Red Hat Satellite 6.6

Installing Satellite Server from a Disconnected Network

Installing Red Hat Satellite Server from a Disconnected Network

Red Hat Satellite 6.6 Installing Satellite Server from a Disconnected Network

Installing Red Hat Satellite Server from a Disconnected Network

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

Legal Notice

Abstract

This guide describes how to install Red Hat Satellite from a disconnected network, perform initial configuration, and configure external services.

Table of Contents

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base system:

- x86_64 architecture

- The latest version of Red Hat Enterprise Linux 7 Server

- 4-core 2.0 GHz CPU at a minimum

- A minimum of 20 GB RAM is required for Satellite Server to function. In addition, a minimum of 4 GB RAM of swap space is also recommended. Satellite running with less RAM than the minimum value might 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, ensure that your environment meets the requirements for installation.

Satellite Server must be installed on a freshly provisioned system that serves no other function except to run Satellite Server. The freshly provisioned system must not have the following users provided by external identity providers to avoid conflicts with the local users that Satellite Server creates:

- postgres

- mongodb

- apache

- tomcat

- foreman

- foreman-proxy

- qpidd

- qdrouterd

- squid

- puppet

NOTE

The Red Hat Satellite Server and Capsule Server versions must match. For example, a Satellite 6.2 Server cannot run a 6.6 Capsule Server and a Satellite 6.6 Server cannot run a 6.2 Capsule Server. Mismatching Satellite Server and Capsule Server versions results in the Capsule Server failing silently.

Certified hypervisors

Satellite Server is fully supported on both physical systems and virtual machines that run on hypervisors that are supported to run Red Hat Enterprise Linux. For more information about certified hypervisors, see Which hypervisors are certified to run Red Hat Enterprise Linux?

FIPS Mode

You can install Satellite Server on a Red Hat Enterprise Linux system that is operating in FIPS mode. For more information, see Enabling FIPS Mode in the *Red Hat Enterprise Linux Security Guide*.

1.2. STORAGE REQUIREMENTS AND GUIDELINES

This section lists minimum storage requirements and provides storage guidelines for Satellite Server and Capsule Server installation.

1.2.1. Storage Requirements

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

The runtime size was measured with Red Hat Enterprise Linux 6, 7, and 8 repositories synchronized.

Table 1.1. Storage Requirements for a Disconnected Satellite Server Installation

Directory	Installation Size	Runtime Size
/var/cache/pulp/	1 MB	30 GB
/var/lib/pulp/	1 MB	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/lib/qpidd/	25 MB	Not Applicable
/var/log/	10 MB	10 GB
/var/lib/pgsql/	100 MB	10 GB
/var/spool/squid/	0 MB	10 GB
/usr	3 GB	Not Applicable
/opt	3 GB	Not Applicable

Directory	Installation Size	Runtime Size
/opt/puppetlabs	500 MB	Not Applicable

1.2.2. Storage Guidelines

Consider the following guidelines when installing Satellite Server to increase efficiency.

- Because most Satellite and Capsule Server data is stored within the **/var** directory, mounting **/var** on LVM storage can help the system to scale.

- Using the same volume for the **/var/cache/pulp/** and **/var/lib/pulp/** directories can decrease the time required to move content from **/var/cache/pulp/** to **/var/lib/pulp/** after synchronizing.

- Use 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, using high latency, low-bandwidth storage causes performance degradation. Ensure your installation has a speed in the range 60 - 80 Megabytes per second. You can use the **fio** tool to get this data. See the Red Hat Knowledgebase solution Impact of Disk Speed on Satellite 6 Operations for more information on using the **fio** tool.

- For improved performance, use solid state drives (SSD) rather than hard disk drives (HDD).

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

File System Guidelines

- Use the XFS file system for Red Hat Satellite 6 because it does not have the inode limitations that **ext4** does. Because Satellite uses a lot of symbolic links it is likely that your system might run out of inodes if using **ext4** and the default number of inodes.

- Do not use NFS with MongoDB because MongoDB does not use conventional I/O to access data files and performance problems occur when both the data files and the journal files are hosted on NFS. If required to use NFS, mount the volume with the following options in the **/etc/fstab** file: **bg**, **nolock**, and **noatime**.

- Do not use NFS for Pulp data storage. Using NFS for Pulp has a negative performance impact on content synchronization.

- Do not use the GFS2 file system as the input-output latency is too high.

Log File Storage

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 more information, see Log Rotation in the System Administrator's Guide.

The exact amount of storage you require for log messages depends on your installation and setup.

SELinux Considerations for NFS Mount

When **/var/lib/pulp** directory is mounted using an NFS share, SELinux blocks 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 configuration and enter the following command:

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

Duplicated Packages

Packages that are duplicated in different repositories are only stored once on the disk. Additional repositories containing duplicate packages 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. Ensure that storage is available on the **/var** file system to prevent storage problems.

Temporary Storage

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 8 RPM content synchronization tasks can run simultaneously by default, with each using up to 1 GB of metadata.

ISO Images

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.

If you plan to use ISO images for installing or updating, you must provide external storage or allow space in **/var/tmp** for temporarily storing ISO files.

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. Consider the number of ISO files you intend synchronizing because the temporary disk space required for them typically exceeds that of RPM content.

Software Collections

Software collections are installed in the **/opt/rh** and **/opt/theforeman/** directories.

Write and execute permissions by the root user are required for installation to the **/opt** directory.

Symbolic links

You cannot use symbolic links for **/var/lib/pulp/** and **/var/lib/mongodb/**,

1.3. 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 is supported only on the latest versions of Red Hat Enterprise Linux 7 Server that is available at the time when Satellite Server 6.6 is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

Red Hat Satellite Server requires a Red Hat Enterprise Linux installation 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.

Install Satellite Server on a freshly provisioned system. Red Hat does not support using the system for anything other than running Satellite Server.

1.4. SUPPORTED BROWSERS

Satellite supports recent versions of Firefox and Google Chrome browsers.

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

1.5. PORTS AND FIREWALLS REQUIREMENTS

For the components of Satellite architecture to communicate, ensure that the required network ports are open and free on the base operating system. You must also ensure that the required network ports are open on any network-based firewalls.

Use this information to configure any network-based firewalls. Note that some cloud solutions must be specifically configured to allow communications between machines because they isolate machines similarly to network-based firewalls. If you use an application-based firewall, ensure that the application-based firewall permits all applications that are listed in the tables and known to your firewall. If possible, disable the application checking and allow open port communication based on the protocol.

Integrated Capsule

Satellite Server has an integrated Capsule and any host that is directly connected to Satellite Server is a Client of Satellite in the context of this section. This includes the base system on which Capsule Server is running.

Clients of Capsule

Hosts which are clients of Capsules, other than Satellite's integrated Capsule, do not need access to Satellite Server. For more information on Satellite Topology, see Capsule Networking in *Planning for Red Hat Satellite 6*.

Required ports can change based on your configuration.

A matrix table of ports is available in the Red Hat Knowledgebase solution Red Hat Satellite 6.6 List of Network Ports.

The following tables indicate the destination port and the direction of network traffic:

Table 1.2. 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 1.3. 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
5646	TCP	AMQP	The Capsule Qpid dispatch router to the Qpid dispatch router in Satellite
5647	TCP	AMQP	Katello Agent to communicate with the Satellite Qpid dispatch router
8000	TCP	HTTP	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, for the discovery image during provisioning, and for communicating with Satellite Server to copy the SSH keys for Remote Execution (Rex) configuration
53	TCP and UDP	DNS	Client DNS queries to a Satellite's integrated Capsule DNS service (Optional)
67	UDP	DHCP	Client to Satellite's integrated Capsule broadcasts, DHCP broadcasts for Client provisioning from a Satellite's integrated Capsule (Optional)
69	UDP	TFTP	Clients downloading PXE boot image files from a Satellites' integrated Capsule for provisioning (Optional)
5000	TCP	HTTPS	Connection to Katello for the Docker registry (Optional)

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

Table 1.4. 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 1.5. Optional Network Ports

Port	Protocol	Service	Required For
22	TCP	SSH	Satellite and Capsule originated communications, for Remote Execution (Rex) and Ansible.
443	TCP	HTTPS	Satellite originated communications, for vCenter compute resource.
5000	TCP	HTTP	Satellite originated communications, for compute resources in OpenStack or for running 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.

