Red Hat Satellite 6.4

Installing Satellite Server from a Connected Network

Installing Red Hat Satellite Server from a Connected Network

Last Updated: 2020-03-06

Red Hat Satellite 6.4 Installing Satellite Server from a Connected Network

Installing Red Hat Satellite Server from a Connected Network

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

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base system:

- x86_64 architecture
- The latest version of Red Hat Enterprise Linux 7 Server
- 4-core 2.0 GHz CPU at a minimum
- A minimum of 20 GB memory is required for the Satellite Server to function. In addition, a minimum of 4 GB of swap space is also recommended. Satellite running with less memory than the minimum value might not operate correctly.
- A unique host name, which can contain lower-case letters, numbers, dots (.) and hyphens (-)
- A current Red Hat Satellite subscription
- Administrative user (root) access
- A system umask of 0022
- Full forward and reverse DNS resolution using a fully-qualified domain name

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

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

- postgres
- mongodb
- apache
- tomcat
- foreman
- foreman-proxy
- qpidd
- qdrouterd
- squid
- puppet

NOTE

The Red Hat Satellite Server and Capsule Server versions must match. For example, a Satellite 6.2 Server cannot run a 6.4 Capsule Server and a Satellite 6.4 Server cannot run a 6.2 Capsule Server. Mismatching Satellite Server and Capsule Server versions results in the Capsule Server failing silently.

NOTE

Self-registered Satellites are not supported.

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

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

Certified hypervisors

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

1.2. STORAGE REQUIREMENTS AND GUIDELINES

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

1.2.1. Storage Requirements

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

The runtime size was measured with Red Hat Enterprise Linux 5, 6, and 7 repositories synchronized.

Table 1.1. Storage Requirements for a Connected Satellite Server Installation

Directory	Installation Size	Runtime Size
/var/cache/pulp/	1 MB	20 GB
/var/lib/pulp/	1 MB	500 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/lib/qpidd/	25 MB	Not Applicable
/var/log/	10 MB	250 MB
/var/lib/pgsql/	100 MB	10 GB
/var/spool/squid/	0 MB	10 GB

Directory	Installation Size	Runtime Size
/usr	3 GB	Not Applicable
/opt	3 GB	Not Applicable
/opt/puppetlabs	500 MB	Not Applicable

1.2.2. Storage Guidelines

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

- Because most Satellite and Capsule Server data is stored within the **/var** directory, mounting **/var** on LVM storage can help the system to scale.
- For the **/var/lib/pulp/** and **/var/lib/mongodb/** directories, use high-bandwidth, low-latency storage, and solid state drives (SSD) rather than hard disk drives (HDD). As Red Hat Satellite has many operations that are I/O intensive, using high latency, low-bandwidth storage causes performance degradation. Ensure your installation has a speed in the range 60 - 80 Megabytes per second. You can use the **fio** tool to get this data. See the Red Hat Knowledgebase solution Impact of Disk Speed on Satellite 6 Operations for more information on using the **fio** tool.
- The **/var/lib/qpidd/** directory uses slightly more than 2 MB per Content Host managed by the **goferd** service. For example, 10 000 Content Hosts require 20 GB of disk space in **/var/lib/qpidd/**.
- Using the same volume for the **/var/cache/pulp/** and **/var/lib/pulp/** directories can decrease the time required to move content from **/var/cache/pulp/** to **/var/lib/pulp/** after synchronizing.

File System Guidelines

- Use the XFS file system for Red Hat Satellite 6 because it does not have the inode limitations that **ext4** does. Because Satellite uses a lot of symbolic links it is likely that your system might run out of inodes if using **ext4** and the default number of inodes.
- Do not use NFS with MongoDB because MongoDB does not use conventional I/O to access data files and performance problems occur when both the data files and the journal files are hosted on NFS. If required to use NFS, mount the volume with the following options in the **/etc/fstab** file: **bg**, **nolock**, and **noatime**.
- Do not use the GFS2 file system as the input-output latency is too high.

SELinux Considerations for NFS Mount

When **/var/lib/pulp** directory is mounted using an NFS share, SELinux blocks the synchronization process. To avoid this, specify the SELinux context of the **/var/lib/pulp** directory in the file system table by adding the following lines to **/etc/fstab**:

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

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

-

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

Duplicated Packages

Packages that are duplicated in different repositories are only stored once on the disk. Additional repositories containing duplicate packages require less additional storage. The bulk of storage resides in the **/var/lib/mongodb/** and **/var/lib/pulp/** directories. These end points are not manually configurable. Ensure that storage is available on the **/var** file system to prevent storage problems.

Temporary Storage

The **/var/cache/pulp/** directory is used to temporarily store content while it is being synchronized. For content in RPM format, a maximum of 5 RPM files are stored in this directory at any time. After each file is synchronized, it is moved to the **/var/lib/pulp/** directory. Up to 8 RPM content synchronization tasks can run simultaneously by default, with each using up to 1 GB of metadata.

