Red Hat Satellite 6.15

Installing Satellite Server in a disconnected network environment

Install and configure Satellite Server in a network without Internet access

Red Hat Satellite 6.15 Installing Satellite Server in a disconnected network environment

Install and configure Satellite Server in a network without Internet access

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

Legal Notice

Abstract

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

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Use the **Create Issue** form in Red Hat Jira to provide your feedback. The Jira issue is created in the Red Hat Satellite Jira project, where you can track its progress.

Prerequisites

- Ensure you have registered a Red Hat account .

Procedure

1. Click the following link: Create Issue. If Jira displays a login error, log in and proceed after you are redirected to the form.

2. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.

3. Click **Create**.

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

Before you install Satellite, ensure that your environment meets the following requirements.

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base operating system:

- x86_64 architecture

- The latest version of Red Hat Enterprise Linux 8

- 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

- Full forward and reverse DNS resolution using a fully-qualified domain name

Satellite only supports **UTF-8** encoding. If your territory is USA and your language is English, set **en_US.utf-8** as the system-wide locale settings. For more information about configuring system locale in Red Hat Enterprise Linux, see Configuring System Locale guide.

Your Satellite must have the Red Hat Satellite Infrastructure Subscription manifest in your Customer Portal. Satellite must have satellite-capsule-6.x repository enabled and synced. To create, manage, and export a Red Hat Subscription Manifest in the Customer Portal, see Creating and managing manifests for a connected Satellite Server in *Subscription Central*.

Satellite Server and Capsule Server do not support shortnames in the hostnames. When using custom certificates, the Common Name (CN) of the custom certificate must be a fully qualified domain name (FQDN) instead of a shortname. This does not apply to the clients of a Satellite.

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:

- apache

- foreman

- foreman-proxy

- postgres

- pulp

- puppet

- redis

- tomcat

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 Certified Guest Operating Systems in Red Hat OpenStack Platform, Red Hat Virtualization, Red Hat OpenShift Virtualization and Red Hat Enterprise Linux with KVM.

SELinux mode

SELinux must be enabled, either in enforcing or permissive mode. Installation with disabled SELinux is not supported.

FIPS mode

You can install Satellite on a Red Hat Enterprise Linux system that is operating in FIPS mode. You cannot enable FIPS mode after the installation of Satellite. For more information, see Installing a RHEL 8 system with FIPS mode enabled in *Security hardening*.
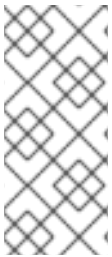
> **NOTE**
>
> Satellite supports DEFAULT and FIPS crypto-policies. The FUTURE crypto-policy is not supported for Satellite and Capsule installations. The FUTURE policy is a stricter forward-looking security level intended for testing a possible future policy. For more information, see Using system-wide cryptographic policies in the Red Hat Enterprise Linux guide.

Inter-Satellite Synchronization (ISS)

In a scenario with air-gapped Satellite Servers, all your Satellite Servers must be on the same Satellite version for ISS Export Sync to work. ISS Network Sync works across all Satellite versions that support it. For more information, see Synchronizing Content Between Satellite Servers in *Managing content*.

1.2. 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 Satellite Server installation

Directory	Installation Size	Runtime Size
/var/log	10 MB	10 GB
/var/lib/pgsql	100 MB	20 GB
/usr	10 GB	Not Applicable

Directory	Installation Size	Runtime Size
/opt/puppetlabs	500 MB	Not Applicable
/var/lib/pulp	1 MB	300 GB

For external database servers: **/var/lib/pgsql** with installation size of 100 MB and runtime size of 20 GB.

For detailed information on partitioning and size, see Partitioning reference in the *Red Hat Enterprise Linux 8 System Design Guide*.

1.3. STORAGE GUIDELINES

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

- If you mount the **/tmp** directory as a separate file system, you must use the **exec** mount option in the /**etc**/**fstab** file. If /**tmp** is already mounted with the **noexec** option, you must change the option to **exec** and re-mount the file system. This is a requirement for the **puppetserver** service to work.

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

- Use high-bandwidth, low-latency storage for the **/var/lib/pulp/** 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 **storage-benchmark** script to get this data. For more information on using the **storage-benchmark** script, see Impact of Disk Speed on Satellite Operations .

File system guidelines

- 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 How to use logrotate utility to rotate log files .

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

SELinux considerations for NFS mount

When the **/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  nfs  context="system_u:object_r:var_lib_t:s0"  1 2
```

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

```
# restorecon -R /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/pulp/** directory. These end points are not manually configurable. Ensure that storage is available on the **/var** file system to prevent storage problems.

Symbolic links

You cannot use symbolic links for /**var**/**lib**/**pulp**/.

1.4. SUPPORTED OPERATING SYSTEMS

You can install the operating system from a disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Satellite Server is supported on the latest version of Red Hat Enterprise Linux 8 that is available at the time when Satellite Server is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

The following operating systems are supported by the installer, have packages, and are tested for deploying Satellite:

Table 1.2. Operating systems supported by satellite-installer

Operating System	Architecture	Notes
Red Hat Enterprise Linux 8	x86_64 only	

Red Hat advises against using an existing system because the Satellite installer will affect the configuration of several components. 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.

Red Hat does not support using the system for anything other than running Satellite Server.

1.5. 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.

1.6. PORT AND FIREWALL 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 operating 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 and an illustration of port connections, see Capsule Networking in *Overview, concepts, and deployment considerations* .

Required ports can change based on your configuration.

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

Table 1.3. Satellite Server incoming traffic

Destination Port	Protocol	Service	Source	Required For	Description
53	TCP and UDP	DNS	DNS Servers and clients	Name resolution	DNS (optional)
67	UDP	DHCP	Client	Dynamic IP	DHCP (optional)
69	UDP	TFTP	Client	TFTP Server (optional)	
443	TCP	HTTPS	Capsule	Red Hat Satellite API	Communication from Capsule
443, 80	TCP	HTTPS, HTTP	Client	Global Registration	Registering hosts to Satellite Port 443 is required for registration initiation, uploading facts, and sending installed packages and traces Port 80 notifies Satellite on the **/unattended/built** endpoint that registration has finished

443	TCP	HTTPS	Red Hat Satellite	Content Mirroring	Management
443	TCP	HTTPS	Red Hat Satellite	Capsule API	Smart Proxy functionality
443, 80	TCP	HTTPS, HTTP	Capsule	Content Retrieval	Content
443, 80	TCP	HTTPS, HTTP	Client	Content Retrieval	Content
1883	TCP	MQTT	Client	Pull based REX (optional)	Content hosts for REX job notification (optional)
5910 – 5930	TCP	HTTPS	Browsers	Compute Resource's virtual console	
8000	TCP	HTTP	Client	Provisioning templates	Template retrieval for client installers, iPXE or UEFI HTTP Boot
8000	TCP	HTTPS	Client	PXE Boot	Installation
8140	TCP	HTTPS	Client	Puppet agent	Client updates (optional)
9090	TCP	HTTPS	Red Hat Satellite	Capsule API	Smart Proxy functionality
9090	TCP	HTTPS	Client	OpenSCAP	Configure Client (if the OpenSCAP plugin is installed)
9090	TCP	HTTPS	Discovered Node	Discovery	Host discovery and provisioning (if the discovery plugin is installed)

Any 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 operating system on which a Capsule Server is running.

A DHCP Capsule performs ICMP ping or TCP echo connection attempts to hosts in subnets with DHCP IPAM set to find out if an IP address considered for use is free. This behavior can be turned off using **satellite-installer --foreman-proxy-dhcp-ping-free-ip=false**.

NOTE

Some outgoing traffic returns to Satellite to enable internal communication and security operations.

Table 1.4. Satellite Server outgoing traffic

Destination Port	Protocol	Service	Destination	Required For	Description
	ICMP	ping	Client	DHCP	Free IP checking (optional)
7	TCP	echo	Client	DHCP	Free IP checking (optional)
22	TCP	SSH	Target host	Remote execution	Run jobs
22, 16514	TCP	SSH SSH/TLS	Compute Resource	Satellite originated communications, for compute resources in libvirt	
53	TCP and UDP	DNS	DNS Servers on the Internet	DNS Server	Resolve DNS records (optional)
53	TCP and UDP	DNS	DNS Server	Capsule DNS	Validation of DNS conflicts (optional)
53	TCP and UDP	DNS	DNS Server	Orchestration	Validation of DNS conflicts
68	UDP	DHCP	Client	Dynamic IP	DHCP (optional)
80	TCP	HTTP	Remote repository	Content Sync	Remote repositories
389, 636	TCP	LDAP, LDAPS	External LDAP Server	LDAP	LDAP authentication, necessary only if external authentication is enabled. The port can be customized when **LDAPAuthSource** is defined

Destination Port	Protocol	Service	Destination	Required For	Description
443	TCP	HTTPS	Satellite	Capsule	Capsule Configuration management Template retrieval OpenSCAP Remote Execution result upload
443	TCP	HTTPS	Amazon EC2, Azure, Google GCE	Compute resources	Virtual machine interactions (query/create/destroy) (optional)
443	TCP	HTTPS	Capsule	Content mirroring	Initiation
443	TCP	HTTPS	Infoblox DHCP Server	DHCP management	When using Infoblox for DHCP, management of the DHCP leases (optional)
623			Client	Power management	BMC On/Off/Cycle/Status
5000	TCP	HTTPS	OpenStack Compute Resource	Compute resources	Virtual machine interactions (query/create/destroy) (optional)
5900 – 5930	TCP	SSL/TLS	Hypervisor	noVNC console	Launch noVNC console