1.6. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER

Capsules and Content Hosts that are clients of a Satellite Server's internal Capsule require access through Satellite's host-based firewall and any network-based firewalls.

Use this section to configure the host-based firewall on the Red Hat Enterprise Linux 7 system that Satellite is installed on, to enable incoming connections from Clients, and to make the configuration persistent across system reboots. For more information on the ports used, see Section 1.5, "Ports and Firewalls Requirements".

Configuring the Firewall

1. To open the ports for client to Satellite communication, enter the following command on the base system that you want to install Satellite on:

```
# firewall-cmd \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5647/tcp" --add-port="8000/tcp" \
--add-port="8140/tcp" --add-port="9090/tcp" \
--add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="5000/tcp"
```

2. Make the changes persistent:

```
# firewall-cmd --runtime-to-permanent
```

1.7. VERIFYING FIREWALL SETTINGS

Use this procedure to verify your changes to the firewall settings.

Procedure

To verify the firewall settings, complete the following step:

1. Enter the following command:

```
# firewall-cmd --list-all
```

For more information, see Getting Started with firewalld in the *Red Hat Enterprise Linux 7 Security Guide*.

1.8. VERIFYING DNS RESOLUTION

Verify the full forward and reverse DNS resolution using a fully-qualified domain name 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 fail. Among these options are content management, subscription management, and provisioning.

CHAPTER 2. INSTALLING SATELLITE SERVER

You can use this chapter to find information about installing Red Hat Satellite Server, performing the initial configuration, creating and installing manifests, and performing additional configuration.

There are two methods of installing Satellite Server:

Connected:

You can obtain the packages required to install Satellite Server directly from the Red Hat Content Delivery Network (CDN). Using the CDN ensures that your system always receives the latest updates.

Disconnected:

You must use an external computer to download an ISO image of the packages and copy the packages to the system you want to install Satellite Server on. Use an ISO image only if you require a disconnected environment. The ISO image might not contain the latest updates.

> **NOTE**
>
> You cannot register Satellite Server to itself.

2.1. 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.

Before You Begin

- A copy of the repositories used in the installation are stored in the **/opt/** directory. Ensure you have a minimum of 3 GB of space for this file system and directory.

2.1.1. 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 **/var/tmp** on the Satellite base system or other accessible storage device.

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

2.1.2. 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/rhel7-server
```

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

```
# mount -o loop rhel7-Server-DVD.iso /media/rhel7-server
```

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

```
# cp /media/rhel7-server/media.repo /etc/yum.repos.d/rhel7-server.repo
```

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

```
baseurl=file:///media/rhel7-server/
```

5. Verify that the repository has been configured.

```
# yum repolist
```

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
```

2.1.3. Installing from the Offline Repositories

1. Ensure the ISO images for Red Hat Enterprise Linux Server and Red Hat Satellite are mounted:

```
# findmnt -t iso9660
```

2. Import the Red Hat GPG keys:

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

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

> # yum update

4. Change to the directory where the Satellite ISO is mounted:

> # cd /media/sat6/

5. Run the installation script in the mounted directory:

> # ./install_packages

If you have successfully installed the Satellite packages, the following message is displayed:
Install is complete. Please run satellite-installer --scenario satellite.

2.1.4. Resolving Package Dependency Errors

If there are package dependency errors during installation of Satellite Server packages, you can resolve the errors by downloading and installing packages from Red Hat Customer Portal. For more information about resolving dependency errors, see the KCS solution How can I use the yum output to solve yum dependency errors?.

If you have successfully installed the Satellite packages, skip this procedure.

Procedure

1. Go to the Red Hat Customer Portal and log in.

2. Click DOWNLOADS.

3. Click the Product that contains the package that you want to download.

4. Ensure that you have the correct **Product Variant**, **Version**, and **Architecture** for your environment.

5. Click the **Packages** tab.

6. In the **Search** field, enter the name of the package.

7. Click the package.

8. From the **Version** list, select the version of the package.

9. At the bottom of the page, click **Download Now**.

10. Copy the package to the Satellite base system.

11. On Satellite Server, change to the directory where the package is located:

> # cd */path-to-package/*

12. Install the package locally:

```
# yum localinstall package_name
```

13. Change to the directory where the Satellite ISO is mounted:

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

14. Verify that you have resoled the package dependency errors by installing the Satellite Server packages. If there are further package dependency errors, repeat this procedure.

```
# ./install_packages
```

If you have successfully installed the Satellite packages, the following message is displayed: **Install is complete. Please run satellite-installer --scenario satellite**.

2.2. PERFORMING THE INITIAL CONFIGURATION

This section details how to perform the initial configuration of the host operating system when installing Red Hat Satellite Server. This includes synchronizing the time, installing the **sos** package, and specifying an installation option.

Before you continue, consider which manifests or packages are relevant for your environment. For more information on manifests, see Managing Subscriptions in the *Red Hat Satellite Content Management Guide*.

2.2.1. Synchronizing the System Clock With chronyd

To minimize the effects of time drift, you must synchronize the system clock on the base system on which you want to install Satellite Server with Network Time Protocol (NTP) servers. If the base system clock is configured incorrectly, certificate verification might fail.

For more information about the **chrony** suite, see Configuring NTP Using the chrony Suite in the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

Procedure

To synchronize the system clock, complete the following steps:

1. Install the **chrony** package:

```
# yum install chrony
```

2. Start and enable the **chronyd** service:

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

2.2.2. 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
```

2.2.3. Specifying Installation Options

Satellite Server is installed using the **satellite-installer** installation script and as part of the initial configuration, you either automatically or manually configure Satellite.

Choose from one of these two methods:

- Automatic Configuration – This method is performed by using an answer file to automate the configuration process when running the installation script. An answer file is a file containing a list of parameters that are read by a command or script. The default Satellite answer file is **/etc/foreman-installer/scenarios.d/satellite-answers.yaml**. The answer file in use is set by the **answer_file** directive in the **/etc/foreman-installer/scenarios.d/satellite.yaml** configuration file.
  To perform the initial configuration using the installation script with an answer file, see Section 2.2.3.2, "Performing the Initial Configuration Automatically using an Answer File" .

- Manual Configuration – This method is performed by running the installation script with one or more command options. The command options override the corresponding default initial configuration options and are recorded in the Satellite answer file. You can run the script as often as needed to configure any necessary options.
  To perform the initial configuration using the installation script with command-line options, see Section 2.2.3.1, "Performing the Initial Configuration Manually" .
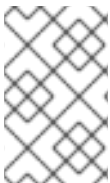
> **NOTE**
>
> Depending on the options that you use when running the Satellite installer, the configuration can take several minutes to complete. An administrator is able to view the answer file to see previously used options for both methods.

2.2.3.1. Performing the Initial Configuration Manually

This initial configuration procedure creates an organization, location, user name, and password. After the initial configuration, you can create additional organizations and locations if required. The initial configuration also installs MongoDB and PostgreSQL databases on the same server. Depending on your deployment, using external databases can benefit performance. For more information see, Section 3.11, "Using External Databases with Satellite".

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.

Considerations for Manual Configuration

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.

Specify a meaningful value for the option: **--foreman-initial-organization**. This can be your company name. An internal label that matches the value is also created and cannot be changed afterwards. 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** runs, it overwrites any manual changes to the Puppet managed files with the initial values. By default, Satellite Server is installed with the Puppet agent running as a service. If required, you can disable Puppet agent on Satellite Server using the **--puppet-runmode=none** option.

If you want to manage DNS files and DHCP files manually, use the **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false** options so that Puppet does not manage the files related to the respective services. For more information on how to apply custom configuration on other services, see Appendix A, *Applying Custom Configuration to Red Hat Satellite* .

Adding Multihomed DHCP details

If you want to use Multihomed DHCP, you must update the network interface file.

1. In the **/etc/systemd/system/dhcpd.service.d/interfaces.conf** file, edit the following line to add Multihomed DHCP:

   ```
   [Service]
   ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid eth0 eth1 eth2
   ```

2. Enter the following command to perform a daemon reload:

   ```
   # systemctl --system daemon-reload
   ```

3. Enter the following command to restart the **dhcpd** service:

   ```
   # systemctl restart dhcpd.service
   ```

Manually Configuring Satellite Server

To perform the initial configuration manually, enter the following command with any additional options that you want to use:

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-initial-admin-username admin_user_name \
--foreman-initial-admin-password admin_password
```

The script displays its progress and writes logs to **/var/log/foreman-installer/satellite.log**.