Software Collections

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

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

Symbolic links

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

Synchronized RHEL ISO

If you plan to synchronize RHEL content ISOs to Satellite, note that all minor versions of Red Hat Enterprise Linux also synchronize. You must plan to have adequate storage on your Satellite to manage this.

1.3. SUPPORTED OPERATING SYSTEMS

You can install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Satellite Server and Red Hat Satellite Capsule Server are supported only on the latest versions of Red Hat Enterprise Linux 7 Server that is available at the time when Satellite 6.4 is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

Red Hat Satellite Server and Red Hat Satellite Capsule Server require Red Hat Enterprise Linux installations with the **@Base** package group with no other package-set modifications, and without third-party configurations or software not directly necessary for the direct operation of the server. This restriction includes hardening and other non-Red Hat security software. If you require such software in your infrastructure, install and verify a complete working Satellite Server first, then create a backup of the system before adding any non-Red Hat software.

Install Satellite Server and Capsule Server on a freshly provisioned system. Do not register Capsule Server to the Red Hat Content Delivery Network (CDN). Red Hat does not support using the system for anything other than running Satellite.

1.4. SUPPORTED BROWSERS

The following web browsers are fully supported:

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

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

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

NOTE

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

1.5. PORTS AND FIREWALLS REQUIREMENTS

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

The following tables indicate the destination port and the direction of network traffic. Use this information to configure any network-based firewalls. Note that some cloud solutions 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 these tables. This includes the base system on which a Capsule Server is running.

Clients of Capsule

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

Required ports can change based on your configuration.

Table 1.2. Ports for Satellite to Red Hat CDN Communication

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

Except in the case of a disconnected Satellite, Satellite Server needs access to the Red Hat CDN. For a list of IP addresses used by the Red Hat CDN (cdn.redhat.com), see the Knowledgebase article Public CIDR Lists for Red Hat on the Red Hat Customer Portal.

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

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

Table 1.4. Ports for Client to Satellite Communication

Port	Protocol	Service	Required For
80	TCP	HTTP	Anaconda, yum, for obtaining Katello certificates, templates, and for downloading iPXE firmware
443	TCP	HTTPS	Subscription Management Services, yum, Telemetry Services, and for connection to the Katello Agent
5647	TCP	amqp	Katello Agent to communicate with Satellite's Qpid dispatch router
8000	TCP	HTTP	Anaconda to download kickstart templates to hosts, and for downloading iPXE firmware
8140	TCP	HTTPS	Puppet agent to Puppet master connections
9090	TCP	HTTPS	Sending SCAP reports to the Smart Proxy in the integrated Capsule, for the discovery image during provisioning, and for communicating with Satellite Server to copy the SSH keys for Remote Execution (Rex) configuration
5000	TCP	HTTPS	Connection to Katello for the Docker registry
7	TCP and UDP	ICMP	External DHCP on a Client to Satellite network, ICMP ECHO to verify IP address is free (Optional)
53	TCP and UDP	DNS	Client DNS queries to a Satellite's integrated Capsule DNS service (Optional)

Port	Protocol	Service	Required For
67	UDP	DHCP	Client to Satellite's integrated Capsule broadcasts, DHCP broadcasts for Client provisioning from a Satellite's integrated Capsule (Optional)
69	UDP	TFTP	Clients downloading PXE boot image files from a Satellites' integrated Capsule for provisioning (Optional)

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

Table 1.5. Ports for Satellite to Capsule Communication

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

Table 1.6. Optional Network Ports

Port	Protocol	Service	Required For
22	TCP	SSH	Satellite and Capsule originated communications, for Remote Execution (Rex) and Ansible.
443	TCP	HTTPS	Satellite originated communications, for vCenter compute resource.
5000	TCP	HTTP	Satellite originated communications, for compute resources in OpenStack or for running containers.
22, 16514	TCP	SSH, SSL/TLS	Satellite originated communications, for compute resources in libvirt.
389, 636	TCP	LDAP, LDAPS	Satellite originated communications, for LDAP and secured LDAP authentication sources.

Port	Protocol	Service	Required For
5900 to 5930	TCP	SSL/TLS	Satellite originated communications, for NoVNC console in web UI to hypervisors.

1.6. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER

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

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

Configuring the Firewall

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

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

2. Make the changes persistent:

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

1.7. VERIFYING FIREWALL SETTINGS

You can verify changes to firewall settings using the **firewall-cmd** command.

To verify firewall settings:

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

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

1.8. VERIFYING DNS RESOLUTION

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

Ensure that the host name and local host resolve correctly.

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

Successful name resolution results in output similar to the following:

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

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

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

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

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

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

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

WARNING

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

1.9. CHANGING DEFAULT SELINUX PORTS

Red Hat Satellite 6 uses a set of predefined ports. Because Red Hat recommends that SELinux on Satellite 6 systems be set to permissive or enforcing, if you need to change the port for any service, you also need to change the associated SELinux port type to allow access to the resources. You only need to change these ports if you use non-standard ports.