Destination Port	Protocol	Service	Destination	Required For	Description
7911	TCP	DHCP, OMAPI	DHCP Server	DHCP	The DHCP target is configured using **--foreman-proxy-dhcp-server** and defaults to localhost ISC and **remote_isc** use a configurable port that defaults to 7911 and uses OMAPI
8443	TCP	HTTPS	Client	Discovery	Capsule sends reboot command to the discovered host (optional)
9090	TCP	HTTPS	Capsule	Capsule API	Management of Capsules

1.7. 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 procedure to configure the host-based firewall on the 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 Port and firewall requirements in *Installing Satellite Server in a connected network environment.*

Procedure

1. Open the ports for clients on Satellite Server:

```
# firewall-cmd \
--add-port="5647/tcp" \
--add-port="8000/tcp" \
--add-port="9090/tcp"
```

2. Allow access to services on Satellite Server:

```
# firewall-cmd \
--add-service=dns \
--add-service=dhcp \
--add-service=tftp \
```

```
--add-service=http \
--add-service=https \
--add-service=puppetmaster
```

3. Make the changes persistent:

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

Verification

- Enter the following command:

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

For more information, see Using and Configuring firewalld in *Red Hat Enterprise Linux 8 Securing networks*.

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.

Procedure

1. 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
```

2. 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 Changing a hostname using hostnamectl in the *Red Hat Enterprise Linux 8 Configuring and managing networking*.

> ⚠️ **WARNING**
>
> Name resolution is critical to the operation of Satellite. If Satellite cannot properly resolve its fully qualified domain name, tasks such as content management, subscription management, and provisioning will fail.

1.9. 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.

Note that you cannot use tuning profiles on Capsules.

You can 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 **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes** directory.

When you run the **satellite-installer** command with the **--tuning** option, deployment configuration settings are applied to Satellite in the following order:

1. The default tuning profile defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** file

2. The tuning profile that you want to apply to your deployment and is defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** directory

3. Optional: If you have configured a **/etc/foreman-installer/custom-hiera.yaml** file, Satellite applies these configuration settings.

Note that the configuration settings that are defined in the **/etc/foreman-installer/custom-hiera.yaml** file override the configuration settings that are defined in the tuning profiles.

Therefore, before applying a tuning profile, you must compare the configuration settings that are defined in the default tuning profile in **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml**, the tuning profile that you want to apply and your **/etc/foreman-installer/custom-hiera.yaml** file, and remove any duplicated configuration from the **/etc/foreman-installer/custom-hiera.yaml** file.

default

Number of hosts: 0 – 5000
RAM: 20G

Number of CPU cores: 4

medium

Number of hosts: 5001 – 10000

RAM: 32G

Number of CPU cores: 8

large

Number of hosts: 10001 – 20000
RAM: 64G

Number of CPU cores: 16

extra-large

Number of hosts: 20001 – 60000
RAM: 128G

Number of CPU cores: 32

extra-extra-large

Number of hosts: 60000+
RAM: 256G

Number of CPU cores: 48+

Procedure

1. Optional: If you have configured the **custom-hiera.yaml** file on Satellite Server, back up the **/etc/foreman-installer/custom-hiera.yaml** file to **custom-hiera.original**. You can use the backup file to restore the **/etc/foreman-installer/custom-hiera.yaml** file to its original state if it becomes corrupted:

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

2. Optional: If you have configured the **custom-hiera.yaml** file on Satellite Server, review the definitions of the default tuning profile in **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** and the tuning profile that you want to apply in **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/**. Compare the configuration entries against the entries in your **/etc/foreman-installer/custom-hiera.yaml** file and remove any duplicated configuration settings in your **/etc/foreman-installer/custom-hiera.yaml** file.

3. Enter the **satellite-installer** command with the **--tuning** option for the profile that you want to apply. For example, to apply the medium tuning profile settings, enter the following command:

   ```
   # satellite-installer --tuning medium
   ```

CHAPTER 2. INSTALLING SATELLITE SERVER

When the intended host for Satellite Server is in a disconnected environment, you can install Satellite Server by using an external computer to download an ISO image of the packages, and copying the packages to the system you want to install Satellite Server on. This method is not recommended for any other situation as ISO images might not contain the latest updates, bug fixes, and functionality.

Use the following procedures to install Satellite Server, perform the initial configuration, and import subscription manifests.

Before you continue, consider which manifests are relevant for your environment. For more information on manifests, see Managing Red Hat Subscriptions in *Managing content*.

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

2.1. DOWNLOADING THE BINARY DVD IMAGES

Use this procedure to download the ISO images for Red Hat Enterprise Linux and Red Hat Satellite.

Procedure

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 for x86_64**

   - **Version** is set to the latest minor version of the product you plan to use as the base operating 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 for x86_64** 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.

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 operating system or other accessible storage device.

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

2.2. CONFIGURING THE BASE OPERATING SYSTEM WITH OFFLINE REPOSITORIES IN RHEL 8

Use this procedure to configure offline repositories for the Red Hat Enterprise Linux 8 and Red Hat Satellite ISO images.

Procedure

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

   ```
   # mkdir /media/rhel8
   ```

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

   ```
   # mount -o loop rhel8-DVD.iso /media/rhel8
   ```

3. To copy the ISO file's repository data file and change permissions, enter:

   ```
   # cp /media/rhel8/media.repo /etc/yum.repos.d/rhel8.repo
   # chmod u+w /etc/yum.repos.d/rhel8.repo
   ```

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

   ```
   [RHEL8-BaseOS]
   name=Red Hat Enterprise Linux BaseOS
   mediaid=None
   metadata_expire=-1
   gpgcheck=0
   cost=500
   baseurl=file:///media/rhel8/BaseOS/

   [RHEL8-AppStream]
   name=Red Hat Enterprise Linux Appstream
   mediaid=None
   metadata_expire=-1
   gpgcheck=0
   cost=500
   baseurl=file:///media/rhel8/AppStream/
   ```

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 Satellite Server.

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

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

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

2.3. OPTIONAL: USING FAPOLICYD ON SATELLITE SERVER

By enabling **fapolicyd** on your Satellite Server, you can provide an additional layer of security by monitoring and controlling access to files and directories. The fapolicyd daemon uses the RPM database as a repository of trusted binaries and scripts.

You can turn on or off the fapolicyd on your Satellite Server or Capsule Server at any point.

2.3.1. Installing fapolicyd on Satellite Server

You can install **fapolicyd** along with Satellite Server or can be installed on an existing Satellite Server. If you are installing **fapolicyd** along with the new Satellite Server, the installation process will detect the fapolicyd in your Red Hat Enterprise Linux host and deploy the Satellite Server rules automatically.

Prerequisites

- Ensure your host has access to the BaseOS repositories of Red Hat Enterprise Linux.

Procedure

1. Install fapolicyd:

   ```
   # dnf install fapolicyd
   ```

2. Start the **fapolicyd** service:

   ```
   # systemctl enable --now fapolicyd
   ```

Verification

- Verify that the **fapolicyd** service is running correctly:

  ```
  # systemctl status fapolicyd
  ```

New Satellite Server or Capsule Server installations

In case of new Satellite Server or Capsule Server installation, follow the standard installation procedures after installing and enabling fapolicyd on your Red Hat Enterprise Linux host.

Additional resources

For more information on fapolicyd, see Blocking and allowing applications using fapolicyd in *Red Hat Enterprise Linux 8 Security hardening*.

2.4. INSTALLING THE SATELLITE PACKAGES FROM THE OFFLINE REPOSITORIES

Use this procedure to install the Satellite packages from the offline repositories.

Procedure

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 operating system is up to date with the Binary DVD image:

```
# dnf 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
```

> **NOTE**
>
> The script contains a command that enables the **satellite:el8** module.
> Enablement of the module **satellite:el8** warns about a conflict with
> **postgresql:10** and **ruby:2.5** as these modules are set to the default module
> versions on Red Hat Enterprise Linux 8. The module **satellite:el8** has a
> dependency for the modules **postgresql:12** and **ruby:2.7** that will be enabled
> with the **satellite:el8** module. These warnings do not cause installation process
> failure, hence can be ignored safely. For more information about modules and
> lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux
> Application Streams Lifecycle.

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

2.5. 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 operating system.

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

    ```
    # cd /path-to-package/
    ```

12. Install the package locally:

    ```
    # dnf install package_name
    ```

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

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

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

    ```
    # ./install_packages
    ```

> **NOTE**
>
> The script contains a command that enables the **satellite:el8** module. Enablement of the module **satellite:el8** warns about a conflict with **postgresql:10** and **ruby:2.5** as these modules are set to the default module versions on Red Hat Enterprise Linux 8. The module **satellite:el8** has a dependency for the modules **postgresql:12** and **ruby:2.7** that will be enabled with the **satellite:el8** module. These warnings do not cause installation process failure, hence can be ignored safely. For more information about modules and lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux Application Streams Lifecycle.

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

2.6. SYNCHRONIZING THE SYSTEM CLOCK WITH CHRONYD

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

For more information about the **chrony** suite, see Using the Chrony suite to configure NTP in *Red Hat Enterprise Linux 8 Configuring basic system settings*.

Procedure

1. Install the **chrony** package:

   ```
   # dnf install chrony
   ```

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

   ```
   # systemctl enable --now chronyd
   ```

2.7. INSTALLING THE SOS PACKAGE ON THE BASE OPERATING SYSTEM

Install the **sos** package on the base operating system so that you can 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.

Procedure

- Install the **sos** package:

  ```
  # satellite-maintain packages install sos
  ```

2.8. CONFIGURING SATELLITE SERVER

Install Satellite Server using the **satellite-installer** installation script. Choose from one of the following methods:

- Section 2.8.1, "Configuring Satellite installation". 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.

2.8.1. Configuring Satellite installation

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 PostgreSQL databases on the same server.

The installation process can take tens of minutes to complete. If you are connecting remotely to the system, use a utility such as **tmux** 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. 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

- 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. When **satellite-installer** runs, it overwrites any manual changes to the managed files with the intended values. This means that running the installer on a broken system should restore it to working order, regardless of changes made. For more information on how to apply custom configuration on other services, see Applying Custom Configuration to Satellite .

Procedure

1. Enter the following command with any additional options that you want to use:

   ```
   # satellite-installer --scenario satellite \
   --foreman-initial-organization "My_Organization" \
   --foreman-initial-location "My_Location" \
   --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**.

2. Unmount the ISO images:

   ```
   # umount /media/sat6
   # umount /media/rhel8
   ```

2.9. DISABLING SUBSCRIPTION CONNECTION

Disable subscription connection on disconnected Satellite Server to avoid connecting to the Red Hat Portal. This will also prevent you from refreshing the manifest and updating upstream entitlements.

Procedure

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

2. Click the **Content** tab.

3. Set the **Subscription Connection Enabled** value to **No**.

CLI procedure

- Enter the following command on Satellite Server:

  ```
  # hammer settings set --name subscription_connection_enabled --value false
  ```

2.10. IMPORTING A RED HAT SUBSCRIPTION MANIFEST INTO SATELLITE SERVER

Use the following procedure to import a Red Hat subscription manifest into Satellite Server.

> **NOTE**
>
> Simple Content Access (SCA) is set on the organization, not the manifest. Importing a manifest does not change your organization's Simple Content Access status.

Prerequisites

- You must have a Red Hat subscription manifest file exported from the Red Hat Customer Portal. For more information, see Creating and Managing Manifests in *Using Red Hat Subscription Management*.

- Ensure that you disable subscription connection on your Satellite Server. For more information, see Section 2.9, "Disabling subscription connection".

Procedure

1. In the Satellite web UI, ensure the context is set to the organization you want to use.

2. In the Satellite web UI, navigate to **Content** > **Subscriptions** and click **Manage Manifest**.

3. In the **Manage Manifest** window, click **Choose File**.

4. Navigate to the location that contains the Red Hat subscription manifest file, then click **Open**.

CLI procedure

1. Copy the Red Hat subscription manifest file from your local machine to Satellite Server:

   ```
   $ scp ~/manifest_file.zip root@satellite.example.com:~/.
   ```

2. Log in to Satellite Server as the **root** user and import the Red Hat subscription manifest file:

   ```
   # hammer subscription upload \
   --file ~/manifest_file.zip \
   --organization "My_Organization"
   ```

You can now enable repositories and import Red Hat content. For more information, see Importing Content in *Managing content*.

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER

3.1. CONFIGURING SATELLITE SERVER TO CONSUME CONTENT FROM A CUSTOM CDN

If you have an internal Content Delivery Network (CDN) or serve content on an accessible web server, you can configure your Satellite Server to consume Red Hat repositories from this CDN server instead of the Red Hat CDN. A CDN server can be any web server that mirrors repositories in the same directory structure as the Red Hat CDN.

You can configure the source of content for each organization. Satellite recognizes automatically which Red Hat repositories from the subscription manifest in your organization are available on your CDN server.

Prerequisites

- You have a CDN server that provides Red Hat content and is accessible by Satellite Server.

- If your CDN server uses HTTPS, ensure you have uploaded the SSL certificate into Satellite. For more information, see Importing Custom SSL Certificates in *Managing content*.

- You have uploaded a manifest to your organization.

Procedure

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

2. Click **Manage Manifest**.

3. Select the **CDN Configuration** tab.

4. Select the **Custom CDN** tab.

5. In the **URL** field, enter the URL of your CDN server from which you want Satellite Server to consume Red Hat repositories.

6. Optional: In the **SSL CA Content Credential**, select the SSL certificate of the CDN server.

7. Click **Update**.

8. You can now enable Red Hat repositories consumed from your internal CDN server.

CLI procedure

1. Connect to your Satellite Server using SSH.

2. Set CDN configuration to your custom CDN server:

```
# hammer organization configure-cdn --name="My_Organization" \
--type=custom_cdn \
--url https://my-cdn.example.com \
--ssl-ca-credential-id "My_CDN_CA_Cert_ID"
```

Additional resources

- [Content Delivery Network Structure](#) in *Overview, concepts, and deployment considerations*

3.2. HOW TO CONFIGURE INTER-SATELLITE SYNCHRONIZATION

Red Hat Satellite uses Inter-Satellite Synchronization (ISS) to synchronize content between two Satellite Servers including those that are air-gapped.

You can use ISS in cases such as:

- If you want to copy some but not all content from your Satellite Server to other Satellite Servers. For example, you have content views that your IT department consumes from Satellite Server, and you want to copy content from those content views to other Satellite Servers.

- If you want to copy all Library content from your Satellite Server to other Satellite Servers. For example, you have Products and repositories that your IT department consumes from Satellite Server in the Library, and you want to copy all Products and repositories in that organization to other Satellite Servers.

> **NOTE**
>
> You cannot use ISS to synchronize content from Satellite Server to Capsule Server. Capsule Server supports synchronization natively. For more information, see [Capsule Server Overview](#) in *Overview, concepts, and deployment considerations* .

There are different ways of using ISS. The way you can use depends on your multi-server setup that can fall to one of the following scenarios.

3.2.1. ISS network sync in a disconnected scenario

In a disconnected scenario, there is the following setup:

- The upstream Satellite Server is connected to the Internet. This server consumes content from the Red Hat Content Delivery Network (CDN) or custom sources.

- The downstream Satellite Server is completely isolated from all external networks.

- The downstream Satellite Server can communicate with a connected upstream Satellite Server over an internal network.

Figure 3.1. Satellite ISS disconnected scenario
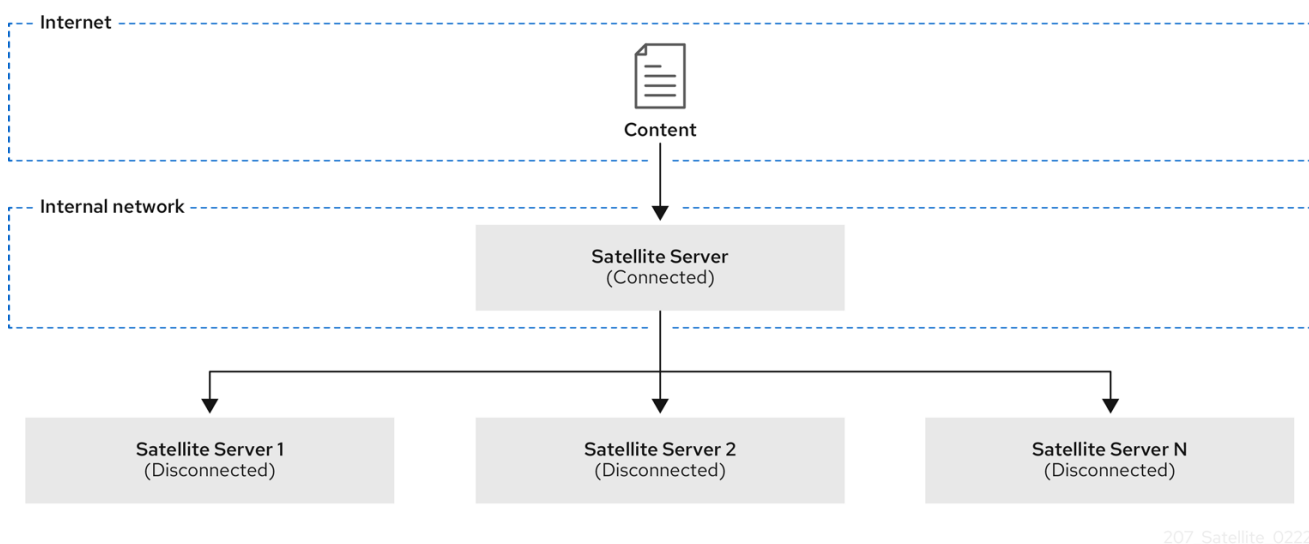
You can configure your downstream Satellite Server to synchronize content from the upstream Satellite Server over the network. See Section 3.3, "Configuring Satellite Server to synchronize content over a network".