Unmount the ISO images:

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

2.2.3.2. Performing the Initial Configuration Automatically using 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 the **satellite-installer** script the first time, the answer file is populated with the standard parameter values for installation. You can change the configuration of Satellite Server at any time.

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

Automatically configuring Satellite Server using an Answer File

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, enter 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** script.

   ```
   # 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
   ```

2.2.4. Creating a Subscription Allocation in Customer Portal

You can access your subscription information on the Red Hat Customer Portal. You can also assign subscriptions for use in on-premise management applications, such as Red Hat Satellite, using subscription allocations.

1. Open https://access.redhat.com/ in your browser and log in to the Red Hat account that you used to register the system to Red Hat Subscription Management.

2. Navigate to **Subscriptions** in the upper-left corner of the Customer Portal.

3. Navigate to **Subscription Allocations**.

4. Click **Create New subscription allocation**

5. In the **Name** field, enter a name.

6. From the **Type** list, select the type and version that corresponds to your Satellite Server.

7. Click **Create**.

2.2.5. Adding Subscriptions to an Allocation

The following procedure explains how to add subscriptions to an allocation.

1. Open https://access.redhat.com/ in your browser and log in to the Red Hat account that you used to register the system to Red Hat Subscription Management.

2. Navigate to **Subscriptions** in the upper-left corner of the Customer Portal.

3. Navigate to **Subscription Allocations**.

4. Click the allocation that you want to update.

5. Click the **Subscriptions** tab.

6. Click **Add Subscriptions**.

7. A list of your Red Hat product subscriptions appears. Enter the **Entitlement Quantity** for each product.

8. Click **Submit** to complete the assignment.

When you have added subscriptions to the allocation, export the manifest file.

2.2.6. Exporting a Subscription Manifest from the Customer Portal

While viewing a subscription allocation that has at least one subscription, you can export a manifest in either of two places:

- From the **Details** tab, under the **Subscription** section, by clicking the **Export Manifest** button.

- From the **Subscriptions** tab, by clicking the **Export Manifest** button.

When the manifest is exported, the Customer Portal encodes the selected subscriptions certificates and creates a .zip archive. This is the Subscription Manifest, which can be uploaded into the Satellite Server.

2.2.7. Importing a Subscription Manifest into the Satellite Server

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

Prerequisites

- Ensure that you enable the disconnected mode on your Satellite Server. For more information, see Section 2.2.8, "Enabling the Disconnected Mode" .

For Web UI Users

1. Ensure the context is set to the organization you want to use.

2. Navigate to **Content** > **Subscriptions**.

3. Click **Manage Manifest** to display the manifest page for the organization.

4. Click **Choose file**, select the Subscription Manifest, then click **Upload**.

5. Click **Choose file** and select the manifest archive in a **.zip** format that you have exported from the Customer Portal.

For CLI Users

The Red Hat Satellite 6 CLI requires the manifest to be on the Satellite Server. On your local client system, copy the manifest to the Satellite Server:

```
[user@client ~]$ scp ~/manifest_file.zip root@satellite.example.com:~/.
```

Then import it using the following command:

```
[root@satellite ~]# hammer subscription upload \
--file ~/manifest_file.zip \
--organization "organization_name"
```

After a few minutes, the CLI reports a successful manifest import.

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

2.2.8. Enabling the Disconnected Mode

Enable the disconnected mode on Satellite Server. When the disconnected mode is enabled, Satellite Server does not access the Red Hat Content Delivery Network (CDN).

Procedure

To enable the disconnected mode on Satellite Server, complete the following steps:

1. In the Satellite web UI, navigate to **Administer** > **Settings**.

2. Click the **Content** tab.

3. Set the **Disconnected mode** value to **Yes**.

For the CLI Users

Enter the following command on Satellite Server:

```
# hammer settings set --name content_disconnected --value true
```

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER

3.1. CONFIGURING SATELLITE TO SYNCHRONIZE CONTENT WITH A LOCAL CDN SERVER

In a disconnected environment, you must ensure that Satellite Server contains the required content to provision systems with the latest security updates, errata, and packages. To do this, follow this procedure to download content ISO images from the Red Hat Customer Portal and import them into a local CDN server. You can host the local CDN server on the base operating system of Satellite Server or on a system that is accessible to Satellite over HTTP. Next, you must configure Satellite Server to synchronize content with the local CDN server.

Procedure

1. Log on to the Red Hat Customer Portal at https://access.redhat.com.

2. In the upper left of the window, click **Downloads** and select **Red Hat Satellite**.

3. Click the **Content ISOs** tab. This page lists all the products that are available in your subscription.

4. Click the link for the product name, such as **Red Hat Enterprise Linux 7 Server (x86_64)** to download the ISO image.

5. Copy all of Satellite Content ISO images to a system that you want to use as a local CDN server. For example, the **/root/isos** directory on Satellite Server.
   Note that storing the content on the same system where Satellite is installed is not a requirement. The CDN can be hosted on a different system inside the same disconnected network as long as it is accessible to Satellite Server over HTTP.

6. On the system that you want to use as your local CDN server, create a local directory that is accessible over httpd. For example, **/var/www/html/pub/sat-import/**:

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

7. Create a mount point and temporarily mount the ISO image at that location:

   ```
   # mkdir /mnt/iso
   # mount -o loop /root/isos/first_iso /mnt/iso
   ```

8. Recursively copy content of the first ISO image to the local directory:

   ```
   # cp -ruv /mnt/iso/* /var/www/html/pub/sat-import/
   ```

9. If you do not plan to use the mounted binary DVD ISO image, unmount and remove the mount point:

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

10. Repeat the above step for each ISO image until you have copied all the data from the Content ISO images into **/var/www/html/pub/sat-import/**.

11. Ensure that the SELinux context for the directory is correct:

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

12. In the Satellite web UI, navigate to **Content** > **Subscriptions**.

13. Click **Manage Manifest**.

14. Edit the **Red Hat CDN URL** field to point to the host name of the system that you use as a local CDN server with the newly created directory, for example:
    http://server.example.com/pub/sat-import/

15. Click **Update** and then upload your manifest into Satellite.

3.2. IMPORTING KICKSTART REPOSITORIES

Kickstart repositories are not provided by the Content ISO image. To use Kickstart repositories in your disconnected Satellite, you must download a binary DVD ISO file for the version of Red Hat Enterprise Linux that you want to use and copy the Kickstart files to Satellite.

Procedure

1. Navigate to the Red Hat Customer Portal at https://access.redhat.com/ and log in.

2. In the upper left of the window, click **Downloads**.

3. Locate and click the version of Red Hat Enterprise Linux that you want to use, for example **Red Hat Enterprise Linux 8**.

4. In the Download Red Hat Enterprise Linux window, locate the binary DVD version of the ISO image, for example, **Red Hat Enterprise Linux 8.1 Binary DVD** and click **Download Now**.

5. When the download completes, copy the ISO image to Satellite Server.

6. On Satellite Server, create a mount point and temporarily mount the ISO image at that location:

    ```
    # mkdir /mnt/iso
    # mount -o loop rhel-8.1-x86_64-dvd.iso /mnt/iso
    ```

7. Create Kickstart directories for AppStream and BaseOS:

    ```
    # mkdir /var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
    ```

    ```
    # mkdir /var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
    ```

    Note that if you use Red Hat Enterprise Linux 7, you must create and complete all the following steps in only one directory **/var/www/html/pub/sat-import/content/dist/rhel/server/7/7.7/x86_64/kickstart/**.