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

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

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

Before You Begin

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

Changing default ports to user-specified ports

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

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

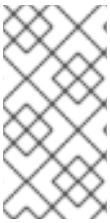
2. Disassociate the previously used port number and port type.

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

CHAPTER 2. INSTALLING SATELLITE SERVER

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

Red Hat Satellite 6.4 uses Puppet 5 by default. Review and update your Puppet modules to support Puppet 5. For more information about updating your Puppet modules to support Puppet 5, see the Upgrading Puppet section in the *Satellite 6.4 Upgrading and Updating Red Hat Satellite* guide.



NOTE

Ensure you have completed upgrading your Puppet modules to support Puppet 4 while on Red Hat Satellite 6.3. For information on upgrading Puppet modules to Puppet 4, see the Upgrading Puppet section in the *Satellite 6.3 Upgrading and Updating Red Hat Satellite* guide.

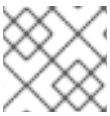
There are two methods of installing Satellite Server:

Connected:

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

Disconnected:

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



NOTE

You cannot register Satellite Server to itself.

2.1. INSTALLING SATELLITE SERVER FROM A CONNECTED NETWORK

When you install Satellite Server from a connected network, you can obtain packages and receive updates directly from the Red Hat Content Delivery Network.

Note that the Satellite 6 installation script is based on Puppet, which means that if you run the installation script more than once, it might overwrite any manual configuration changes. To avoid this and determine which future changes apply, use the **--noop** argument when you run the installation script. This argument ensures that no actual changes are made. Potential changes are written to **/var/log/foreman-installer.log**.

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

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

You can restore the previous file as follows:

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

2.1.1. Registering to Red Hat Subscription Management

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

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

```
# subscription-manager register
```

The command displays output similar to the following:

```
# subscription-manager register
Username: user_name
Password:
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-7683431bcf9a
```

2.1.2. Identifying and Attaching the Satellite Subscription to the Host

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

1. Identify your Satellite subscription.

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

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

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

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

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

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

```
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2014
System Type:       Physical
```

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

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

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

The output should be similar to the following:

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

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

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

The outputs displays something similar to the following:

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

2.1.3. Configuring Repositories

1. Disable all existing repositories.

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

2. Enable the Red Hat Satellite, Red Hat Enterprise Linux, and Red Hat Software Collections repositories.

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-6.4-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-ansible-2.6-rpms
```

NOTE

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

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

```
# yum clean all
```

4. Verify that the repositories have been enabled.

```
# yum repolist enabled
```

2.1.4. Installing the Satellite Server Packages

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

1. Update all packages:

```
# yum update
```

2. Install the Satellite Server packages:

```
# yum install satellite
```

2.2. PERFORMING THE INITIAL CONFIGURATION

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

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

2.2.1. Synchronizing Time

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

Two NTP based time synchronizers are available: **chronyd** and **ntpd**. The **chronyd** implementation is specifically recommended for systems that are frequently suspended and for systems that have intermittent network access. The **ntpd** implementation should only be used when you specifically need support for a protocol or driver not yet supported by **chronyd**.

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

Synchronizing Time using chronyd

1. Install chronyd.

```
# yum install chrony
```

2. Start and enable the chronyd service.

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

2.2.2. Installing the SOS Package on the Host Operating System

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

Install the **sos** package.

```
# yum install sos
```

2.2.3. Specifying Installation Options

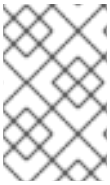
Satellite Server is installed using the **satellite-installer** installation script and as part of the initial configuration, you either automatically or manually configure Satellite.

Choose from one of these two methods:

- Automatic Configuration - This method is performed by using an answer file to automate the configuration process when running the installation script. An answer file is a file containing a list of parameters that are read by a command or script. The default Satellite answer file is **/etc/foreman-installer/scenarios.d/satellite-answers.yaml**. The answer file in use is set by the **answer_file** directive in the **/etc/foreman-installer/scenarios.d/satellite.yaml** configuration file.

To perform the initial configuration using the installation script with an answer file, see Section 2.2.3.2, "Performing the Initial Configuration Automatically using an Answer File" .

- Manual Configuration - This method is performed by running the installation script with one or more command options. The command options override the corresponding default initial configuration options and are recorded in the Satellite answer file. You can run the script as often as needed to configure any necessary options.
  To perform the initial configuration using the installation script with command-line options, see Section 2.2.3.1, "Performing the Initial Configuration Manually" .



NOTE

Depending on the options that you use when running the Satellite installer, the configuration can take several minutes to complete. An administrator is able to view the answer file to see previously used options for both methods.

2.2.3.1. Performing the Initial Configuration Manually

This initial configuration procedure creates an organization, location, user name, and password. After the initial configuration, you can create additional organizations and locations if required. The initial configuration also installs MongoDB and PostgreSQL databases on the same server. Depending on your deployment, using external databases can benefit performance.

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

Manually configuring Satellite Server

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

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

By default, all configuration files configured by the installer are managed by Puppet. When **satellite-installer** runs, it overwrites any manual changes to the Puppet managed files with the initial values. By default, Satellite Server is installed with the Puppet agent running as a service. If required, you can disable Puppet agent on Satellite Server using the **--puppet-runmode=none** option.

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

If you want to use external databases with Satellite, before you run the **satellite installer** tool, you must set up and point to external databases. For more information, see Using External Databases with Satellite in *Installing Satellite Server from a Connected Network* .

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