3.2.2. ISS export sync in an air-gapped scenario

In an air-gapped scenario, there is the following setup:

- The upstream Satellite Server is connected to the Internet. This server consumes content from the Red Hat CDN or custom sources.

- The downstream Satellite Server is completely isolated from all external networks.

- The downstream Satellite Server does not have a network connection to a connected upstream Satellite Server.

Figure 3.2. Satellite ISS air-gapped scenario
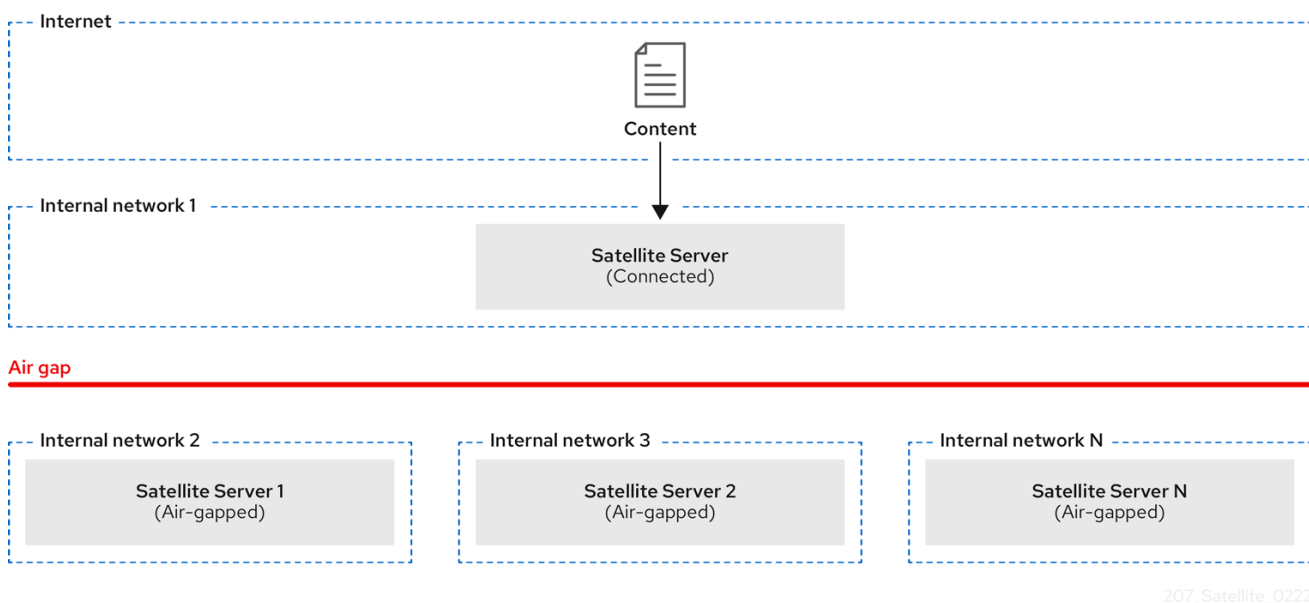
The only way for an air-gapped downstream Satellite Server to receive content updates is by exporting payload from the upstream Satellite Server, bringing it physically to the downstream Satellite Server, and importing the payload. For more information, see Synchronizing Content Between Satellite Servers in *Managing content*.

Configure your downstream Satellite Server to synchronize content using exports. See Section 3.4, "Configuring Satellite server to synchronize content using exports".

3.3. CONFIGURING SATELLITE SERVER TO SYNCHRONIZE CONTENT OVER A NETWORK

Configure a downstream Satellite Server to synchronize repositories from a connected upstream Satellite Server over HTTPS.

Prerequisites

- A network connection exists between the upstream Satellite Server and the downstream Satellite Server.

- You imported the subscription manifest on both the upstream and downstream Satellite Server.

- On the upstream Satellite Server, you enabled the required repositories for the organization. For more information, see Enabling Red Hat Repositories in *Managing content*.

- The upstream user is an admin or has the following permissions:

  - **view_organizations**

  - **view_products**

  - **export_content**

  - **view_lifecycle_environments**

  - **view_content_views**

- On the downstream Satellite Server, you have imported the SSL certificate of the upstream Satellite Server using the contents of **http://*upstream-satellite.example.com*/pub/katello-server-ca.crt**. For more information, see Importing SSL Certificates in *Managing content*.

- The downstream user is an admin or has the permissions to create product repositories and organizations.

Procedure

1. Navigate to **Content** > **Subscriptions**.

2. Click **Manage Manifest**.

3. Navigate to the **CDN Configuration** tab.

4. Select the **Network Sync** tab.

5. In the **URL** field, enter the address of the upstream Satellite Server.

6. In the **Username**, enter your username for upstream login.

7. In the **Password**, enter your password or personal access token for upstream login.

8. In the **Organization label** field, enter the label of the upstream organization.

9. Optional: In the **Lifecycle Environment Label** field, enter the label of the upstream lifecycle environment. Default is **Library**.

10. Optional: In the **Content view label** field, enter the label of the upstream content view. Default is **Default_Organization_View**.

11. From the **SSL CA Content Credential** menu, select a CA certificate used by the upstream Satellite Server.

12. Click **Update**.

13. In the Satellite web UI, navigate to **Content** > **Products**.

14. Select the product that contains the repositories that you want to synchronize.

15. From the **Select Action** menu, select **Sync Now** to synchronize all repositories within the product.
    You can also create a synchronization plan to ensure updates on a regular basis. For more information, see Creating a Synchronization Plan in *Managing content*.

CLI procedure

1. Connect to your downstream Satellite Server using SSH.

2. View information about the upstream CA certificate:

   ```
   # hammer content-credential show \
   --name="My_Upstream_CA_Cert" \
   --organization="My_Downstream_Organization"
   ```

   Note the ID of the CA certificate for the next step.

3. Set CDN configuration to an upstream Satellite Server:

   ```
   # hammer organization configure-cdn --name="My_Downstream_Organization" \
   --type=network_sync \
   --url https://upstream-satellite.example.com \
   --username upstream_username --password upstream_password \
   --ssl-ca-credential-id "My_Upstream_CA_Cert_ID" \ --upstream-organization-
   label="_My_Upstream_Organization" \
   [--upstream-lifecycle-environment-label="My_Lifecycle_Environment"] \
   [--upstream-content-view-label="My_Content_View"]
   ```

   The default lifecycle environment label is **Library**. The default content view label is **Default_Organization_View**.

3.4. CONFIGURING SATELLITE SERVER TO SYNCHRONIZE CONTENT USING EXPORTS

If you deployed your downstream Satellite Server as air-gapped, configure your Satellite Server as such to avoid attempts to consume content from a network.

Procedure

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

2. Click **Manage Manifest**.

3. Switch to the **CDN Configuration** tab.

4. Select the **Export Sync** tab.

5. Click **Update**.

CLI procedure

1. Log in to your Satellite Server using SSH.

2. Set CDN configuration to sync using exports:

   ```
   # hammer organization configure-cdn --name="My_Organization" --type=export_sync
   ```

Additional resources

For more information about synchronizing content using exports, see How to Synchronize Content Using Export and Import in *Managing content*.

3.5. 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.

To import Kickstart repositories for Red Hat Enterprise Linux 9, complete Section 3.5.1, "Importing Kickstart repositories for Red Hat Enterprise Linux 9".

To import Kickstart repositories for Red Hat Enterprise Linux 8, complete Section 3.5.2, "Importing Kickstart repositories for Red Hat Enterprise Linux 8".

To import Kickstart repositories for Red Hat Enterprise Linux 7, complete Section 3.5.3, "Importing Kickstart repositories for Red Hat Enterprise Linux 7".

3.5.1. Importing Kickstart repositories for Red Hat Enterprise Linux 9

Use this procedure to import Kickstart repositories for Red Hat Enterprise Linux 9.

Procedure

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

2. Click **Red Hat Enterprise Linux**.

3. Select a product variant and a product version from the list. For example, product variant **Red Hat Enterprise Linux for x86_64** and product version **9.0**.

4. Locate the full installation image, for example, **Red Hat Enterprise Linux 9.0 Binary DVD**, and click **Download Now**. Note that you cannot provision hosts using the minimal ISO.

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-binary-dvd.iso /mnt/iso
```

Replace *rhel–binary–dvd.iso* with the name of your ISO image.

7. Create directories for Red Hat Enterprise Linux 9 AppStream and BaseOS Kickstart repositories:

```
# mkdir --parents /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

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

```
# cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

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

9. Add the following entries to the listing files:
To the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/listing** file, append **kickstart** with a new line.

To the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/listing** file, append **kickstart** with a new line.

To the **/var/www/html/pub/satellite-import/content/dist/rhel8/listing** file, append the version number with a new line. For example, for the Red Hat Enterprise Linux 9.0 binary ISO, append **9.0**.

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

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo
```

11. Open the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo** file for editing.

12. 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**

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

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

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

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

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

16. Save and close the file.

17. Verify that the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo** file has the following format:

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b

[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 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 9.0.0

[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 = 9.0.0
```

```
[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
```

18. Open the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo** file for editing.

19. 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**

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

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

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

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

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

23. Save and close the file.

24. Verify that the **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/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 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
```

```
version = 9.0.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 9.0.0

[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
```

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

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

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

3.5.2. Importing Kickstart repositories for Red Hat Enterprise Linux 8

Use this procedure to import Kickstart repositories for Red Hat Enterprise Linux 8.

