Hat Enterprise Linux X Server) (RPMs).

d. Unselect Red Hat Satellite 6.1 for Red Hat Enterprise Linux X Server RPMs x86_64.

e. Find and expand the **Repository Set** Red Hat Satellite 6.2 (for Red Hat Enterprise Linux X Server) (RPMs).

f. Select Red Hat Satellite 6.2 for Red Hat Enterprise Linux X Server RPMs x86_64.

g. Find and expand the Red Hat Enterprise Linux Server **Product**.

h. Find and expand Red Hat Satellite Tools 6.2 (for Red Hat Enterprise Linux X Server) (RPMs).

i. Select Red Hat Satellite Tools 6.2 for Red Hat Enterprise Linux X Server RPMs x86_64.

j. In the **Other** tab, find and expand Red Hat Software Collections for RHEL Server.

k. Find and expand Red Hat Software Collections RPMs (for Red Hat Enterprise Linux X Server).

l. Select Red Hat Software Collections RPMs for Red Hat Enterprise Linux X Server x86_64.

10. Synchronize the newly enabled repositories.

    a. In the Satellite web UI, go to **Content** > **Sync Status**.

    b. Click the arrow next to the product to view available repositories.

    c. Select the repositories for 6.2.

    d. Click **Synchronize Now**.
    If you get an error when trying to update the Satellite Tools repository, make sure you **DO NOT** delete the manifest from the Customer Portal or in the Satellite Web UI as this will unregister all of your content hosts. See the Red Hat Knowledgebase solution Cannot enable Red Hat Satellite Tools Repo on Satellite 6.2 for more information.

11. Update any pre-existing Content Views that utilize 6.1 version repositories with the new version for 6.2. Publish and promote updated versions of any Content Views that now have the new 6.2 version repositories.

12. Enable the new repositories on the base system when the repositories have finished synchronizing.

    - On Red Hat Enterprise Linux 6, enter the following commands:

    ```
    # subscription-manager repos \
    --enable rhel-6-server-satellite-6.2-rpms \
    --enable rhel-6-server-satellite-tools-6.2-rpms \
    --enable rhel-6-server-rpms \
    --enable rhel-server-rhscl-6-rpms
    ```

    - On Red Hat Enterprise Linux 7, enter the following commands:

    ```
    # subscription-manager repos \
    --enable rhel-7-server-satellite-6.2-rpms \
    --enable rhel-7-server-satellite-tools-6.2-rpms \
    --enable rhel-7-server-rpms \
    --enable rhel-server-rhscl-7-rpms
    ```

13. In the Satellite web UI, go to **Hosts** > **Discovered hosts**. If there are discovered hosts available, turn them off and then delete all entries under the **`Discovered hosts`** page. Select all other organizations in turn using the organization setting menu and repeat this action as required. Reboot these hosts after the upgrade has completed.

14. Make sure all external Capsule Servers are assigned to an organization, otherwise they might get unregistered due to host-unification changes.

15. Clear the repository cache.

```
# yum clean all
```

16. Download the following packages.

```
# yum install --downloadonly rubygem-
smart_proxy_remote_execution_ssh \
rubygem-smart_proxy_openscap rubygem-smart_proxy_dynflow \
tfm-rubygem-smart_proxy_dynflow_core \
tfm-rubygem-foreman_remote_execution katello-client-bootstrap
```

17. Download all updated packages.

```
# yum update --downloadonly
```

18. Stop Katello services.

```
# katello-service stop
```

19. Install the previously downloaded packages.

```
# yum install rubygem-smart_proxy_remote_execution_ssh \
rubygem-smart_proxy_openscap rubygem-smart_proxy_dynflow \
tfm-rubygem-smart_proxy_dynflow_core \
tfm-rubygem-foreman_remote_execution katello-client-bootstrap
```

20. Install all updated packages from the **yum** cache.

```
# yum update --cacheonly
```

21. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **--noop** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

    a. Add the following line to the **/etc/httpd/conf/httpd.conf** configuration file.

    ```
    Include /etc/httpd/conf.modules.d/*.conf
    ```

    b. Restart the **httpd** service.