```
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-admin-username admin_user_name \
--foreman-admin-password admin_password \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

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

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

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

2.2.3.2. Performing the Initial Configuration Automatically using an Answer File

You can use answer files to automate installations with customized options. The initial answer file is sparsely populated and after you run the **satellite-installer** script the first time, the answer file is populated with the standard parameter values for installation. If you have already installed Satellite Server using the method described in Section 2.2.3.1, "Performing the Initial Configuration Manually", then you do not need to use this method. You can, however, use it to make changes to the configuration of Satellite Server at any time.

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

Automatically configuring Satellite Server using an Answer File

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

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

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

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

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

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

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

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

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

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

2.2.4. Creating a Subscription Allocation in Customer Portal

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

1. Open https://access.redhat.com/ in your browser and log in to the Red Hat account that you used to register the system to Red Hat Subscription Management.
2. Navigate to **Subscriptions** in the upper-left corner of the Customer Portal.
3. Navigate to **Subscription Allocations**.
4. Click **Create New subscription allocation**
5. In the **Name** field, enter a name.
6. From the **Type** list, select the type and version that corresponds to your Satellite Server.
7. Click **Create**.

2.2.5. Adding Subscriptions to an Allocation

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

1. Navigate to **Subscription Allocations**.
2. Select the name of the subscription you want to change.
3. Click the **Subscriptions** tab.
4. Click **Add Subscriptions**.
5. A list of your Red Hat product subscriptions appears. Enter the **Entitlement Quantity** for each product.
6. Click **Submit** to complete the assignment.

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

2.2.6. Exporting a Subscription Manifest from the Customer Portal

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

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

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

2.2.7. Importing a Subscription Manifest into the Satellite Server

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

For Web UI Users

1. Ensure the context is set to the organization you want to use.
2. Navigate to **Content** > **Red Hat Subscriptions**.
3. Click **Manage Manifest** to display the manifest page for the organization.
4. Click **Choose file**, select the Subscription Manifest, then click **Upload**.

For CLI Users

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

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

Then import it using the following command:

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

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

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

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER

3.1. INSTALLING THE SATELLITE TOOLS REPOSITORY

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

To Install the Satellite Tools Repository Using the Web UI:

1. In the Satellite web UI, navigate to **Content** > **Red Hat Repositories**.
2. Use the Search field to enter the following repository name: **Red Hat Satellite Tools 6.4 (for RHEL 7 Server) (RPMs)**.
3. In the Available Repositories pane, click on **Red Hat Satellite Tools 6.4 (for RHEL 7 Server) (RPMs)** to expand the repository set.
   If the **Red Hat Satellite Tools 6.4** items are not visible, it may be because they are not included in the Subscription Manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the Subscription Manifest and import it into Satellite.
4. For the **x86_64** entry, click the **Enable** icon to enable the repository.

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

For CLI Users

Enable the Satellite Tools repository:

```
# hammer repository-set enable --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--basearch='x86_64' \
--name 'Red Hat Satellite Tools 6.4 (for RHEL 7 Server) (RPMs)'
```

To Synchronize the Satellite Tools Repository Using the Web UI:

1. Navigate to **Content** > **Sync Status**.
   A list of product repositories available for synchronization is displayed.
2. Click the arrow next to the product content to view available content.
3. Select the content you want to synchronize.
4. Click **Synchronize Now**.

For CLI Users

Synchronize your Satellite Tools repository:

```
$ hammer repository synchronize --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--name 'Red Hat Satellite Tools 6.4 for RHEL 7 Server RPMs x86_64' \
--async
```

excluding

3.2. CONFIGURING SATELLITE SERVER WITH AN HTTP PROXY

If your network uses an HTTP Proxy, you can configure Satellite Server to use an HTTP proxy for requests to the Red Hat Content Delivery Network (CDN) or another content source. Use the FQDN instead of the IP address where possible to avoid losing connectivity because of network changes.

The following procedure configures a proxy only for downloading content for Satellite.

Authentication Methods

Only basic authentication is supported: add your user name and password information to the **--katello-proxy-url** option, or use the **--katello-proxy-username** and **--katello-proxy-password** options.

To Configure Satellite with an HTTP Proxy

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

```
# unset http_proxy
# unset https_proxy
# unset no_proxy
```

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

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

3. Verify that Satellite Server can connect to the Red Hat CDN and can synchronize its repositories.

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

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

Host name	Port	Protocol
api.access.redhat.com (if using Red Hat Insights)	443	HTTPS

Satellite Server communicates with the Red Hat CDN securely over SSL. Use of an SSL interception proxy interferes with this communication. These hosts must be whitelisted on the proxy.