8. To the listing files **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/listing** and **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/listing**, append **kickstart** with a new line:

```
kickstart
```

9. To the listing file **/var/www/html/pub/sat-import/content/dist/rhel8/listing**, append the version number of the operating system ISO that you use with a new line. For example, for the RHEL 8.1 binary ISO, add **8.1** with a new line:

```
8.1
```

10. Copy the **kickstart** files from the ISO image:

```
# cp -a /mnt/iso/AppStream/* \
/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart

# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ \
/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
```

Note that for BaseOS, you must also copy the contents of the **/mnt/iso/images/** directory.

11. Copy the **.treeinfo** files from the ISO image:

```
# cp /mnt/iso/.treeinfo \
/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo

# cp /mnt/iso/.treeinfo \
/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo
```

12. Open the **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo** file for editing.

13. In the **[general]** section, make the following changes:

    - Change **packagedir = AppStream/Packages** to **packagedir = Packages**

    - Change **repository = AppStream** to **repository = .**

    - Change **variant = AppStream** to **variant = BaseOS**

    - Change **variants = AppStream,BaseOS** to **variants = BaseOS**

14. In the **[tree]** section, change **variants = AppStream,BaseOS** to **variants = BaseOS**.

15. In the **[variant-BaseOS]** section, make the following changes:

    - Change **packages = BaseOS/Packages** to **packages = Packages**

    - Change **repository = BaseOS** to **repository = .**

16. Delete the **[media]** and **[variant-AppStream]** sections.

17. Save and close the file.

18. Verify that the **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo** file has the following format:

```
[checksums]
```

```
images/efiboot.img =
sha256:9ad9beee4c906cd05d227a1be7a499c8d2f20b3891c79831325844c845262bb6
images/install.img =
sha256:e246bf4aedfff3bb54ae9012f959597cdab7387aadb3a504f841bdc2c35fe75e
images/pxeboot/initrd.img =
sha256:a66e3c158f02840b19c372136a522177a2ab4bd91cb7269fb5bfdaaf7452efef
images/pxeboot/vmlinuz =
sha256:789028335b64ddad343f61f2abfdc9819ed8e9dfad4df43a2694c0a0ba780d16

[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 8.1.0

[header]
type = productmd.treeinfo
version = 1.2

[images-x86_64]
efiboot.img = images/efiboot.img
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[images-xen]
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[stage2]
mainimage = images/install.img

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = BaseOS

[variant-BaseOS]
id = BaseOS
name = BaseOS
packages = Packages
```

```
repository = .
type = variant
uid = BaseOS
```

19. Open the **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo** file for editing.

20. In the **[general]** section, make the following changes:

    - Change **packagedir = AppStream/Packages** to **packagedir = Packages**

    - Change **repository = AppStream** to **repository = .**

    - Change **variants = AppStream,BaseOS** to **variants = AppStream**

21. In the **[tree]** section, change **variants = AppStream,BaseOS** to **variants = AppStream**.

22. In the **[variant-AppStream]** section, make the following changes:

    - Change **packages = AppStream/Packages** to **packages = Packages**

    - Change **repository = AppStream** to **repository = .**

23. Delete the following sections from the file: **[checksums]**, **[images-x86_64]**, **[images-xen]**, **[media]**, **[stage2]**, **[variant-BaseOS]**.

24. Save and close the file.

25. Verify that the **/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo** file has the following format:

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
version = 8.1.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[tree]
arch = x86_64
```

```
build_timestamp = 1571146127
platforms = x86_64,xen
variants = AppStream

[variant-AppStream]
id = AppStream
name = AppStream
packages = Packages
repository = .
type = variant
uid = AppStream
```

26. If you do not plan to use the mounted binary DVD ISO image, unmount and remove the directory:

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

27. In the Satellite web UI, enable the Kickstart repositories.

3.3. REVERTING SATELLITE TO DOWNLOAD CONTENT FROM RED HAT CDN

If your environment changes from disconnected to connected, you can reconfigure a disconnected Satellite to download content directly from the Red Hat CDN.

Procedure

1. In the Satellite web UI, navigate to **Content** > **Subscriptions**.

2. Click **Manage Manifest**.

3. Edit the **Red Hat CDN URL** field to point to the Red Hat CDN URL:
   https://cdn.redhat.com

4. Click **Save**.

Satellite Server is now configured to download content from the CDN the next time that it synchronizes repositories.

3.4. ENABLING 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 Capsule Server is a client of Satellite Server and therefore must also have the Katello agent installed.

Prerequisites

- Ensure that you import all content ISO images that you require into Satellite Server.

Procedure

To enable the Satellite Tools repository, complete the following steps:

1. In the Satellite web UI, navigate to **Content** > **Red Hat Repositories**.

2. Use the Search field to enter the following repository name: **Red Hat Satellite Tools 6.6 (for RHEL 7 Server) (RPMs)**.

3. In the Available Repositories pane, click on **Red Hat Satellite Tools 6.6 (for RHEL 7 Server) (RPMs)** to expand the repository set.
   If the **Red Hat Satellite Tools 6.6** 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. For the **x86_64** entry, click the **Enable** icon to enable the repository.

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.

For CLI Users

Enable the Satellite Tools repository using the **hammer repository-set enable** command:

```
# hammer repository-set enable --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--basearch='x86_64' \
--name 'Red Hat Satellite Tools 6.6 (for RHEL 7 Server) (RPMs)'
```

3.5. SYNCHRONIZING THE SATELLITE TOOLS REPOSITORY

Use this section to synchronize the Satellite Tools repository from the Red Hat Content Delivery Network (CDN) to your Satellite. This repository repository provides the **katello-agent** and **puppet** packages for clients registered to Satellite Server.

Procedure

To synchronize the Satellite Tools repository, complete the following steps:

1. In the Satellite web UI, navigate to **Content** > **Sync Status**.
   A list of product repositories available for synchronization is displayed.

2. Click the arrow next to the **Red Hat Enterprise Linux Server** product to view available content.

3. Select **Red Hat Satellite Tools 6.6 (for RHEL 7 Server) RPMs x86_64**

4. Click **Synchronize Now**.

For CLI Users

Synchronize your Satellite Tools repository using the **hammer repository synchronize** command:

```
# hammer repository synchronize --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--name 'Red Hat Satellite Tools 6.6 for RHEL 7 Server RPMs x86_64' \
--async
```

3.6. ENABLING POWER MANAGEMENT ON MANAGED HOSTS

To perform power management tasks on managed hosts using the intelligent platform management interface (IPMI) or a similar protocol, you must enable the baseboard management controller (BMC) module on Satellite Server.

Prerequisites

- All managed hosts must have a network interface of BMC type. Satellite Server uses this NIC to pass the appropriate credentials to the host. For more information, see Adding a Baseboard Management Controller (BMC) Interface in *Managing Hosts*.

Procedure

- To enable BMC, enter the following command:

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

3.7. CONFIGURING DNS, DHCP, AND TFTP ON SATELLITE SERVER

To configure the DNS, DHCP, and TFTP services on Satellite Server, use the **satellite-installer** command with the options appropriate for your environment.

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

Any changes to the settings require entering the **satellite-installer** command again. You can enter the command multiple times and each time it updates all configuration files with the changed values.

Prerequisites

Before you can configure DNS, DHCP and TFTP services, ensure that the following conditions are met:

- Ensure that the following information is available to you:

    - 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.

- Contact your network administrator to ensure that you have the correct settings.

Procedure

- Enter the **satellite-installer** command with the options appropriate for your environment. The following example shows configuring full provisioning services:

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns true \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-reverse 2.0.192.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "192.0.2.100 192.0.2.150" \
--foreman-proxy-dhcp-gateway 192.0.2.1 \
--foreman-proxy-dhcp-nameservers 192.0.2.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername 192.0.2.3
```

You can monitor the progress of the **satellite-installer** command displayed in your prompt. You can view the logs in **/var/log/foreman-installer/satellite.log**. You can view the settings used, including the **admin_password** parameter, in the **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** file.

For more information about configuring DHCP, DNS, and TFTP services, see the Configuring Network Services section in the *Provisioning Guide*.