    - On Red Hat Enterprise Linux 6, enter the following command:

    ```
    # service httpd restart
    ```

    - On Red Hat Enterprise Linux 7, enter the following command:

    ```
    # systemctl restart httpd
    ```

c. Start the **postgresql** and **mongod** database services.

- On Red Hat Enterprise Linux 6, enter the following command:

```
# service postgresql start
# service mongod start
```

- On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start postgresql
# systemctl start mongod
```

d. Enter the command with the **--noop** option as follows.

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

Review the **/var/log/foreman-installer/satellite.log** to see what changes **would be applied** if the **--noop** option was omitted. Look for the **+++** and **---** symbols indicating changes to configurations files. Because the above "no operation" command does not actually create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

e. Stop Katello services.

```
# katello-service stop
```

22. Perform the upgrade by running the installer script with the **--upgrade** option.

```
# satellite-installer --scenario satellite --upgrade
```

WARNING

If you run the command from a directory containing a ***config*** subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

In such a case, change directory, for example to the ***root*** user's home directory, and run the command again.

23. Check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

24. If you made changes in the previous step, restart Katello services.

```
# katello-service restart
```

25. Update the Discovery template in the Satellite web UI.

    a. Go to **Hosts** > **Provisioning templates**.

    b. Select **PXELinux global default**.

    c. In the **Template editor** dialog box, edit the **PXELinux global default** template discovery menu entry, updating the stanza that begins with **LABEL discovery** to match the following text:

    ```
    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.
    - The **`proxy.url`** specifies the URL of the Satellite Capsule or Server. Both HTTP and HTTPS protocols are supported.

26. If you have the OpenSCAP plug-in installed, but do not have the default OpenSCAP content available, run the following command.

    ```
    # foreman-rake foreman_openscap:bulk_upload:default
    ```

27. In the Satellite web UI, go to **Configure** > **Discovery Rules** and associate selected organizations and locations with discovery rules.

28. Synchronize the Satellite Server with the Red Hat Customer Portal.

    a. Go to **Content > Sync Status**.
    A list of product repositories available for synchronization is displayed.

    b. Click the arrow next to the product content to view available content.

    c. Select the content you want to synchronize.

    d. Click **Synchronize Now**.
    Content synchronization can take a long time and it depends on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

29. (Optional) Publish the required Content Views.
    You must publish the Content View so that it is visible and usable by hosts. Before publishing, you should ensure that the Content View definition has the necessary products, repositories, and filters.

    a. From the Main Menu, select **Content > Content Views**.

b. From the Name column, select the Satellite Server Content View.

c. Click **Publish New Version**.

d. Enter a comment and click **Save**.

30. (Optional) Promote the Content View.

a. From the Main Menu, select **Content > Content Views**.

b. In the Name column, select the Satellite Server Content View.

c. On the Versions tab, select the latest version and click **Promote**.

d. Identify the promotion path, select the appropriate life-cycle environment, and click **Promote Version**.
When the operation completes, you can see the updated Content View status on the **Versions** tab.

31. After upgrading, add the **`--assumeyes`** parameter to **`katello-backup`** commands in your backup script.
From Satellite 6.2.13, you must add the **`--assumeyes`** parameter because the **`katello-backup`** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

6.9. POST UPGRADE CLEANUP

All the procedures in this section are optional. You can choose to perform only those procedures that are relevant to your installation.

6.9.1. Removing Redundant Firewall Rules

Red Hat Satellite 6.2 does not use Elasticsearch and therefore firewall rules related to Elasticsearch can be removed. These are the lines with destination port 9200.

Removing Redundant Firewall Rules on Red Hat Enterprise Linux 6

1. List the firewall rules.

```
# iptables -nL --line-numbers
```

2. Identify the following lines and remove them. Note that the chain name is OUTPUT and the line numbers might differ.

```
Chain OUTPUT (policy ACCEPT)
num  target   prot opt source         destination
1    ACCEPT   tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
owner UID match 496
2    ACCEPT   tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
owner UID match 0
3    DROP     tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
```

3. Remove the iptables rules.

```
iptables -D <chain-name> <line-number>
```

For example, to remove line 1 from the above output, enter a command as follows:

```
# iptables -D OUTPUT 1
```

4. After removing the lines, save the changes.