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

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

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

# port for http proxy server
proxy_port = 8080

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

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

SELinux Considerations for Custom Ports

SELinux ensures access of Red Hat Satellite 6 and Red Hat Subscription Manager only to specific ports. In the case of the HTTP cache, the TCP ports are 8080, 8118, 8123, and 10001 - 10010. If you use a port that does not have SELinux type **http_cache_port_t**, complete the following steps:

1. To verify the ports that are permitted by SELinux for the HTTP cache, enter a command as follows:

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

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

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

For more information on SELinux port settings, see Section 1.9, “Changing Default SELinux ports”.

3.3. USING AN HTTP PROXY FOR ALL SATELLITE HTTP REQUESTS

If your Satellite Server must remain behind a firewall that blocks HTTP and HTTPS, you can configure a proxy for communication with external systems, including compute resources.

Note that if you are using compute resources for provisioning, and you want to use a different HTTP proxy with the compute resources, the proxy that you set for all Satellite communication takes precedence over the proxies that you set for compute resources.

To set an HTTP proxy for all outgoing HTTP connections from Satellite, complete the following steps:

1. In the Satellite web UI, navigate to **Administer** > **Settings**.
2. In the **HTTP(S) proxy** row, select the adjacent **Value** column and enter the proxy URL.
3. Click the tick icon to save your changes.

Excluding Hosts from Receiving Proxied Requests

If you use an HTTP Proxy for all Satellite HTTP or HTTPS requests, you can prevent certain hosts from communicating through the proxy.

To exclude one or more hosts from communicating through the proxy, complete the following steps:

1. In the Satellite web UI, navigate to **Administer** > **Settings**.
2. In the **HTTP(S) proxy except hosts**row, select the adjacent **Value** column and enter the names of one or more hosts that you want to exclude from proxy requests.
3. Click the tick icon to save your changes.

3.4. ENABLING POWER MANAGEMENT ON MANAGED HOSTS

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

The BMC service enables you to perform a range of power management tasks. The underlying protocol for this feature is IPMI; also referred to as the BMC function. IPMI uses a special network interface on the managed hardware that is connected to a dedicated processor that runs independently of the host's CPUs. In many instances the BMC functionality is built into chassis-based systems as part of chassis management (a dedicated module in the chassis).

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

Before You Begin

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

Enable Power Management on Managed Hosts

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

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

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

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

If you want to configure external services, see Chapter 4, *Configuring External Services*.

If you want to disable these services in Satellite in order to manage them manually, see Section 3.6, "Disabling DNS, DHCP, and TFTP for Unmanaged Networks".

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

Before You Begin

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

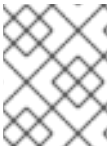
Configure DNS, DHCP, and TFTP on Satellite Server

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

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

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

   The script displays its progress and writes logs to **/var/log/foreman-installer/satellite.log**. You can view the settings used, including the **admin_password** parameter, in the **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** file.



NOTE

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

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

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

Procedure

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

1. On Satellite Server, enter the following command:

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

2. In the Satellite web UI, navigate to **Infrastructure** > **Subnets** and select a subnet.
3. Click the **Capsules** tab and clear the **DHCP Capsule**, **TFTP Capsule**, and **Reverse DNS Capsule** fields.
4. Navigate to **Infrastructure** > **Domains** and select a domain.
5. Clear the **DNS Capsule** field.
6. Optional: If you use a DHCP service supplied by a third party, configure your DHCP server to pass the following options:

```
Option 66: IP_address_of_Satellite_or_Capsule
Option 67: /pxelinux.0
```

   For more information about DHCP options, see RFC 2132.

NOTE

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

3.7. CONFIGURING SATELLITE SERVER FOR OUTGOING EMAILS

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

Prerequisites

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

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

To Configure Satellite Server for Outgoing Emails:

1. In the Satellite web UI, navigate to **Administer → Settings**.
2. Click the **Email** tab and set the configuration options to match your preferred delivery method. The changes have an immediate effect.
   a. The following example shows the configuration options for using an SMTP server:

   Table 3.1. Using an SMTP server as a delivery method

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

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

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

   Table 3.2. Using gmail.com as an SMTP server

Name	Example value
Delivery method	SMTP
SMTP address	smtp.gmail.com
SMTP authentication	plain
SMTP HELO/EHLO domain	smtp.gmail.com

Name	Example value
SMTP enable StartTLS auto	Yes
SMTP password	*password*
SMTP port	587
SMTP username	*user*@gmail.com

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

Table 3.3. Using sendmail as a delivery method

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

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

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

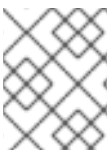
- Mark the CA certificate of the SMTP server as trusted. To do so, execute the following commands on Satellite Server:

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

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

- Alternatively, in the web UI, set the **SMTP enable StartTLS auto** option to **No**.