For more information about disabling the DNS, DHCP, and TFTP services in Satellite and managing them manually, see Disabling DNS, DHCP, and TFTP for Unmanaged Networks .

For more information about configuring external services, see Configuring External Services.

3.8. DISABLING DNS, DHCP, AND TFTP FOR UNMANAGED NETWORKS

If you want to manage TFTP, DHCP, and DNS services manually, you must prevent Satellite from maintaining these services on the operating system and disable orchestration to avoid DHCP and DNS validation errors. However, Satellite does not remove the back-end services on the operating system.

Procedure

To prevent Satellite from maintaining DHCP, DNS, and TFTP services on the operating system, and disable orchestration, complete the following steps:

1. On Satellite Server, enter the following command:

   ```
   # satellite-installer --foreman-proxy-dhcp false \
   --foreman-proxy-dns false \
   --foreman-proxy-tftp false
   ```

2. In the Satellite web UI, navigate to **Infrastructure** > **Subnets** and select a subnet.

3. Click the **Capsules** tab and clear the **DHCP Capsule**, **TFTP Capsule**, and **Reverse DNS Capsule** fields.

4. Navigate to **Infrastructure** > **Domains** and select a domain.

5. Clear the **DNS Capsule** field.

6. Optional: If you use a DHCP service supplied by a third party, configure your DHCP server to pass the following options:

> Option 66: *IP_address_of_Satellite_or_Capsule*
> Option 67: /pxelinux.0

For more information about DHCP options, see RFC 2132.

> **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 to turn orchestration on, make sure the required DHCP and DNS records as well as the TFTP files are in place for the existing Satellite hosts in order to prevent host deletion failures in the future.

3.9. CONFIGURING SATELLITE SERVER FOR OUTGOING EMAILS

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

Prerequisites

If you have upgraded from a previous release, rename or remove the configuration file **/usr/share/foreman/config/email.yaml** and restart the **httpd** service. For example:

```
# mv /usr/share/foreman/config/email.yaml \
/usr/share/foreman/config/email.yaml-backup
# systemctl restart httpd
```

To Configure Satellite Server for Outgoing Emails:

1. In the Satellite web UI, navigate to **Administer → Settings**.

2. Click the **Email** tab and set the configuration options to match your preferred delivery method. The changes have an immediate effect.

   a. The following example shows the configuration options for using an SMTP server:

   Table 3.1. Using an SMTP server as a delivery method

Name	Example value
Delivery method	SMTP
SMTP address	*smtp.example.com*
SMTP authentication	login
SMTP HELO/EHLO domain	*example.com*
SMTP password	*password*
SMTP port	25

Name	Example value
SMTP username	*satellite@example.com*

The **SMTP username** and **SMTP password** specify the login credentials for the SMTP server.

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

Table 3.2. Using gmail.com as an SMTP server

Name	Example value
Delivery method	SMTP
SMTP address	smtp.gmail.com
SMTP authentication	plain
SMTP HELO/EHLO domain	smtp.gmail.com
SMTP enable StartTLS auto	Yes
SMTP password	*password*
SMTP port	587
SMTP username	*user*@gmail.com

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

Table 3.3. Using sendmail as a delivery method

Name	Example value
Delivery method	Sendmail
Sendmail arguments	–i –t –G

The **Sendmail arguments** specify the options passed to the **sendmail** command. The default value is **-i -t**. For more information see the **sendmail 1** man page.

3. If you decide to send email using 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, in the web UI, set the **SMTP enable StartTLS auto** option to **No**.

4. Click **Test email** to send a test message to the user's email address to confirm the configuration is working. If a message fails to send, the web UI displays an error. See the log at **/var/log/foreman/production.log** for further details.

> **NOTE**
>
> For information on configuring email notifications for individual users or user groups, see Configuring Email Notifications in *Administering Red Hat Satellite* .

3.10. CONFIGURING SATELLITE SERVER WITH A CUSTOM SSL CERTIFICATE

By default, Red Hat Satellite 6 uses a self-signed SSL certificate to enable encrypted communications between Satellite Server, external Capsule Servers, and all hosts. If you cannot use a Satellite self-signed certificate, you can configure Satellite Server to use an SSL certificate signed by an external Certificate Authority.

To configure your Satellite Server with a custom certificate, complete the following procedures:

1. Section 3.10.1, "Creating a Custom SSL Certificate for Satellite Server"

2. Section 3.10.2, "Deploying a Custom SSL Certificate to Satellite Server"

3. Section 3.10.3, "Deploying a Custom SSL Certificate to Hosts"

4. If you have external Capsule Servers registered to Satellite Server, you must configure them with custom SSL certificates. For more information, see Configuring Capsule Server with a Custom SSL Certificate in *Installing Capsule Server* .

3.10.1. Creating a Custom SSL Certificate for Satellite Server

Use this procedure to create a custom SSL certificate for Satellite Server. If you already have a custom SSL certificate for Satellite Server, skip this procedure.

When you configure Satellite with custom certificates, note the following considerations:

- You must use the Privacy-Enhanced Mail (PEM) encoding for the SSL certificates.

- You cannot use the same certificate for both Satellite and Capsule.

- The same Certificate Authority must sign certificates for Satellite and Capsule.

Procedure

To create a custom SSL certificate, complete the following steps:

1. To store all the source certificate files, create a directory that is accessible only to the **root** user.

```
# mkdir /root/satellite_cert
```

2. Create a private key with which to sign the Certificate Signing Request (CSR).
   Note that the private key must be unencrypted. If you use a password-protected private key,
   remove the private key password.

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

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

3. Create the **/root/satellite_cert/openssl.cnf** configuration file for the Certificate Signing
   Request (CSR) and include the following content:

   ```
   [ req ]
   req_extensions = v3_req
   distinguished_name = req_distinguished_name
   x509_extensions = usr_cert
   prompt = no

   [ req_distinguished_name ]  ❶
   C  = Country Name (2 letter code)
   ST = State or Province Name (full name)
   L  = Locality Name (eg, city)
   O  = Organization Name (eg, company)
   OU = The division of your organization handling the certificate
   CN = satellite.example.com  ❷

   [ v3_req ]
   basicConstraints = CA:FALSE
   keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
   extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
   subjectAltName = @alt_names

   [ usr_cert ]
   basicConstraints=CA:FALSE
   nsCertType = client, server, email
   keyUsage = nonRepudiation, digitalSignature, keyEncipherment
   extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
   nsComment = "OpenSSL Generated Certificate"
   subjectKeyIdentifier=hash
   authorityKeyIdentifier=keyid,issuer

   [ alt_names ]
   DNS.1 = satellite.example.com  ❸
   ```

   ❶ In the **[req_distinguished_name]** section, enter information about your organization.

   ❷ Set the certificate's Common Name **CN** to match the fully qualified domain name (FQDN)
   of your Satellite Server. To confirm a FQDN, on that Satellite Server, enter the **hostname -
   f** command. This is required to ensure that the **katello-certs-check** command validates the
   certificate correctly.

   ❸ Set the Subject Alternative Name (SAN) **DNS.1** to match the fully qualified domain name
   (FQDN) of your server.

4. Generate the Certificate Signing Request (CSR):

```
# openssl req -new \
-key /root/satellite_cert/satellite_cert_key.pem \ 1
-config /root/satellite_cert/openssl.cnf \ 2
-out /root/satellite_cert/satellite_cert_csr.pem 3
```

1   Path to the private key.

2   Path to the configuration file.

3   Path to the CSR to generate.

5. Send the certificate signing request to the Certificate Authority. The same Certificate Authority must sign certificates for Satellite Server and Capsule Server.
When you submit the request, 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.

3.10.2. Deploying a Custom SSL Certificate to Satellite Server

Use this procedure to configure your Satellite Server to use a custom SSL certificate signed by a Certificate Authority. The **katello-certs-check** command validates the input certificate files and returns the commands necessary to deploy a custom SSL certificate to Satellite Server.