```
# service iptables save
```

5. Make sure the iptables service is started and enabled.

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

Removing Redundant Firewall Rules on Red Hat Enterprise Linux 7

1. List the direct rules.

   a. For IPv4:

   ```
   # firewall-cmd --direct --get-rules ipv4 filter OUTPUT
   ```

   b. For IPv6:

   ```
   # firewall-cmd --direct --get-rules ipv6 filter OUTPUT
   ```

2. Remove the firewall direct rules.

   a. For IPv4:

   ```
   firewall-cmd --direct --remove-rule IPv4 filter <chain_name> rule
   ```

   Example:

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

   b. For IPv6:

   ```
   firewall-cmd --direct --remove-rule IPv6 filter <chain_name> rule
   ```

   Example:

   ```
   # firewall-cmd --direct --remove-rule ipv6 filter OUTPUT 0 -o lo
   -p \
   ```

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

3. Ensure the firewall service is enabled and started.

```
# systemctl enable firewalld
# systemctl start firewalld
```

6.9.2. Removing Elasticsearch

Red Hat Satellite 6.2 does not use Elasticsearch and therefore the packages and directory used by Elasticsearch can be removed.

Removing Elasticsearch and Related Packages

Remove the following packages as they are no longer needed.

```
# yum erase elasticsearch sigar-java sigar snappy-java \
lucene4-contrib lucene4
```

Removing Elasticsearch User

Remove the user created by Elasticsearch.

```
# userdel -r elasticsearch
```

Removing Elasticsearch Directory

Remove the database directory and its contents.

```
# rm -rf /var/lib/elasticsearch
```

6.9.3. Removing the Previous Version of the Satellite Tools Repository

After completing the upgrade to Satellite 6.2, the Red Hat Satellite Tools 6.1 repository can be removed from Content Views and then disabled.

1. Disable Version 6.1 of the Satellite Tools Repository

    a. In the Satellite web UI, go to **Content** > **Red Hat Repositories** and select the **RPM** tab.

    b. Find and expand the Red Hat Enterprise Linux Server **Product**.

    c. Find and expand the **Repository Set** Red Hat Satellite Tools 6.1 (for Red Hat Enterprise Linux X Server) (RPMs).

    d. Unselect Red Hat Satellite Tools 6.1 for Red Hat Enterprise Linux X Server RPMs x86_64.

If the check box is dimmed, then the repository is still contained in a Content View. The orphaned packages in the repository will be removed automatically by a scheduled task (cron job).

CHAPTER 7. UPDATING SATELLITE SERVER, CAPSULE SERVER, AND CONTENT HOSTS

Updating Between Minor Versions of Satellite

Updating is the process of migrating Satellite Server, Capsule Server, and Content Hosts to a new minor version. Updates typically patch security vulnerabilities and correct minor issues discovered after code is released. Generally speaking, updates require little time and are non-disruptive to your operating environment. Before updating, check the Red Hat Satellite Release Notes for potential conflicts.

Follow these procedures to update between minor versions, for example, from 6.2.0 to 6.2.1.

7.1. UPDATING SATELLITE SERVER

Prerequisites

- Ensure you have synchronized Satellite Server repositories for Satellite, Capsule, and Satellite Tools.
- Ensure each external Capsule and Content Host can be updated by promoting the updated repositories to all relevant Content Views.

Updating Satellite Server to the Next Minor Version

To Update Satellite Server:

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-6.2-rpms
rhel-server-rhscl-X-rpms
```

   Where *X* is the major version of Red Hat Enterprise Linux you are using. If required, see Section 3.1.3, “Configuring Repositories” for more information on disabling and enabling repositories. If you have a self-registered Satellite, the **`rhel-X-server-satellite-tools-6.2-rpms`** repository, which provides Katello Agent, can also be present. If required, see Section 4.7.1, “Installing the katello Agent” for more information.

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 Katello services:

```
# katello-service stop
```

4. Update all packages:

```
# yum update
```

For a self-registered Satellite, enter the following command:

```
# yum update --cache --disableplugin=enabled_repos_upload \
--disableplugin=package_upload \
--disableplugin=subscription-manager
```

If a kernel update occurs, make a note to reboot **after** the upgrade is complete. Do **not** reboot at this point.

5. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **`--noop`** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

   a. Add the following line to the **`/etc/httpd/conf/httpd.conf`** configuration file.

```
Include /etc/httpd/conf.modules.d/*.conf
```

   b. Restart the **`httpd`** service.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service httpd restart
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl restart httpd
```

   c. Start the **`postgresql`** and **`mongod`** database services.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service postgresql start
# service mongod start
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start postgresql
# systemctl start mongod
```

6. Enter the command with the **`--noop`** option as follows.

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

Review the **`/var/log/foreman-installer/satellite.log`** to see what changes **would be applied** if the **`--noop`** option was omitted. Look for the **`+++`** and **`---`** symbols indicating changes to configurations files. Because the above "no operation" command does not actually

create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

7. Stop Katello services.

```
# katello-service stop
```

8. Perform the update by running the installer script with the **--upgrade** option.

```
# satellite-installer --scenario satellite --upgrade
```

9. If a kernel update occurred during the **yum update** step, reboot the system:

```
# reboot
```

10. If you are on a self-registered Satellite, and did not reboot the system in the previous step, restart **goferd**:

- On Red Hat Enterprise Linux 6, run the following command:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7, run the following command:

```
# systemctl restart goferd
```

11. If you have updated from Satellite 6.2.12 or lower, add the **--assumeyes** parameter to **katello-backup** commands in your backup script if it is not there yet.
From Satellite 6.2.13, you must add the **--assumeyes** parameter because the **katello-backup** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

7.2. UPDATING CAPSULE SERVER

Updating Capsule Servers to the Next Minor Version

To Update a Capsule Server:

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-6.2-rpms
rhel-server-rhscl-X-rpms
rhel-X-server-satellite-tools-6.2-rpms
```

Where *X* is the major version of Red Hat Enterprise Linux you are using. If required, see

Section 4.3, “Configuring Repositories” for more information on disabling and enabling repositories. The **rhel-X-server-satellite-tools-6.2-rpms** repository provides Katello Agent. If required, see Section 4.7.1, “Installing the katello Agent” for more information. The Red Hat Software Collections repository is optional, it is required for the Remote Execution feature.

2. Stop Katello services:

```
# katello-service stop
```

3. Update all packages:

```
# yum update
```

If a kernel update occurs, make a note to reboot **after** the upgrade is complete. Do **not** reboot at this point.

4. If you have custom Apache server configurations, they will be reverted to the installation defaults in the next step. If you want to see what will be changed **when you perform the upgrade**, you can enter the upgrade command with the **--noop** (no operation) option and review the changes that will be applied when you enter the upgrade command in the following step. If you choose not to do this test, skip to the next step now. Alternatively, proceed as follows:

   a. Add the following line to the **/etc/httpd/conf/httpd.conf** configuration file.

```
Include /etc/httpd/conf.modules.d/*.conf
```

   b. Restart the **httpd** service.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service httpd restart
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl restart httpd
```

   c. Start the **mongod** database service.

   - On Red Hat Enterprise Linux 6, enter the following command:

```
# service mongod start
```

   - On Red Hat Enterprise Linux 7, enter the following command:

```
# systemctl start mongod
```

   d. Enter the command with the **--noop** option as follows.

```
# satellite-installer --scenario capsule --upgrade --verbose --
noop
```

Review the **/var/log/foreman-installer/capsule.log** to see what changes **would**

be applied if the **--noop** option was omitted. Look for the **+++** and **---** symbols indicating changes to configurations files. Because the above "no operation" command does not actually create the files, and some Puppet resources in the module expect them to be there, some failure messages are to be expected.

e. Stop Katello services.

```
# katello-service stop
```

- Perform the update by running the installer script with the **--upgrade** option.

```
# satellite-installer --scenario capsule --upgrade
```

5. If a kernel update occurred during the **yum update** step, reboot the system:

```
# reboot
```

6. If you did not reboot the system in the previous step, restart **goferd**:

- On Red Hat Enterprise Linux 6, run the following command:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7, run the following command:

```
# systemctl restart goferd
```

7. If you have updated from Satellite 6.2.12 or lower, add the **--assumeyes** parameter to **katello-backup** commands in your backup script if it is not there yet.
From Satellite 6.2.13, you must add the **--assumeyes** parameter because the **katello-backup** command prompts for confirmation that the backup is to proceed. For more information about backing up Satellite Server, see Backup and Disaster Recovery in the *Server Administration Guide*.