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



NOTE

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

3.8. CONFIGURING SATELLITE SERVER WITH A CUSTOM SERVER CERTIFICATE

SSL certificates are used to protect information and enable secure communication. Red Hat Satellite 6

creates self-signed SSL certificates to enable encrypted communications between the Satellite Server, external Capsule Servers, and all hosts. Instead of using these self-signed certificates, you can install custom SSL certificates issued by a Certificate Authority which is an external, trusted company. For example, your company might have a security policy stating that SSL certificates must be obtained from a Certificate Authority. To obtain the certificate, create a Certificate Signing Request and send it to the Certificate Authority, as described in Section 3.8.1, "Obtain an SSL Certificate for Satellite Server" . In return, you receive a signed SSL certificate.

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

1. Section 3.8.1, "Obtain an SSL Certificate for Satellite Server"
2. Section 3.8.2, "Validate the Satellite Server's SSL Certificate"
3. Section 3.8.3, "Run the Satellite Installer with Custom Certificate Parameters"
4. Section 3.8.4, "Install the New Certificate on all Hosts Connected to the Satellite Server"
5. If you have external Capsule Servers registered to the Satellite Server, proceed to Configuring Capsule Server with a Custom Server Certificate in the *Installing Capsule Server* guide to configure the Capsule Servers to use a custom certificate.

3.8.1. Obtain an SSL Certificate for Satellite Server

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

IMPORTANT

Use PEM encoding for the SSL Certificates.

Procedure

To obtain custom SSL certificates for Satellite Server, complete the following steps:

1. Create a directory to store all the source certificate files, accessible only to the **root** user, for example **/root/sat_cert**.

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

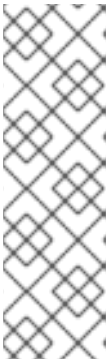
2. Create a private key with which to sign the Certificate Signing Request (CSR).

NOTE

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

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

3. Create the **/root/sat_cert/openssl.cnf** configuration file for the Certificate Signing Request (CSR) and include the following content. In the **[req_distinguished_name]** section, enter information about your organization.



NOTE

The certificate's Common Name (CN) and the Subject Alternative Name (SAN) DNS.1 must match the fully-qualified domain name (FQDN) of the server on which it is used. If you are requesting a certificate for a Satellite Server, this is the FQDN of Satellite Server. If you are requesting a certificate for a Capsule Server, this is the FQDN of Capsule Server.

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

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

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

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

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

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

4. Generate the Certificate Signing Request (CSR):

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

5. Send the certificate signing request to the Certificate Authority. The same Certificate Authority must sign certificates for Satellite Server and Capsule Server.
When you submit the request, specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the Certificate Authority for the preferred method. In response to the request you can expect to receive a Certificate Authority bundle, and a signed

certificate, in separate files.

3.8.2. Validate the Satellite Server's SSL Certificate

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

1. Validate the custom SSL certificate input files. Change the files' names to match your files. 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/sat_cert/satellite_cert.pem \ 1
  -k /root/sat_cert/satellite_cert_key.pem \ 2
  -b /root/sat_cert/ca_cert_bundle.pem 3
```

1. Certificate file for the Satellite Server, signed by your Certificate Authority
2. Satellite Server's private key, used to sign the certificate
3. Certificate Authority bundle

3.8.3. Run the Satellite Installer with Custom Certificate Parameters

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

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

The commands in this section are output by the **katello-certs-check** command, as detailed in Section 3.8.2, "Validate the Satellite Server's SSL Certificate" , and can be copied and pasted into a terminal.

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

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

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

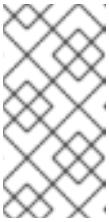
   Important parameters in this command include **--certs-update-server** and **--certs-update-server-ca**, which specify that the server's SSL certificate and certificate authority are to be updated. For a brief description of all the installer's parameters, enter the command: **satellite-installer --scenario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

NOTE

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

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

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



NOTE

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

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

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

Now that the custom SSL certificate has been installed on the Satellite Server, it must also be installed on every host registered to the Satellite Server.

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

Procedure

Enter the following commands on all applicable hosts.

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

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

2. Install the custom SSL certificate on the host.

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

3.9. USING EXTERNAL DATABASES WITH SATELLITE

As part of the installation process for Red Hat Satellite, the **satellite-installer** command installs MongoDB and PostgreSQL databases on the same server as Satellite. In certain Satellite deployments, using external databases can help with the server load. However, there are many factors that can affect Satellite Server's performance. Moving to an external database might not help your specific problem.

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

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

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

3.9.1. MongoDB as an External Database Considerations

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

Advantages of External MongoDB

- Increase in free memory and free CPU on Satellite
- Flexibility to tune the MongoDB server's system without adversely affecting Satellite operations

Disadvantages of External MongoDB

- Increase in deployment complexity that can make troubleshooting more difficult
- An external MongoDB server is an additional system to patch and maintain
- If either the Satellite or the Mongo database server suffers a hardware or storage failure, Satellite is not operational
- If there is latency between the Satellite and the external database server, performance can suffer

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

3.9.2. PostgreSQL as an External Database Considerations

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

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.

3.9.3. Overview

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

1. Use Section 1.2, "Storage Requirements and Guidelines" to plan the storage requirements for your external databases
2. Prepare PostgreSQL with databases for Foreman and Candlepin and dedicated users owning them
3. Prepare MongoDB with user **pulp** owning the **pulp_database**
4. Follow the initial steps to install Satellite and ensure that the databases are accessible from Satellite
5. Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**

Preparing Red Hat Enterprise Linux Server 7 for Database Installation

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

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

1. Use the instructions in Identifying and Attaching the Satellite Subscription to the Host to attach a Satellite subscription to your server.
2. To install MongoDB and PostgreSQL servers on Red Hat Enterprise Linux Server 7, you must disable all repositories and enable only the following repositories:

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

3.9.4. Installing MongoDB

You can install only the same version of MongoDB that is installed with the **satellite-installer** tool during an internal database installation. You can install MongoDB using Red Hat Software Collections

(RHSCL) repositories or from an external source, as long as the version is supported. Satellite supports MongoDB version 3.4.

1. To install MongoDB, enter the following command:

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

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

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

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

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

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

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

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