Procedure

To deploy a custom certificate on your Satellite Server, complete the following steps:

1. Validate the custom SSL certificate input files. Note that for the **katello-certs-check** command to work correctly, Common Name (CN) in the certificate must match the FQDN of Satellite Server.

```
# katello-certs-check \
-c /root/satellite_cert/satellite_cert.pem \ 1
-k /root/satellite_cert/satellite_cert_key.pem \ 2
-b /root/satellite_cert/ca_cert_bundle.pem 3
```

1   Path to the Satellite Server certificate file that is signed by a Certificate Authority.

2   Path to the private key that was used to sign the Capsule Server certificate.

3   Path to the Certificate Authority bundle.

If the command is successful, it returns two **satellite-installer** commands, one of which you must use to deploy a certificate to Satellite Server.

Example output of katello-certs-check

```
Validation succeeded.

To install the Red Hat Satellite Server with the custom certificates, run:
```

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

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

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

2. From the output of the **katello-certs-check** command, depending on your requirements, enter the **satellite-installer** command that deploys a certificate to a new Satellite or updates a certificate on a currently running Satellite.
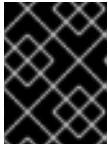
> **IMPORTANT**
>
> Do not delete the certificate archive file after you deploy the certificate. It is required, for example, when upgrading Satellite Server.

3. On a computer with network access to Satellite Server, navigate to the following URL: **https://*satellite.example.com***.

4. In your browser, view the certificate details to verify the deployed certificate.

3.10.3. Deploying a Custom SSL Certificate to Hosts

After you configure Satellite Server to use a custom SSL certificate, you must also install the **katello-ca-consumer** package on every host that is registered to this Satellite Server.

Until BZ#1683835 is resolved, you cannot upgrade the **katello-ca-consumer** package; you must remove the old package and install the new one. Upgrading the **katello-ca-consumer** package fails because the upgrade reverts the **baseurl** setting in **rhsm.conf** to **subscription.rhsm.redhat.com**.

Procedure

On each host, complete the following steps to install the **katello-ca-consumer** package:

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

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

2. Install the **katello-ca-consumer** package on the host:

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

3.11. USING EXTERNAL DATABASES WITH SATELLITE

As part of the installation process for Red Hat Satellite, the **satellite-installer** command installs MongoDB and PostgreSQL databases on the same server as Satellite. In certain Satellite deployments,

using external databases can help with the server load. However, there are many factors that can affect Satellite Server's performance. Moving to an external database might not help your specific problem.

Depending on your requirements, you can use external databases for either MongoDB or PostgreSQL database, or both.

Red Hat does not provide support or tools for external database maintenance. This includes backups, upgrades, and database tuning. Customers using an external database require their own database administrator to support and maintain the database.

If your Satellite deployment requires external databases, use the following information to set up and point to external databases from Satellite.

FIPS-related Restrictions

Until BZ#1743706 is resolved, you cannot use an external MongoDB with Satellite in FIPS mode.

3.11.1. MongoDB as an External Database Considerations

Pulp uses the MongoDB database. If you want to use MongoDB as an external database, the following information can help you discern if this option is right for your Satellite configuration. Satellite supports MongoDB version 3.4.

Advantages of External MongoDB

- Increase in free memory and free CPU on Satellite

- Flexibility to tune the MongoDB server's system without adversely affecting Satellite operations

Disadvantages of External MongoDB

- Increase in deployment complexity that can make troubleshooting more difficult

- An external MongoDB server is an additional system to patch and maintain

- If either the Satellite or the Mongo database server suffers a hardware or storage failure, Satellite is not operational

- If there is latency between the Satellite and the external database server, performance can suffer

If you suspect that your Mongo database is slow, you can work with Red Hat Support to troubleshoot. You might be encountering a configuration problem or existing performance problems with Satellite 6 that moving to an external database server might not help. Red Hat Support can examine existing known issues and also work with the Satellite Engineering team to determine the root cause.

3.11.2. PostgreSQL as an External Database Considerations

Foreman, Katello, and Candlepin use the PostgreSQL database. If you want to use PostgreSQL as an external database, the following information can help you discern if this option is right for your Satellite configuration. Satellite supports PostgreSQL version 9.2.

Advantages of External PostgreSQL:

- Increase in free memory and free CPU on Satellite

- Flexibility to set **shared_buffers** on the PostgreSQL database to a high number without the risk of interfering with other services on Satellite

- Flexibility to tune the PostgreSQL server's system without adversely affecting Satellite operations

Disadvantages of External PostgreSQL

- Increase in deployment complexity that can make troubleshooting more difficult

- The external PostgreSQL server is an additional system to patch and maintain

- If either Satellite or the PostgreSQL database server suffers a hardware or storage failure, Satellite is not operational

- If there is latency between the Satellite server and database server, performance can suffer

If you suspect that the PostgreSQL database on your Satellite is causing performance problems, use the information in Satellite 6: How to enable postgres query logging to detect slow running queries to determine if you have slow queries. Queries that take longer than one second are typically caused by performance issues with large installations, and moving to an external database might not help. If you have slow queries, contact Red Hat Support.

3.11.3. Overview

To create and use a remote database for Satellite, you must complete the following procedures:

1. Use Section 1.2, "Storage Requirements and Guidelines" to plan the storage requirements for your external databases

2. Prepare PostgreSQL with databases for Foreman and Candlepin and dedicated users owning them

3. Prepare MongoDB with user **pulp** owning the **pulp_database**

4. Follow the initial steps to install Satellite and ensure that the databases are accessible from Satellite

5. Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**

Preparing Red Hat Enterprise Linux Server 7 for Database Installation

You require a freshly provisioned system with the latest Red Hat Enterprise Linux Server 7 that meets the storage requirements from Section 1.2, "Storage Requirements and Guidelines".

Subscriptions for Red Hat Software Collections and Red Hat Enterprise Linux do not provide the correct service level agreement for using Satellite with external databases. You must also attach a Satellite subscription to the base system that you want to use for the external database.

1. Use the instructions in Attaching the Satellite Infrastructure Subscription to attach a Satellite subscription to your server.

2. To install MongoDB and PostgreSQL servers on Red Hat Enterprise Linux Server 7, you must disable all repositories and enable only the following repositories:

```
# subscription-manager repos --disable '*'
# subscription-manager repos --enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-rpms
```

3.11.4. Installing MongoDB

You can install only the same version of MongoDB that is installed with the **satellite-installer** tool during an internal database installation. You can install MongoDB using Red Hat Software Collections (RHSCL) repositories or from an external source, as long as the version is supported. Satellite supports MongoDB version 3.4.

1. To install MongoDB, enter the following command:

   ```
   # yum install rh-mongodb34 rh-mongodb34-syspaths
   ```

2. Start and enable the **rh-mongodb34** service:

   ```
   # systemctl start rh-mongodb34-mongod
   # systemctl enable rh-mongodb34-mongod
   ```

3. Create a Pulp user on MongoDB for database **pulp_database**:

   ```
   # mongo pulp_database \
   --eval "db.createUser({user:'pulp',pwd:'pulp_password',roles:[{role:'dbOwner',
   db:'pulp_database'},{ role: 'readWrite', db: 'pulp_database'}]})"
   ```

4. In the **/etc/opt/rh/rh-mongodb34/mongod.conf** file, specify the bind IP:

   ```
   bindIp: your_mongodb_server_bind_IP,::1
   ```

5. Edit the **/etc/opt/rh/rh-mongodb34/mongod.conf** file to enable authentication in the **security** section:

   ```
   security:
     authorization: enabled
   ```

6. Restart the **rh-mongodb34-mongod** service:

   ```
   # systemctl restart rh-mongodb34-mongod
   ```

7. Open port 27017 for MongoDB:

   ```
   # firewall-cmd --add-port=27017/tcp
   # firewall-cmd --runtime-to-permanent
   ```

8. From Satellite Server, test that you can access the database. If the connection succeeds, the command returns **1**.

   ```
   # scl enable rh-mongodb34 " mongo --host mongo.example.com \
   -u pulp -p pulp_password --port 27017 --eval 'ping:1' pulp_database"
   ```

3.11.5. Installing PostgreSQL

You can install only the same version of PostgreSQL that is installed with the **satellite-installer** tool during an internal database installation. You can install PostgreSQL using Red Hat Enterprise Linux Server 7 repositories or from an external source, as long as the version is supported. Satellite supports PostgreSQL version 9.2.