7.3. UPDATING CONTENT HOSTS

Updating Content Hosts to the Next Minor Version

To Update a Content Host, enter the following commands:

1. Update all packages:

```
# yum update
```

2. If a kernel update occurs, reboot the system:

```
# reboot
```

3. If you did not reboot the system in the previous step, restart **goferd**:

- On Red Hat Enterprise Linux 6, run the following command:

```
# service goferd restart
```

- On Red Hat Enterprise Linux 7, run the following command:

```
# systemctl restart goferd
```

CHAPTER 8. UNINSTALLING SATELLITE SERVER AND CAPSULE SERVER

If you no longer need Satellite Server or Capsule Server, you can uninstall them.

8.1. UNINSTALLING SATELLITE SERVER

Uninstalling Satellite Server and Capsule Server erases all applications used on the target system. If you use any applications or application data for purposes other than Satellite Server, you should back up the information before the removal process.

Before you Begin

The uninstall script issues two warnings, requiring confirmation before removing all packages and configuration files in the system.

WARNING

This script will erase many packages and config files. Important packages such as the following will be removed:

- httpd (apache)
- mongodb
- tomcat6
- puppet
- ruby
- rubygems
- All Katello and Foreman Packages

Uninstall Satellite Server

1. Uninstall Satellite Server.

```
# katello-remove
```

```
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...
```

8.2. UNINSTALLING CAPSULE SERVERS

Uninstalling Capsule Server erases all applications used on the target system. If you use any applications or application data for purposes other than Satellite Server, you should back up the information before the removal process.

Before you Begin

The uninstall script issues two warnings, requiring confirmation before removing all packages and configuration files in the system.

WARNING

This script erases packages and config files. Important packages such as the following will be removed:

- httpd (apache)
- mongodb
- tomcat6
- puppet
- ruby
- rubygems
- All Katello and Foreman Packages

Uninstall Capsule Server

1. Uninstall Capsule Server.

```
$ capsule-remove
```

The following message displays.

```
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...
```

CHAPTER 9. WHERE TO FIND MORE INFORMATION

At the end of the initial installation and setup, you can perform additional configuration and set up your Satellite environment. You can use the following Satellite documentation resources to assist you:

- *Hammer CLI Guide*
- *Server Administration Guide*
- *Host Configuration Guide*
- *Content Management Guide*
- *Puppet Guide*
- *Virtual Instances Guide*

APPENDIX A. LARGE DEPLOYMENT CONSIDERATIONS

Increasing the Maximum Number of File Descriptors for Apache

With more than 800 content hosts registered, Apache can reach several system-level limits, resulting in new content host registration failure. To avoid this, file descriptor limits must be increased before deploying a large number of content hosts.

1. On Red Hat Enterprise Linux 7, create the **`/etc/systemd/system/httpd.service.d/limits.conf`** file and insert the following text:

```
[Service]
LimitNOFILE=65536
```

2. Apply the changes to the unit.

```
# systemctl daemon-reload
```

3. Restart Katello services.

```
# katello-service restart
```

Increasing the Maximum Number of File Descriptors for qpid

With more than 1100 content hosts with goferd running for errata updates, the qpid reach system-level limits, resulting in registration failures. To avoid this, file descriptors limits must be increased before deploying a large number of content hosts.

Increasing the Maximum Number of File Descriptors for qpid Using Red Hat Enterprise Linux 7

1. Create the **`/etc/systemd/system/qpidd.service.d/limits.conf`** file and insert the following text:

```
[Service]
LimitNOFILE=65536
```

2. Apply the changes to the unit.

```
# systemctl daemon-reload
# systemctl restart qpidd.service
```

Increasing the Maximum Number of File Descriptors for qpid Using Red Hat Enterprise Linux 6

1. Edit the **`/etc/security/limits.conf`** file and insert the following text:

```
qpidd - nofile 65536
```

2. Restart the qpidd service.

```
# service qpidd restart
```

Increasing the Shared Buffer and Work Memory

You can increase the **shared_buffer** and **work_mem** to 256M and 4M respectively.