```
security:
  authorization: enabled
```

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

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

7. Open port 27017 for MongoDB:

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

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

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

3.9.5. Installing PostgreSQL

You can install only the same version of PostgreSQL that is installed with the **satellite-installer** tool during an internal database installation. Satellite supports only a specific version of PostgreSQL that is available through Red Hat Enterprise Linux Server 7 repositories. You can install PostgreSQL using **rhel-7-server-rpms** repositories or from an external source, as long as the version is supported. For more information about the repository that contains the supported version of PostgreSQL, and what version is supported, see the Package Manifest.

1. To install PostgreSQL, enter the following command:

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

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

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

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

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

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

```
listen_addresses = '*'
```

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

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

6. Add the following line to the file:

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

7. Restart PostgreSQL service to update with the changes:

```
# systemctl restart postgresql
```

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

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

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

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

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

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

11. 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"
```

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

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

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

For PostgreSQL, enter the following command:

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

For MongoDB, enter the following command:

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

3.10. RESTRICTING ACCESS TO MONGOD

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

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

1. Configure the Firewall.

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

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

2. Make the changes persistent:

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

CHAPTER 4. CONFIGURING EXTERNAL SERVICES

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

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

4.1. CONFIGURING SATELLITE WITH EXTERNAL DNS

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

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

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

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

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

controls  {
    inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "capsule"; };
    inet 192.168.38.2 port 953 allow { 192.168.38.1; 192.168.38.2; } keys { "capsule"; };
};

options  {
    directory "/var/named";
    forwarders { 8.8.8.8; 8.8.4.4; };
};

include "/etc/named.rfc1912.zones";

zone "38.168.192.in-addr.arpa" IN {
    type master;
    file "dynamic/38.168.192-rev";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};

zone "virtual.lan" IN {
    type master;
    file "dynamic/virtual.lan";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};
```

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

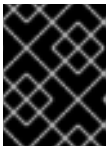
3. Create a key file.

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

This command can take a long time to complete.

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

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



IMPORTANT

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

5. Create zone files.

```
# cat /var/named/dynamic/virtual.lan
$ORIGIN .
$TTL 10800      ; 3 hours
virtual.lan             IN SOA  service.virtual.lan. root.virtual.lan. (
                                9          ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                3600       ; minimum (1 hour)
                                )
                        NS      service.virtual.lan.
$ORIGIN virtual.lan.
$TTL 86400      ; 1 day
capsule                 A       192.168.38.1
service                 A       192.168.38.2
```

6. Create the reverse zone file.

```
# cat /var/named/dynamic/38.168.192-rev
$ORIGIN .
$TTL 10800      ; 3 hours
38.168.192.in-addr.arpa IN SOA  service.virtual.lan. root.38.168.192.in-addr.arpa. (
                                4          ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                3600       ; minimum (1 hour)
                                )
                        NS      service.virtual.lan.
$ORIGIN 38.168.192.in-addr.arpa.
```

```
$TTL 86400      ; 1 day
1               PTR     capsule.virtual.lan.
2               PTR     service.virtual.lan.
```

There should be no extra non-ASCII characters.

4.2. VERIFYING AND STARTING THE DNS SERVICE

1. Validate the syntax.

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

2. Start the server.

```
# systemctl restart named
```

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

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. Test that the DNS service can resolve the new host.

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

5. If necessary, delete the new entry.

```
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

6. Configure the firewall for external access to the DNS service (UDP and TCP on port 53).

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

4.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP

You can use this section to configure your Red Hat Satellite Server to work with an external DHCP server.