Procedure

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

2. Click **Red Hat Enterprise Linux**.

3. Select a product variant and a product version from the list. For example, product variant **Red Hat Enterprise Linux for x86_64** and product version **8.1**.

4. Locate the full installation 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-binary-dvd.iso /mnt/iso
    ```

Replace *rhel-binary-dvd.iso* with the name of your ISO image.

7. Create directories for Red Hat Enterprise Linux 8 AppStream and BaseOS Kickstart repositories:

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

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

   ```
   # cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
   # cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-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.

9. Add the following entries to the listing files:
   To the **/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/listing** file, append **kickstart** with a new line.

   To the **/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/listing** file, append **kickstart** with a new line.

   To the **/var/www/html/pub/satellite-import/content/dist/rhel8/listing** file, append the version number with a new line. For example, for the Red Hat Enterprise Linux 8.1 binary ISO, append **8.1**.

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

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

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

12. 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**

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

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

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

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

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

16. Save and close the file.

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

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b

[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
```

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

19. 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**

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

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

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

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

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

23. Save and close the file.

24. Verify that the **/var/www/html/pub/satellite-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
```

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

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

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

3.5.3. Importing Kickstart repositories for Red Hat Enterprise Linux 7

Use this procedure to import Kickstart repositories for Red Hat Enterprise Linux 7.

Procedure

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

2. Click **Red Hat Enterprise Linux**.

3. Click **Switch to version 7 and below** above the **Product Variant** list.

4. Select a product variant and a product version from the list. For example, product variant **Red Hat Enterprise Linux for x86_64** and product version **7.9**.

5. Locate the full installation image, for example, **Red Hat Enterprise Linux 7.9 Binary DVD**, and click **Download Now**.

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

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

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

Replace *rhel-binary-dvd.iso* with the name of your ISO image.

8. Create Kickstart directories:

   ```
   # mkdir --parents /var/www/html/pub/satellite-
   import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
   ```

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

   ```
   # cp -a /mnt/iso/* /var/www/html/pub/satellite-
   import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
   ```

10. Add the following entries to the listing files:
    To the **/var/www/html/pub/satellite-import/content/dist/rhel/server/7/listing** file, append the
    version number with a new line. For example, for the Red Hat Enterprise Linux 7.9 ISO, append
    7.9.

    To the **/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/listing** file, append
    the architecture with a new line. For example, **x86_64**.

    To the **/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/x86_64/listing** file,
    append **kickstart** with a new line.

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

    ```
    # cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
    import/content/dist/rhel/server/7/7.9/x86_64/kickstart/treeinfo
    ```

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

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

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

3.6. ENABLING AND SYNCHRONIZING THE SATELLITE CLIENT 6 REPOSITORY

The Satellite Client 6 repository provides the **katello-host-tools** and **puppet** packages for hosts
registered to Satellite. You must periodically synchronize the repository from the Red Hat Content
Delivery Network (CDN) to your Satellite Server and enable the repository on your hosts.

3.6.1. Synchronizing the Satellite Client 6 repository for Red Hat Enterprise Linux 9 and Red Hat Enterprise Linux 8

To use the CLI instead of the Satellite web UI, see the procedure relevant for your Red Hat Enterprise
Linux version:

- CLI procedure for Red Hat Enterprise Linux 9

- CLI procedure for Red Hat Enterprise Linux 8

Procedure

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

2. Click the arrow next to the **Red Hat Enterprise Linux for x86_64** product to view available content.

3. Select **Red Hat Satellite Client 6 for RHEL 9 x86_64 RPMs** or **Red Hat Satellite Client 6 for RHEL 8 x86_64 RPMs**.

4. Click **Synchronize Now**.

CLI procedure for Red Hat Enterprise Linux 9

- Synchronize your Satellite Client 6 repository:

```
# hammer repository synchronize \
--name "Red Hat Satellite Client 6 for RHEL 9 x86_64 RPMs" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux for x86_64"
```

CLI procedure for Red Hat Enterprise Linux 8

- Synchronize your Satellite Client 6 repository:

```
# hammer repository synchronize \
--name "Red Hat Satellite Client 6 for RHEL 8 x86_64 RPMs" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux for x86_64"
```

Additional resources

- For details about the **hammer repository synchronize** command, enter **hammer repository synchronize --help**.

3.6.2. Synchronizing the Satellite Client 6 repository for Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 6

> **NOTE**
>
> You require **Red Hat Enterprise Linux Extended Lifecycle Support (ELS) Add-on** subscription to synchronize the repositories of Red Hat Enterprise Linux 6. For more information, see Red Hat Enterprise Linux Extended Lifecycle Support (ELS) Add-on guide.

To use the CLI instead of the Satellite web UI, see the procedure relevant for your Red Hat Enterprise Linux version:

- CLI procedure for Red Hat Enterprise Linux 7

- CLI procedure for Red Hat Enterprise Linux 6

Procedure

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

2. Click the arrow next to the **Red Hat Enterprise Linux Server** or **Red Hat Enterprise Linux Server – Extended Lifecycle Support**.

3. Select **Red Hat Satellite Client 6 (for RHEL 7 Server) RPMs x86_64** or **Red Hat Satellite Client 6 for RHEL 6 Server – ELS RPMs x86_64** based on your operating system version.

4. Click **Synchronize Now**.

CLI procedure for Red Hat Enterprise Linux 7

- Synchronize your Satellite Client 6 repository:

```
# hammer repository synchronize \
--async \
--name "Red Hat Satellite Client 6 for RHEL 7 Server RPMs x86_64" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server"
```

CLI procedure for Red Hat Enterprise Linux 6

- Synchronize your Satellite Client 6 repository:

```
# hammer repository synchronize \
--async \
--name "Red Hat Satellite Client 6 for RHEL 6 Server - ELS RPMs x86_64" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server - Extended Lifecycle Support"
```

Additional resources

- For details about the **hammer repository synchronize** command, enter **hammer repository synchronize --help**.

3.6.3. Enabling the Satellite Client 6 repository for Red Hat Enterprise Linux 9 and Red Hat Enterprise Linux 8

To use the CLI instead of the Satellite web UI, see the procedure relevant for your Red Hat Enterprise Linux version:

- CLI procedure for Red Hat Enterprise Linux 9

- CLI procedure for Red Hat Enterprise Linux 8

Prerequisites

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

Procedure

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

2. In the Available Repositories pane, enable the **Recommended Repositories** to get the list of repositories.

3. Click **Red Hat Satellite Client 6 for RHEL 9 x86_64 (RPMs)**or **Red Hat Satellite Client 6 for RHEL 8 x86_64 (RPMs)** to expand the repository set.

4. For the **x86_64** architecture, click the **+** icon to enable the repository.
   If the **Satellite Client 6** items are not visible, it may be because they are not included in the Red Hat subscription manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the Red Hat subscription manifest and import it into Satellite. For more information, see Managing Red Hat Subscriptions in *Managing content*.

   Enable the Satellite Client 6 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.

CLI procedure for Red Hat Enterprise Linux 9

- Enable the Satellite Client 6 repository:

  ```
  # hammer repository-set enable \
  --basearch="x86_64" \
  --name "Red Hat Satellite Client 6 for RHEL 9 x86_64 (RPMs)" \
  --organization "My_Organization" \
  --product "Red Hat Enterprise Linux for x86_64"
  ```

CLI procedure for Red Hat Enterprise Linux 8

- Enable the Satellite Client 6 repository:

  ```
  # hammer repository-set enable \
  --basearch="x86_64" \
  --name "Red Hat Satellite Client 6 for RHEL 8 x86_64 (RPMs)" \
  --organization "My_Organization" \
  --product "Red Hat Enterprise Linux for x86_64"
  ```

Additional resources

- For details about the **hammer repository-set enable** command, enter **hammer repository-set enable --help**.

3.6.4. Enabling the Satellite Client 6 repository for Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 6

> **NOTE**
>
> You require **Red Hat Enterprise Linux Extended Lifecycle Support (ELS) Add-on** subscription to enable the repositories of Red Hat Enterprise Linux 6. For more information, see Red Hat Enterprise Linux Extended Lifecycle Support (ELS) Add-on guide.

To use the CLI instead of the Satellite web UI, see the procedure relevant for your Red Hat Enterprise Linux version:

- CLI procedure for Red Hat Enterprise Linux 7

- CLI procedure for Red Hat Enterprise Linux 6

Prerequisites

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

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

    2. In the **Available Repositories** pane, enable the **Recommended Repositories** to get the list of repositories.

    3. In the **Available Repositories** pane, click on **Satellite Client 6 (for RHEL 7 Server) (RPMs)** or **Satellite Client 6 (for RHEL 6 Server – ELS) (RPMs)** to expand the repository set. If the **Satellite Client 6** items are not visible, it may be because they are not included in the Red Hat subscription manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the Red Hat subscription manifest and import it into Satellite. For more information, see Managing Red Hat Subscriptions in *Managing content*.

    4. For the **x86_64** architecture, click the **+** icon to enable the repository. Enable the Satellite Client 6 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.

CLI procedure for Red Hat Enterprise Linux 7

- Enable the Satellite Client 6 repository:

    ```
    # hammer repository-set enable \
    --basearch="x86_64" \
    --name "Red Hat Satellite Client 6 (for RHEL 7 Server) (RPMs)" \
    --organization "My_Organization" \
    --product "Red Hat Enterprise Linux Server"
    ```

CLI procedure for Red Hat Enterprise Linux 6

- Enable the Satellite Client 6 repository:

    ```
    # hammer repository-set enable \
    --basearch="x86_64" \
    --name "Red Hat Satellite Client 6 (for RHEL 6 Server - ELS) (RPMs)" \
    --organization "My_Organization" \
    --product "Red Hat Enterprise Linux Server - Extended Lifecycle Support"
    ```

Additional resources

- For details about the **hammer repository-set enable** command, enter **hammer repository-set enable --help**.

3.7. CONFIGURING REMOTE EXECUTION FOR PULL CLIENT ON SATELLITE SERVER

By default, Remote Execution uses SSH as the transport mechanism for the Script provider. However, Remote Execution also offers pull-based transport, which you can use if your infrastructure prohibits outgoing connections from Satellite to hosts.

This comprises **pull-mqtt** mode on Satellite in combination with a pull client running on hosts.

> **NOTE**
>
> The **pull-mqtt** mode works only with the Script provider. Ansible and other providers will continue to use their default transport settings.

To use **pull-mqtt** mode on Satellite Server, follow the procedure below:

Procedure

1. Enable the pull-based transport on your Satellite Server:

   ```
   # satellite-installer --foreman-proxy-plugin-remote-execution-script-mode pull-mqtt
   ```

2. Configure the firewall to allow MQTT service on port 1883:

   ```
   # firewall-cmd --add-service=mqtt
   ```

   In **pull-mqtt** mode, hosts subscribe for job notifications to either your Satellite Server or any Capsule Server through which they are registered. Therefore, it is recommended to ensure that Satellite Server sends remote execution jobs to that same Satellite Server or Capsule Server.

3. Make the changes persistent:

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

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

5. On the **Content** tab, set the value of **Prefer registered through Capsule for remote execution** to **Yes**.

After you set up the pull-based transport on Satellite, you must also configure it on each host. For more information, see Transport Modes for Remote Execution in *Managing hosts*.

3.8. ENABLING POWER MANAGEMENT ON HOSTS

To perform power management tasks on 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 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"
  ```

3.9. CONFIGURING DNS, DHCP, AND TFTP

You can manage DNS, DHCP, and TFTP centrally within the Satellite environment, or you can manage them independently after disabling their maintenance on Satellite. You can also run DNS, DHCP, and TFTP externally, outside of the Satellite environment.

3.9.1. 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.

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

- 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 \
  --foreman-proxy-dns true \
  --foreman-proxy-dns-managed true \
  --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-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**.

Additional resources

- For more information about the **satellite-installer** command, enter **satellite-installer --help**.

3.9.2. 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

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. In the Satellite web UI, 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.
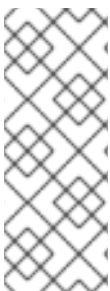
> **NOTE**
>
> Satellite does not perform orchestration when a Capsule is not set for a given subnet and domain. When enabling or disabling Capsule associations, orchestration commands for existing hosts can fail if the expected records and configuration files are not present. When associating a Capsule to turn orchestration on, ensure 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.

3.9.3. Additional resources

- For more information about configuring DNS, DHCP, and TFTP externally, see Chapter 4, *Configuring Satellite Server with external services* .

- For more information about configuring DHCP, DNS, and TFTP services, see Configuring Network Services in *Provisioning hosts*.

3.10. 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

- Some SMTP servers with anti-spam protection or grey-listing features are known to cause problems. To setup outgoing email with such a service either install and configure a vanilla SMTP service on Satellite Server for relay or use the **sendmail** command instead.

Procedure

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
SMTP username	*user@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 location	/usr/sbin/sendmail
Sendmail arguments	-i

For security reasons, both Sendmail location and Sendmail argument settings are read-only and can be only set in **/etc/foreman/settings.yaml**. Both settings currently cannot be set via **satellite-installer**. 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 Satellite 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 Satellite web UI displays an error. See the log at **/var/log/foreman/production.log** for further details.

Additional resources

- For information on configuring email notifications for individual users or user groups, see Configuring Email Notification Preferences in *Administering Red Hat Satellite* .

3.11. CONFIGURING SATELLITE SERVER WITH A CUSTOM SSL CERTIFICATE

By default, Red Hat Satellite 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 (CA).

When you configure Red Hat Satellite with custom SSL certificates, you must fulfill the following requirements:

- You must use the privacy-enhanced mail (PEM) encoding for the SSL certificates.

- You must not use the same SSL certificate for both Satellite Server and Capsule Server.

- The same CA must sign certificates for Satellite Server and Capsule Server.

- An SSL certificate must not also be a CA certificate.

- An SSL certificate must include a subject alt name (SAN) entry that matches the common name (CN).

- An SSL certificate must be allowed for Key Encipherment using a Key Usage extension.

- An SSL certificate must not have a shortname as the CN.

- You must not set a passphrase for the private key.

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

1. Section 3.11.1, "Creating a custom SSL certificate for Satellite Server"

2. Section 3.11.2, "Deploying a custom SSL certificate to Satellite Server"

3. Section 3.11.3, "Deploying a custom SSL certificate to hosts"

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

3.11.1. 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.

Procedure

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 CSR and include the following content:

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

   [ req_distinguished_name ]
   commonName = satellite.example.com

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

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

4. Optional: If you want to add Distinguished Name (DN) details to the CSR, add the following information to the **[req_distinguished_name]** section:

   ```
   [req_distinguished_name]
   CN = satellite.example.com
   countryName =My_Country_Name          1
   stateOrProvinceName = My_State_Or_Province_Name          2
   localityName = My_Locality_Name          3
   organizationName = My_Organization_Or_Company_Name
   organizationalUnitName = My_Organizational_Unit_Name          4
   ```

   1   Two letter code

   2   Full name

   3   Full name (example: New York)

   4   Division responsible for the certificate (example: IT department)

5. Generate 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

6. Send the certificate signing request to the certificate authority (CA). The same CA 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 CA for the preferred method. In response to the request, you can expect to receive a CA bundle and a signed certificate, in separate files.

3.11.2. 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.

> **IMPORTANT**
>
> Do not store the SSL certificates or .tar bundles in **/tmp** or **/var**/**tmp** directory. The operating system removes files from these directories periodically. As a result, **satellite-installer** fails to execute while enabling features or upgrading Satellite Server.

Procedure

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 Satellite Server certificate file that is signed by a Certificate Authority.

2 Path to the private key that was used to sign Satellite 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
```

Note that you must not access or modify **/root/ssl-build**.

2. From the output of the **katello-certs-check** command, depending on your requirements, enter the **satellite-installer** command that installs a new Satellite with custom SSL certificates or updates certificates on a currently running Satellite.
If you are unsure which command to run, you can verify that Satellite is installed by checking if the file **/etc/foreman-installer/scenarios.d/.installed** exists. If the file exists, run the second **satellite-installer** command that updates certificates.
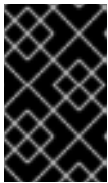
> **IMPORTANT**
>
> **satellite-installer** needs the certificate archive file after you deploy the certificate. Do not modify or delete it. It is required, for example, when upgrading Satellite Server.

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.

3.11.3. Deploying a custom SSL certificate to hosts

After you configure Satellite to use a custom SSL certificate, you must deploy the certificate to hosts registered to Satellite.

Procedure

- Update the SSL certificate on each host:

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

3.12. USING EXTERNAL DATABASES WITH SATELLITE

As part of the installation process for Red Hat Satellite, the **satellite-installer** command installs PostgreSQL databases on the same server as Satellite. In certain Satellite deployments, using external databases instead of the default local databases can help with the server load.

Red Hat does not provide support or tools for external database maintenance. This includes backups, upgrades, and database tuning. You must have your own database administrator to support and maintain external databases.

To create and use external databases for Satellite, you must complete the following procedures:

1. Section 3.12.2, "Preparing a host for external databases" . Prepare a Red Hat Enterprise Linux 8 server to host the external databases.

2. Section 3.12.3, "Installing PostgreSQL" . Prepare PostgreSQL with databases for Satellite, Candlepin and Pulp with dedicated users owning them.

3. Section 3.12.4, "Configuring Satellite Server to use external databases" . Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**.

3.12.1. 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 decide if this option is right for your Satellite configuration. Satellite supports PostgreSQL version 12.

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.

3.12.2. Preparing a host for external databases

Install a freshly provisioned system with the latest Red Hat Enterprise Linux 8 to host the external databases.

Subscriptions for 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 operating system that you want to use for the external databases.

Prerequisites

- The prepared host must meet Satellite's Storage Requirements.

Procedure

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

2. Disable all repositories and enable only the following repositories:

   ```
   # subscription-manager repos --disable '*'
   # subscription-manager repos \
   --enable=satellite-6.15-for-rhel-8-x86_64-rpms \
   --enable=satellite-maintenance-6.15-for-rhel-8-x86_64-rpms \
   --enable=rhel-8-for-x86_64-baseos-rpms \
   --enable=rhel-8-for-x86_64-appstream-rpms
   ```

3. Enable the following module:

   ```
   # dnf module enable satellite:el8
   ```

   > **NOTE**
   >
   > Enablement of the module **satellite:el8** warns about a conflict with **postgresql:10** and **ruby:2.5** as these modules are set to the default module versions on Red Hat Enterprise Linux 8. The module **satellite:el8** has a dependency for the modules **postgresql:12** and **ruby:2.7** that will be enabled with the **satellite:el8** module. These warnings do not cause installation process failure, hence can be ignored safely. For more information about modules and lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux Application Streams Lifecycle.

3.12.3. Installing PostgreSQL

You can install only the same version of PostgreSQL that is installed with the **satellite-installer** tool during an internal database installation. Satellite supports PostgreSQL version 12.

Procedure

1. To install PostgreSQL, enter the following command:

   ```
   # dnf install postgresql-server postgresql-evr postgresql-contrib
   ```

2. To initialize PostgreSQL, enter the following command:

   ```
   # postgresql-setup initdb
   ```

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

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

Note that the default configuration of external PostgreSQL needs to be adjusted to work with Satellite. The base recommended external database configuration adjustments are as follows:

- checkpoint_completion_target: 0.9

- max_connections: 500

- shared_buffers: 512MB

- work_mem: 4MB

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_ip*/32   md5

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

   # systemctl enable --now postgresql

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

   # firewall-cmd --add-service=postgresql

9. Make the changes persistent:

   # firewall-cmd --runtime-to-permanent

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

    $ su - postgres -c psql

11. Create three databases and dedicated roles: one for Satellite, one for Candlepin, and one for Pulp:

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

12. Connect to the Pulp database:

    ```
    postgres=# \c pulpcore
    You are now connected to database "pulpcore" as user "postgres".
    ```

13. Create the **hstore** extension:

```
pulpcore=# CREATE EXTENSION IF NOT EXISTS "hstore";
CREATE EXTENSION
```

14. Exit the **postgres** user:

```
# \q
```

15. 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"
# PGPASSWORD='Pulpcore_Password' psql -h postgres.example.com -p 5432 -U pulp -d
pulpcore -c "SELECT 1 as ping"
```

3.12.4. Configuring Satellite Server to use external databases

Use the **satellite-installer** command to configure Satellite to connect to an external PostgreSQL database.

Prerequisites

- You have installed and configured a PostgreSQL database on a Red Hat Enterprise Linux server.

Procedure

1. To configure the external databases for Satellite, enter the following command:

```
# satellite-installer \
--foreman-db-database foreman \
--foreman-db-host postgres.example.com \
--foreman-db-manage false \
--foreman-db-password Foreman_Password \
--foreman-proxy-content-pulpcore-manage-postgresql false \
--foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \
--foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \
--foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password \
--foreman-proxy-content-pulpcore-postgresql-user pulp \
--katello-candlepin-db-host postgres.example.com \
--katello-candlepin-db-name candlepin \
--katello-candlepin-db-password Candlepin_Password \
--katello-candlepin-manage-db false
```

To enable the Secure Sockets Layer (SSL) protocol for these external databases, add the following options:

```
--foreman-db-root-cert <path_to_CA>
--foreman-db-sslmode verify-full
--foreman-proxy-content-pulpcore-postgresql-ssl true
```

--foreman-proxy-content-pulpcore-postgresql-ssl-root-ca <path_to_CA>
--katello-candlepin-db-ssl true
--katello-candlepin-db-ssl-ca <path_to_CA>
--katello-candlepin-db-ssl-verify true

CHAPTER 4. CONFIGURING SATELLITE SERVER WITH EXTERNAL SERVICES

If you do not want to configure the DNS, DHCP, and TFTP services on Satellite Server, use this section to configure your Satellite Server to work with external DNS, DHCP, and TFTP services.

4.1. CONFIGURING SATELLITE SERVER WITH EXTERNAL DNS

You can configure Satellite Server with external DNS. Satellite Server uses the **nsupdate** utility to update DNS records on the remote server.

To make any changes persistent, you must enter the **satellite-installer** command with the options appropriate for your environment.

Prerequisites

- You must have a configured external DNS server.

- This guide assumes you have an existing installation.

Procedure

1. Copy the **/etc/rndc.key** file from the external DNS server to Satellite Server:

   ```
   # scp root@dns.example.com:/etc/rndc.key /etc/foreman-proxy/rndc.key
   ```

2. Configure the ownership, permissions, and SELinux context:

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

3. To test the **nsupdate** utility, add a host remotely:

   ```
   # echo -e "server DNS_IP_Address\n \
   update add aaa.example.com 3600 IN A Host_IP_Address\n \
   send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
   # nslookup aaa.example.com DNS_IP_Address
   # echo -e "server DNS_IP_Address\n \
   update delete aaa.example.com 3600 IN A Host_IP_Address\n \
   send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
   ```

4. Enter the **satellite-installer** command to make the following persistent changes to the **/etc/foreman-proxy/settings.d/dns.yml** file:

   ```
   # satellite-installer --foreman-proxy-dns=true \
   --foreman-proxy-dns-managed=false \
   --foreman-proxy-dns-provider=nsupdate \
   --foreman-proxy-dns-server="DNS_IP_Address" \
   --foreman-proxy-keyfile=/etc/foreman-proxy/rndc.key
   ```

5. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.

6. Locate the Satellite Server and select **Refresh** from the list in the **Actions** column.

7. Associate the DNS service with the appropriate subnets and domain.

4.2. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP

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

1. Section 4.2.1, "Configuring an external DHCP server to use with Satellite Server"

2. Section 4.2.2, "Configuring Satellite Server with an external DHCP server"

4.2.1. Configuring an external DHCP server to use with Satellite Server

To configure an external DHCP server running Red Hat Enterprise Linux to use with Satellite Server, you must install the ISC DHCP Service and Berkeley Internet Name Domain (BIND) utilities 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.

> **NOTE**
>
> If you use dnsmasq as an external DHCP server, enable the **dhcp-no-override** setting. This is required because Satellite creates configuration files on the TFTP server under the **grub2/** subdirectory. If the **dhcp-no-override** setting is disabled, hosts fetch the bootloader and its configuration from the root directory, which might cause an error.

Procedure

1. On your Red Hat Enterprise Linux host, install the ISC DHCP Service and Berkeley Internet Name Domain (BIND) utilities packages:

   ```
   # dnf install dhcp-server bind-utils
   ```

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:

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

4. Edit the **dhcpd** configuration file for all 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 "My_Secret";
    };
    omapi-key omapi_key;
```

Note that the **option routers** value is the IP address of your Satellite Server or Capsule Server 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
   ```

8. Make the changes persistent:

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

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

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

10. 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
    ```

11. 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
    ```

12. Enable and start the DHCP service:

    ```
    # systemctl enable --now dhcpd
    ```

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

    ```
    # dnf install nfs-utils
    # systemctl enable --now nfs-server
    ```

14. 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
    ```

15. 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
    ```

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

    ```
    # mount -a
    ```

17. 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.

18. Reload the NFS server:

    ```
    # exportfs -rva
    ```

19. Configure the firewall for DHCP omapi port 7911:

    ```
    # firewall-cmd --add-port=7911/tcp
    ```

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

    ```
    # firewall-cmd \
    --add-service mountd \
    --add-service nfs \
    --add-service rpc-bind \
    --zone public
    ```

21. Make the changes persistent:

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

4.2.2. Configuring Satellite Server with an external DHCP server

You can configure Satellite Server with an external DHCP server.

Prerequisites

- Ensure that you have configured an external DHCP server and that you have shared the DHCP configuration and lease files with Satellite Server. For more information, see Section 4.2.1, "Configuring an external DHCP server to use with Satellite Server".

Procedure

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

   ```
   # satellite-maintain packages 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
   $ cat /mnt/nfs/etc/dhcp/dhcpd.conf
   $ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases
   $ 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 \
   --enable-foreman-proxy-plugin-dhcp-remote-isc \
   ```

```
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-dhcp-server=My_DHCP_Server_FQDN \
--foreman-proxy-dhcp=true \
--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=My_Secret \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911
```

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

4.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

You can configure Satellite Server with external TFTP services.

Procedure

1. Create the TFTP directory for NFS:

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

2. In the **/etc/fstab** file, add the following line:

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

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

   ```
   # mount -a
   ```

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

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

5. If the TFTP service is running on a different server than the DHCP service, update the **tftp_servername** setting with the FQDN or IP address of the server that the TFTP service is running on:

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

6. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.

7. Locate the Satellite Server and select **Refresh** from the list in the **Actions** column.

8. Associate the TFTP service with the appropriate subnets and domain.

4.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL IDM DNS

When Satellite Server adds a DNS record for a host, it first determines which Capsule is providing DNS for that domain. It then communicates with the Capsule that is configured to provide DNS service for

your deployment and adds the record. The hosts are not involved in this process. Therefore, you must 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.