1. To install PostgreSQL, enter the following command:

   ```
   # yum install postgresql-server
   ```

2. To initialize, start, and enable PostgreSQL service, enter the following commands:

   ```
   # postgresql-setup initdb
   # systemctl start postgresql
   # systemctl enable postgresql
   ```

3. Edit the **/var/lib/pgsql/data/postgresql.conf** file:

   ```
   # vi /var/lib/pgsql/data/postgresql.conf
   ```

4. Remove the **#** and edit to listen to inbound connections:

   ```
   listen_addresses = '*'
   ```

5. Edit the **/var/lib/pgsql/data/pg_hba.conf** file:

   ```
   # vi /var/lib/pgsql/data/pg_hba.conf
   ```

6. Add the following line to the file:

   ```
   host  all   all   satellite_server_ip/24   md5
   ```

7. Restart PostgreSQL service to update with the changes:

   ```
   # systemctl restart postgresql
   ```

8. Open the **postgresql** port on the external PostgreSQL server:

   ```
   # firewall-cmd --add-service=postgresql
   # firewall-cmd --runtime-to-permanent
   ```

9. Switch to the **postgres** user and start the PostgreSQL client:

   ```
   $ su - postgres -c psql
   ```

10. Create two databases and dedicated roles, one for Satellite and one for Candlepin:

    ```
    CREATE USER "foreman" WITH PASSWORD 'Foreman_Password';
    CREATE USER "candlepin" WITH PASSWORD 'Candlepin_Password';
    CREATE DATABASE foreman OWNER foreman;
    CREATE DATABASE candlepin OWNER candlepin;
    ```

11. Exit the **postgres** user:

```
# \q
```

12. From Satellite Server, test that you can access the database. If the connection succeeds, the commands return **1**.

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com -p 5432 -U foreman
-d foreman -c "SELECT 1 as ping"
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U
candlepin -d candlepin -c "SELECT 1 as ping"
```

13. To install and configure the remote database for Satellite, enter the following command:

```
satellite-installer --scenario satellite \
  --foreman-db-host postgres.example.com \
  --foreman-db-password Foreman_Password \
  --foreman-db-database foreman \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password Candlepin_Password \
  --katello-candlepin-manage-db false \
  --katello-pulp-db-username pulp \
  --katello-pulp-db-password pulp_password \
  --katello-pulp-db-seeds mongo.example.com:27017 \
  --katello-pulp-db-name pulp_database
```

You can query the status of your databases. For example, enter the following command with the **--only** and add **postgresql** or **rh-mongodb34-mongod**:

For PostgreSQL, enter the following command:

```
# satellite-maintain service status --only postgresql
```

For MongoDB, enter the following command:

```
# satellite-maintain service status --only rh-mongodb34-mongod
```

3.12. RESTRICTING ACCESS TO MONGOD

To reduce the risk of data loss, configure only the **apache** and **root** users to have access to the MongoDB database daemon, **mongod**.

To restrict access to **mongod** on your Satellite Server, you must update your firewall configuration.

Procedure

1. Update the firewall configuration by entering the following command:

```
# 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. Make the changes persistent:

```
# firewall-cmd --runtime-to-permanent
```

3.13. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES

If your Satellite deployment includes more than 5000 hosts, you can use predefined tuning profiles to improve performance of Satellite Server. Choose one of the profiles depending on the number of hosts your Satellite manages and available hardware resources. The tuning profiles are available in the satellite-support GitHub repository.

MEDIUM

Number of managed hosts: 5001–10000
RAM: 32G

Number of CPU cores: 8

LARGE

Number of managed hosts: 10001–20000
RAM: 64G

Number of CPU cores: 16

X-LARGE

Number of managed hosts: 20001–60000
RAM: 128G

Number of CPU cores: 32

2X-LARGE

Number of managed hosts: 60000+
RAM: 256G

Number of CPU cores: 48+

Procedure

To configure a tuning profile for your Satellite deployment, complete the following steps:

1. On Satellite Server, back up the **/etc/foreman-installer/custom-hiera.yaml** file to **custom-hiera.original**:

   ```
   # mv /etc/foreman-installer/custom-hiera.yaml \
   /etc/foreman-installer/custom-hiera.original
   ```

2. In your browser, navigate to tuning profiles in the *satellite-support GitHub repository* and click the profile that matches your Satellite deployment.

3. Click the **Raw** button and preserve the URL to download the profile in the next step.

4. Download the tuning profile, for example, using **curl** with the URL of the profile in the raw format:

   ```
   # curl -O https://raw.githubusercontent.com/RedHatSatellite/satellite-support/master/tuning-profiles/custom-hiera-large-64G.yaml
   ```

5. Copy the tuning profile that matches your Satellite deployment to **/etc/foreman-installer/custom-hiera.yaml**:

   ```
   # cp custom-hiera-large-64G.yaml /etc/foreman-installer/custom-hiera.yaml
   ```

6. Optional: If you added custom settings to **custom-hiera.yaml**, copy them from the backed up to the new **custom-hiera.yaml** file:

7. Optional: If you upgrade your Satellite, search the backed up **/etc/foreman-installer/custom-hiera.original** file for the strings in the following format and copy them to the new **/etc/foreman-installer/custom-hiera.yaml** file:

   ```
   # Added by foreman-installer during upgrade, run the installer with --upgrade-mongo-storage
   to upgrade to WiredTiger.
   mongodb::server::storage_engine: 'mmapv1'
   ```

8. Enter the **satellite-installer** command with no arguments to apply settings:

   ```
   # satellite-installer
   ```

CHAPTER 4. 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 Disabling DNS, DHCP, and TFTP for Unmanaged Networks for more information.

4.1. 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 file for a domain.
   The following example configures a domain **virtual.lan** as one subnet 192.168.38.0/24, a security key named **capsule**, and sets forwarders to Google's public DNS addresses (8.8.8.8 and 8.8.4.4). 192.168.38.2 is the IP address of a DNS server and 192.168.38.1 is the IP address of a Satellite Server or a Capsule Server.

   ```
   # cat /etc/named.conf
   include "/etc/rndc.key";

   controls  {
      inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "capsule"; };
      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.

4.2. VERIFYING AND STARTING THE DNS SERVICE

1. Validate the syntax.

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

2. Start the server.

   ```
   # 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).

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

4.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP

To configure Satellite Server with external DHCP, you must complete the following procedures:

1. Section 4.3.1, "Configuring an External DHCP Server to Use with Satellite Server"

2. Section 4.3.2, "Configuring Satellite Server with an External DHCP Server"

4.3.1. Configuring an External DHCP Server to Use with Satellite Server

To configure an external DHCP server to use with Satellite Server, on a Red Hat Enterprise Linux server, you must install the ISC DHCP Service and Berkeley Internet Name Domain (BIND) packages. You must also share the DHCP configuration and lease files with Satellite Server. The example in this procedure

uses the distributed Network File System (NFS) protocol to share the DHCP configuration and lease files.

Procedure

To configure an external DHCP server to use with Satellite Server, complete the following steps:

1. On a Red Hat Enterprise Linux Server server, install the ISC DHCP Service and Berkeley Internet Name Domain (BIND) packages:

   ```
   # yum install dhcp bind
   ```

2. Generate a security token:

   ```
   # dnssec-keygen -a HMAC-MD5 -b 512 -n HOST omapi_key
   ```

   As a result, a key pair that consists of two files is created 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. The following is an example:

   ```
   # 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;
   ```

   Note that the **option routers** value is the Satellite or Capsule IP address that you want to use with an external DHCP service.

5. Delete the two key files from the directory that they were created in.

6. On Satellite Server, define each subnet. Do not set DHCP Capsule for the defined Subnet yet. To prevent conflicts, set up the lease and reservation ranges separately. For example, if the lease range is 192.168.38.10 to 192.168.38.100, in the Satellite web UI define the reservation range as 192.168.38.101 to 192.168.38.250.