1. On Red Hat Enterprise Linux 7, create the **/var/lib/pgsql/data/postgresql.conf** file and insert the following text:

```
work_mem = 4MB
shared_buffers = 256MB
```

2. Restart postgresql services.

```
# service postgresql restart
```

Increasing Concurrent Content Host Registrations

To avoid reaching system-level limits, you can increase the global passenger queue limit to accommodate up to 250 concurrent content hosts.

1. Adjust the maximum passenger pool size to 1.5 times the physical CPU cores available to the Satellite Server.
   For example, if you have a Satellite Server with 16 cores, then the maximum passenger pool size is 24. This number is referenced as an example and you should use the number applicable to your environment.

2. Edit the **/etc/httpd/conf.d/passenger.conf** file, updating the **IfModule** stanza to match the following text:

```
<IfModule mod_passenger.c>
  PassengerRoot /usr/share/gems/gems/passenger-
4.0.18/lib/phusion_passenger/locations.ini
  PassengerRuby /usr/bin/ruby
  PassengerMaxPoolSize 24
  PassengerMaxRequestQueueSize 200
  PassengerStatThrottleRate 120
</IfModule>
```

3. Edit the Foreman Passenger application configuration file **/etc/httpd/conf.d/05-foreman-ssl.conf**, updating the stanza starting with **PassengerAppRoot** to match the following text:

```
 PassengerAppRoot /usr/share/foreman
 PassengerRuby /usr/bin/tfm-ruby
 PassengerMinInstances 6
 PassengerStartTimeout 90
 PassengerMaxPreloaderIdleTime 0
 PassengerMaxRequests 10000
 PassengerPreStart https://example.com
```

4. Edit the Puppet Passenger application configuration file **/etc/httpd/conf.d/25-puppet.conf**, adding the following text to the end of the virtual host definition:

```
 PassengerMinInstances 6
 PassengerStartTimeout 90
 PassengerMaxPreloaderIdleTime 0
```

```
PassengerMaxRequests 10000
PassengerPreStart https://example.com:8140
```

5. Change the maximum connections in the **/var/lib/pgsql/data/postgresql.conf** file.

```
max_connections = 500
```

6. Restart postgresql services.

```
# service postgresql restart
```

Increasing the maximum number of open files for qdrouterd

With more than 1000 content hosts registered, **qdrouterd** can reach the default maximum number of open files. To avoid this, increase the maximum number of open files on the Satellite Server and all external Capsule Servers.

1. Calculate the required maximum number of open files, using the following equation.

```
(3 x number of content hosts) + 100
```

   For example, with 1020 content hosts, the new maximum should be set to 3160 ((3 x 1020) + 100).

2. On Red Hat Enterprise Linux 7, create the file **/etc/systemd/system/qdrouterd.service.d/limits.conf** and add the following text.

```
[Service]
LimitNOFILE=maximum_number_of_files
```

   a. Apply the changes to the unit.

```
# systemctl daemon-reload
```

   b. Restart the Satellite services.

```
# katello-service restart
```

3. On Red Hat Enterprise Linux 6, edit the file **/etc/security/limits.conf** and add the following line.

```
qdrouterd   -   nofile  maximum_number_of_files
```

   Add the new line **before** the **# End of file** line because anything past that is ignored.

   a. Restart the **qdrouterd** service.

```
# service qdrouterd restart
```

APPENDIX B. CAPSULE SERVER SCALABILITY CONSIDERATIONS

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit. The tested limit is 17 Capsule Servers with 2 vCPUs on a Satellite Server with Red Hat Enterprise Linux 6.6 and 7 hosts. However, scalability is highly variable, especially when managing Puppet clients.

Capsule Server scalability when managing Puppet clients depends on the number of CPUs, the run-interval distribution, and the number of Puppet managed resources. The Capsule Server has a limitation of 100 concurrent Puppet agents running at any single point in time. Running more than 100 concurrent Puppet agents results in a 503 HTTP error.