Satellite Server can be configured to use a Red Hat Identity Management (IdM) server to provide DNS service. For more information about Red Hat Identity Management, see the Linux Domain Identity, Authentication, and Policy Guide.

To configure Satellite Server to use a Red Hat Identity Management (IdM) server to provide DNS service, use one of the following procedures:

- Section 4.4.1, "Configuring dynamic DNS update with GSS-TSIG authentication"

- Section 4.4.2, "Configuring dynamic DNS update with TSIG authentication"

To revert to internal DNS service, use the following procedure:

- Section 4.4.3, "Reverting to internal DNS service"

> **NOTE**
>
> You are not required to use Satellite Server to manage DNS. When you are using the realm enrollment feature of Satellite, where provisioned hosts are enrolled automatically to IdM, the **ipa-client-install** script creates DNS records for the client. Configuring Satellite Server with external IdM DNS and realm enrollment are mutually exclusive. For more information about configuring realm enrollment, see External Authentication for Provisioned Hosts in *Installing Satellite Server in a connected network environment* .

4.4.1. Configuring dynamic DNS update with GSS-TSIG authentication

You can configure the IdM server to use the generic security service algorithm for secret key transaction (GSS-TSIG) technology defined in RFC3645. To configure the IdM server to use the GSS-TSIG technology, you must install the IdM client on the Satellite Server base operating system.

Prerequisites

- You must ensure the IdM server is deployed and the host-based firewall is configured correctly. For more information, see Port Requirements for IdM in the *Installing Identity Management Guide*.

- You must contact the IdM server administrator to ensure that you obtain an account on the IdM server with permissions to create zones on the IdM server.

- You should create a backup of the answer file. You can use the backup to restore the answer file to its original state if it becomes corrupted. For more information, see Configuring Satellite Server.

Procedure

To configure dynamic DNS update with GSS-TSIG authentication, complete the following steps:

Creating a Kerberos principal on the IdM server

1. Obtain a Kerberos ticket for the account obtained from the IdM administrator:

   ```
   # kinit idm_user
   ```

2. Create a new Kerberos principal for Satellite Server to use to authenticate on the IdM server:

   > # ipa service-add *capsule/satellite.example.com*

Installing and configuring the idM client

1. On the base operating system of either the Satellite or Capsule that is managing the DNS service for your deployment, install the **ipa-client** package:

   > # satellite-maintain packages install ipa-client

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

   > # ipa-client-install

3. Obtain a Kerberos ticket:

   > # kinit admin

4. Remove any preexisting **keytab**:

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

5. Obtain the **keytab** 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. For the **dns.keytab** file, set the group and owner to **foreman-proxy**:

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

7. Optional: To verify that the **keytab** file is valid, enter the following command:

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

Configuring DNS zones in the IdM web UI

1. Create and configure the zone that you want to manage:

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

   b. Select **Add** and enter the zone name. For example, **example.com**.

   c. Click **Add and Edit**

d. Click the Settings tab and in the **BIND update policy** box, add the following to the semi-colon separated list:

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

e. Set **Dynamic update** to **True**.

f. Enable **Allow PTR sync**.

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

2. Create and configure the reverse zone:

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

b. Click **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. Click the **Settings** tab and in the **BIND update policy** box, add the following to the semi-colon separated list:

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

f. Set **Dynamic update** to **True**.

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

Configuring the Satellite or Capsule Server that manages the DNS service for the domain

1. Configure your Satellite Server or Capsule Server to connect to your DNS service:

```
# satellite-installer \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns=true
```

2. For each affected Capsule, update the configuration of that Capsule in the Satellite web UI:

a. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.

b. Configure the domain:

i. In the Satellite web UI, navigate to **Infrastructure** > **Domains** and select the domain name.

ii. In the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.

c. Configure the subnet:

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

    ii. In the **Subnet** tab, set **IPAM** to **None**.

    iii. In the **Domains** tab, select the domain that you want to manage using the IdM server.

    iv. In the **Capsules** tab, ensure **Reverse DNS Capsule** is set to the Capsule where the subnet is connected.

    v. Click **Submit** to save the changes.

4.4.2. Configuring dynamic DNS update with TSIG authentication

You can configure an IdM server to use the secret key transaction authentication for DNS (TSIG) technology that uses the **rndc.key** key file for authentication. The TSIG protocol is defined in RFC2845.

Prerequisites

- You must ensure the IdM server is deployed and the host-based firewall is configured correctly. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.

- You must obtain **root** user access on the IdM server.

- You must confirm whether Satellite Server or Capsule Server is configured to provide DNS service for your deployment.

- You must configure DNS, DHCP and TFTP services on the base operating system of either the Satellite or Capsule that is managing the DNS service for your deployment.

- You must create a backup of the answer file. You can use the backup to restore the answer file to its original state if it becomes corrupted. For more information, see Configuring Satellite Server.

Procedure

To configure dynamic DNS update with TSIG authentication, complete the following steps:

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:

   ```
   ######################################################################

   include "/etc/rndc.key";
   controls  {
   inet _IdM_Server_IP_Address_ port 953 allow { _Satellite_IP_Address_; } keys { "rndc-key";
   };
   };
   ######################################################################
   ```

2. Reload the **named** service to make the changes take effect:

```
# systemctl reload named
```

3. In the IdM web UI, navigate to **Network Services** > **DNS** > **DNS Zones** and click the name of the zone. In the **Settings** tab, apply the following changes:

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

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

   b. Set **Dynamic update** to **True**.

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

4. Copy the **/etc/rndc.key** file from the IdM server to the base operating system of your Satellite Server. Enter the following command:

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

5. To set the correct ownership, permissions, and SELinux context for the **rndc.key** file, enter the following command:

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

6. Assign the **foreman-proxy** user to the **named** group manually. Normally, satellite-installer ensures that the **foreman-proxy** user belongs to the **named** UNIX group, however, in this scenario Satellite does not manage users and groups, therefore you need to assign the **foreman-proxy** user to the **named** group manually.

   ```
   # usermod -a -G named foreman-proxy
   ```

7. On Satellite Server, enter the following **satellite-installer** command to configure Satellite to use the external DNS server:

   ```
   # satellite-installer \
   --foreman-proxy-dns-managed=false \
   --foreman-proxy-dns-provider=nsupdate \
   --foreman-proxy-dns-server="IdM_Server_IP_Address" \
   --foreman-proxy-dns-ttl=86400 \
   --foreman-proxy-dns=true \
   --foreman-proxy-keyfile=/etc/rndc.key
   ```

Testing external updates to the DNS zone in the IdM server

1. Ensure that the key in the **/etc/rndc.key** file on Satellite Server is the same key file that is used on the IdM server:

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

2. 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
   ```

3. 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
   ```

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

5. 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
   ```

6. 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.

4.4.3. Reverting to internal DNS service

You can revert to using Satellite Server and Capsule Server as your DNS providers. You can use a backup of the answer file that was created before configuring external DNS, or you can create a backup of the answer file. For more information about answer files, see Configuring Satellite Server.

Procedure

On the Satellite or Capsule Server that you want to configure to manage DNS service for the domain, complete the following steps:

Configuring Satellite or Capsule as a DNS server

- If you have created a backup of the answer file before configuring external DNS, restore the answer file and then enter the **satellite-installer** command:

  ```
  # satellite-installer
  ```

- If you do not have a suitable backup of the answer file, create a backup of the answer file now. To configure Satellite or Capsule as DNS server without using an answer file, enter the following **satellite-installer** command on Satellite or Capsule:

```
# satellite-installer \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1" \
--foreman-proxy-dns=true
```

For more information, see Configuring DNS, DHCP, and TFTP on Capsule Server .

After you run the **satellite-installer** command to make any changes to your Capsule configuration, you must update the configuration of each affected Capsule in the Satellite web UI.

Updating the configuration in the Satellite web UI

1. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.

2. For each Capsule that you want to update, from the **Actions** list, select **Refresh**.

3. Configure the domain:

   a. In the Satellite web UI, navigate to **Infrastructure** > **Domains** and click the domain name that you want to configure.

   b. In the **Domain** tab, set **DNS Capsule** to the Capsule where the subnet is connected.

4. Configure the subnet:

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

   b. In the **Subnet** tab, set **IPAM** to **DHCP** or **Internal DB**.

   c. In the **Domains** tab, select the domain that you want to manage using Satellite or Capsule.

   d. In the **Capsules** tab, set **Reverse DNS Capsule** to the Capsule where the subnet is connected.

   e. Click **Submit** to save the changes.

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 Restoring Manual Changes Overwritten by a Puppet Run.

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

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.

APPENDIX B. RESTORING 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.

Procedure

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.

APPENDIX C. REVERTING SATELLITE SERVER TO DOWNLOAD CONTENT FROM RED HAT CDN

If your environment changes from disconnected to connected, you can reconfigure a disconnected Satellite Server 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. Switch to the **CDN Configuration** tab.

4. Select **Red Hat CDN**.

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

6. Click **Update**.

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

CLI procedure

1. Log in to the Satellite Server using SSH.

2. Use Hammer to reconfigure the CDN:

   ```
   # hammer organization configure-cdn --name="My_Organization" --type=redhat_cdn
   ```