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

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

8. On Satellite Server, determine the UID and GID of the **foreman** user:

```
# id -u foreman
993
# id -g foreman
990
```

9. On the DHCP server, create the **foreman** user and group with the same IDs as determined in a previous step:

```
# groupadd -g 990 foreman
# useradd -u 993 -g 990 -s /sbin/nologin foreman
```

10. To ensure that the configuration files are accessible, 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
```

11. Start the DHCP service:

```
# systemctl start dhcpd
```

12. Export the DHCP configuration and lease files using NFS:

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

13. Create directories for the DHCP configuration and lease files that you want to export using NFS:

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

14. To create mount points for the created directories, add the following line to the **/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
```

15. Mount the file systems in **/etc/fstab**:

```
# mount -a
```

16. 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)

Note that the IP address that you enter is the Satellite or Capsule IP address that you want to use with an external DHCP service.

17. Reload the NFS server:

    ```
    # exportfs -rva
    ```

18. Configure the firewall for the DHCP omapi port 7911:

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

19. Optional: Configure the firewall for external access to NFS. Clients are configured using NFSv3.

    - Use the **firewalld** 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 --runtime-to-permanent
        ```

4.3.2. Configuring Satellite Server with an External DHCP Server

You can configure Satellite Server with an external DHCP server.

Prerequisite

- Ensure that you have configured an external DHCP server and that you have shared the DHCP configuration and lease files with Capsule Server. For more information, see Section 4.3.1, "Configuring an External DHCP Server to Use with Satellite Server".

Procedure

To configure Satellite Server with external DHCP, complete the following steps:

1. Install the **nfs-utils** utility:

    ```
    # 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 the Remote Procedure Call (RPC) communication paths:

```
# showmount -e DHCP_Server_FQDN
# rpcinfo -p DHCP_Server_FQDN
```

5. Add the following lines to the **/etc/fstab** file:

```
DHCP_Server_FQDN:/exports/etc/dhcp /mnt/nfs/etc/dhcp nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcp_etc_t:s0" 0 0

DHCP_Server_FQDN:/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. To verify that the **foreman-proxy** user can access the files that are shared over the network, display the DHCP configuration and lease 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. Enter the **satellite-installer** command 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=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5Yl3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=DHCP_Server_FQDN
```

9. Restart the foreman-proxy service:

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

10. Log in to the Satellite Server web UI.

11. Navigate to **Infrastructure** > **Capsules**. Locate the Capsule Server that you want to configure with external DHCP, and from the list in the **Actions** column, select **Refresh**.

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

4.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

Use this procedure to configure your Satellite Server with external TFTP services.

You can use TFTP services through NAT, for more information see Using TFTP services through NAT in the *Provisioning* guide.

Before You Begin

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

Configure Satellite Server with External TFTP

1. Install and enable the TFTP server.

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

2. Enable and activate the **tftp.socket** unit.

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

3. Configure the PXELinux environment.

   ```
   # mkdir -p /var/lib/tftpboot/{boot,pxelinux.cfg,grub2}
   # cp /usr/share/syslinux/{pxelinux.0,menu.c32,chain.c32} \
   /var/lib/tftpboot/
   ```

4. Restore SELinux file contexts.

   ```
   # restorecon -RvF /var/lib/tftpboot/
   ```

5. Create the TFTP directory to be exported using NFS.

   ```
   # mkdir -p /exports/var/lib/tftpboot
   ```

6. Add the newly created mount point to the /etc/fstab file.

   ```
   /var/lib/tftpboot /exports/var/lib/tftpboot none bind,auto 0 0
   ```

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

   ```
   # mount -a
   ```

8. 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.

9. Reload the NFS server.

```
# exportfs -rva
```

4.4.1. Configuring the Firewall for External Access to TFTP

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

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

4.5. CONFIGURING SATELLITE OR CAPSULE 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 automates the process with 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 4.5.1, "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 4.5.2, "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 using the Realm enrollment feature of Satellite, where provisioned hosts are enrolled automatically to IdM, then the **ipa-client-install** script creates DNS records for the client. The following procedure and Realm enrollment are therefore mutually exclusive. For more information on configuring Realm enrollment, see External Authentication for Provisioned Hosts in *Administering Red Hat Satellite* .

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.

4.5.1. 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. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.

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. Make a backup of the answer file in case you have to revert the changes. See Specifying Installation Options 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 **ipa-client** package on Satellite Server or Capsule Server:

   - On Satellite Server, enter the following command:

     > # satellite-maintain packages install ipa-client

   - On Capsule Server, enter the following command:

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

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

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

Configure the Satellite or Capsule Server 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.

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

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.

4.5.2. 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. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.

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. Make a backup of the answer file in case you have to revert the changes. See Specifying Installation Options 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. Reload **named** to make the changes take effect.

   ```
   # systemctl reload named
   ```

3. 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.

4. Copy the **/etc/rndc.key** file from the IdM server to Satellite's base system as follows.

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

5. Ensure that the ownership, permissions, and SELinux context are correct.

   ```
   # restorecon -v /etc/rndc.key
   # chown -v root:named /etc/rndc.key
   # chmod -v 640 /etc/rndc.key
   ```

6. 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 fails and returns the SERVFAIL error message if the record was successfully deleted.

4.5.3. Reverting to Internal DNS Service

To revert to using Satellite Server and Capsule Server as DNS providers, follow this procedure.

On the Satellite or Capsule Server that is to manage DNS for the domain.

- If you backed up the answer file before the change to external DNS, restore the answer file and then run the installation script:

  ```
  # satellite-installer
  ```

- If you do not have a suitable backup of the answer file, back up the answer file now, and then run the installation script on Satellite and Capsules as described below.
  See Specifying Installation Options for more information on the answer file.

To configure Satellite or Capsule as DNS server without using an answer file.

```
# satellite-installer \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1" \
--foreman-proxy-dns-tsig-principal="foremanproxy/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab
```

See Configuring DNS, DHCP, and TFTP on Capsule Server for more information.

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 **DHCP** or **Internal DB**.

   c. On the **Domains** tab, ensure the domain to be managed by the Satellite or Capsule 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.

CHAPTER 5. UNINSTALLING SATELLITE SERVER

If you no longer need Satellite Server or Capsule Server, you can uninstall them.

Uninstalling Satellite 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 **katello-remove** script issues two warnings, requiring confirmation before removing all packages and configuration files in the system.

> **WARNING**
>
> This script erases many packages and config files, such as the following important packages:
>
> - httpd (apache)
>
> - mongodb
>
> - tomcat6
>
> - puppet
>
> - ruby
>
> - rubygems
>
> - All Katello and Foreman Packages

Uninstall Satellite Server

1. Uninstall Satellite Server.

   ```
   # katello-remove
   ```

APPENDIX A. 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 the installer flags **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false**. If these flags are not specified during the initial installer run, rerunning of the installer overwrites all manual changes, for example, rerun for upgrade purposes. If changes are overwritten, you must run the restore procedure to restore the manual changes. For more information, see How to Restore Manual Changes Overwritten by a Puppet Run in *Installing Satellite Server from a Connected Network* .

To view all installer flags available for custom configuration, run **satellite-installer --scenario satellite --full-help**. Some Puppet classes are not exposed to the Satellite installer. To manage them manually and prevent the installer from overwriting their values, specify the configuration values by adding entries to 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 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
  ```

- For Pulp, to configure the number of pulp workers:

  ```
  pulp::num_workers: 8
  ```

The Puppet modules for the Satellite installer are stored under **/usr/share/foreman-installer/modules**. Check the **.pp** files (for example: *moduleName*/manifests/*example*.pp) to look up the classes, parameters, and values. Alternatively, use the **grep** command to do keyword searches.

Setting some values may have unintended consequences that affect the performance or functionality of Red Hat Satellite. Consider the impact of the changes before you apply them, and test the changes in a non-production environment first. If you do not have a non-production Satellite environment, run the Satellite installer with the **--noop** and **--verbose** options. If your changes cause problems, remove the offending lines from **custom-hiera.yaml** and rerun the Satellite installer. If you have any specific questions about whether a particular value is safe to alter, contact Red Hat support.

A.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.