For example, assuming that 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 4 CPUs has 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.

If you want to scale your Capsule Server when managing Puppet clients, the following assumptions are made:

- There are no external Puppet clients reporting directly to the Satellite 6 integrated Capsule.
- All other Puppet clients report directly to an external Capsule.
- There is an evenly distributed run-interval of all Puppet agents.

NOTE

Deviating from the even distribution increases the risk of filling the passenger request queue. The limit of 100 concurrent requests applies.

The following table describes the scalability limits using the recommended 4 CPUs with Red Hat Enterprise Linux 7.

Table B.1. Puppet Scalability Using 4 CPUs with Red Hat Enterprise Linux 7 (Recommended)

Puppet Managed Resources per Host	Run-Interval Distribution
1	3000-2500
10	2400-2000
20	1700-1400

The following table describes the scalability limits using the minimum 2 CPUs with Red Hat Enterprise Linux 7.

Table B.2. Puppet Scalability Using 2 CPUs with with Red Hat Enterprise Linux 7

Puppet Managed Resources per Host	Run-Interval Distribution
1	1700-1450
10	1500-1250
20	850-700

The following table describes the scalability limits using the recommended 4 CPUs with Red Hat Enterprise Linux 6.

Table B.3. Puppet Scalability Using 4 CPUs with Red Hat Enterprise Linux 6 (Recommended)

Puppet Managed Resources per Host	Run-Interval Distribution
1	2250-1875
10	1600-1250
20	700-560

The following table describes the scalability limits using the minimum 2 CPUs.

Table B.4. Puppet Scalability Using 2 CPUs with Red Hat Enterprise Linux 6

Puppet Managed Resources per Host	Run-Interval Distribution
1	Not tested
10	1020-860
20	375-330

APPENDIX C. APPLYING CUSTOM CONFIGURATION TO RED HAT SATELLITE

When you install and configure Satellite for the first time using **satellite-installer**, you can specify that the DNS and DHCP configuration files are not to be managed by Puppet using **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false**. If these options are not specified during the initial installer run, any manual changes will be overwritten by a rerun of the installer, for example, rerun for upgrade purposes. If changes are overwritten, you will need to run the restore procedure to restore the manual changes. See Section C.1, “How to restore manual changes overwritten by a Puppet run” for more information.

The installer does not have an option for all configuration files that you may want to manage manually. To specify Satellite configuration values which will not be overwritten by the installer, add entries to the configuration file **/etc/foreman-installer/custom-hiera.yaml**. This configuration file is in YAML format, consisting of one entry per line in the format of **<puppet class>::<parameter name>: <value>**. Configuration values specified in this file will persist across installer reruns.

Common examples include:

- For Apache, to set the ServerTokens directive to only return the Product name:

```
apache::server_tokens: Prod
```

- To turn off the Apache server signature entirely:

```
apache::server_signature: Off
```

- To turn off TRACE:

```
apache::trace_enable: Off
```

- For Puppet, to enable the future parser:

```
puppet::server_parser: future
```

- For Pulp, to configure the number of pulp workers:

```
pulp::num_workers: 8
```

C.1. HOW TO RESTORE MANUAL CHANGES OVERWRITTEN BY A PUPPET RUN

If your manual configuration has been overwritten by a Puppet run, you can restore the files to the previous state. The following example shows you how to restore a DHCP configuration file overwritten by a Puppet run.

1. Copy the file you intend to restore. This allows you to compare the files to check for any mandatory changes required by the upgrade. This is not common for DNS or DHCP services.

```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.backup
```

2. Check the log files to note down the md5sum of the overwritten file. For example:

```
# journalctl -xe
...
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed
/etc/dhcp/dhcpd.conf to puppet with sum
622d9820b8e764ab124367c68f5fa3a1
...
```

3. Restore the overwritten file:

```
# puppet filebucket restore --local --bucket \
/var/lib/puppet/clientbucket /etc/dhcp/dhcpd.conf \
622d9820b8e764ab124367c68f5fa3a1
```

4. Compare the backup file and the restored file, and edit the restored file to include any mandatory changes required by the upgrade.