To configure the DHCP server and share the DHCP configuration and lease files

1. Deploy a Red Hat Enterprise Linux Server and install the ISC DHCP Service and Berkeley Internet Name Domain (BIND).

```
# yum install dhcp bind
```

2. Generate a security token in an empty directory.

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

The above command can take a long time, for less-secure proof-of-concept deployments you can use a non-blocking random number generator.

```
# dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
```

This creates the key pair in two files in the current directory.

3. Copy the secret hash from the key.

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

4. Edit the **dhcpd** configuration file for all of the subnets and add the key as in the example:

```
# cat /etc/dhcp/dhcpd.conf
default-lease-time 604800;
max-lease-time 2592000;
log-facility local7;

subnet 192.168.38.0 netmask 255.255.255.0 {
 range 192.168.38.10 192.168.38.100;
 option routers 192.168.38.1;
 option subnet-mask 255.255.255.0;
 option domain-search "virtual.lan";
 option domain-name "virtual.lan";
 option domain-name-servers 8.8.8.8;
}

omapi-port 7911;
key omapi_key {
 algorithm HMAC-MD5;
 secret "jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

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

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

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

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

8. Determine the UID and GID numbers of the foreman user on the Satellite Server.

```
# id -u foreman
```

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

9. Create the same user and group with the same IDs on the DHCP server.

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

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

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

11. Start the DHCP service.

```
# systemctl start dhcpd
```

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

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

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

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

14. Add the following line to the **/etc/fstab** file to create mount points for the newly created directories.

```
/var/lib/dhcpd /exports/var/lib/dhcpd none bind,auto 0 0
/etc/dhcp /exports/etc/dhcp none bind,auto 0 0
```

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

```
# mount -a
```

16. Ensure the following lines are present in /etc/exports:

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)

/exports/etc/dhcp 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)

/exports/var/lib/dhcpd 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

17. Reload the NFS server.

```
# exportfs -rva
```

18. Configure the firewall for the DHCP omapi port 7911 for the Satellite Server.

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

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

- Use the **firewalld** daemon's NFS service to configure the firewall.

```
# firewall-cmd --zone public --add-service mountd \
&& firewall-cmd --zone public --add-service rpc-bind \
&& firewall-cmd --zone public --add-service nfs \
&& firewall-cmd --runtime-to-permanent
```

To configure the Satellite Server

1. Install the NFS client.

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

2. Create the DHCP directories for NFS.

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

3. Change the file owner.

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

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

```
# showmount -e your_DHCP_server_FQDN
# rpcinfo -p your_DHCP_server_FQDN
```

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

```
your_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

your_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. Read the relevant files.

```
# su foreman-proxy -s /bin/bash
bash-4.2$ cat /mnt/nfs/etc/dhcp/dhcpd.conf
bash-4.2$ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases
bash-4.2$ exit
```

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

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=your_DHCP_server_FQDN
```

9. Restart the foreman-proxy service.

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

10. Log in to the Satellite Server web UI.

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

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

4.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

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

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

Before You Begin

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

Configure Satellite Server with External TFTP

1. Install and enable the TFTP server.

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

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

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

3. Configure the PXELinux environment.

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

4. Restore SELinux file contexts.

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

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

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

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

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

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

```
# mount -a
```

8. Ensure the following lines are present in **/etc/exports**:

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)
```

```
/exports/var/lib/tftpboot 192.168.38.1(rw,async,no_root_squash,no_subtree_check,nohide)
```

The first line is common to the DHCP configuration and therefore should already be present if the previous procedure was completed on this system.

9. Reload the NFS server.

```
# exportfs -rva
```

4.4.1. Configuring the Firewall for External Access to TFTP

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

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

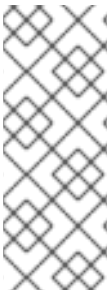
4.5. CONFIGURING SATELLITE OR CAPSULE WITH EXTERNAL IDM DNS

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

The first method is to install the IdM client which automates the process with the *generic security service algorithm for secret key transaction* (GSS-TSIG) technology defined in RFC3645. This method requires installing the IdM client on the Satellite Server or Capsule's base system and having an account created by the IdM server administrator for use by the Satellite administrator. See Section 4.5.1, "Configuring Dynamic DNS Update with GSS-TSIG Authentication" to use this method.

The second method, *secret key transaction authentication for DNS* (TSIG), uses an **rndc.key** for authentication. It requires root access to the IdM server to edit the BIND configuration file, installing the

BIND utility on the Satellite Server's base system, and coping the **rndc.key** to between the systems. This technology is defined in RFC2845. See Section 4.5.2, "Configuring Dynamic DNS Update with TSIG Authentication" to use this method.



NOTE

You are not required to use Satellite to manage DNS. If you are using the Realm enrollment feature of Satellite, where provisioned hosts are enrolled automatically to IdM, then the **ipa-client-install** script creates DNS records for the client. The following procedure and Realm enrollment are therefore mutually exclusive. For more information on configuring Realm enrollment, see External Authentication for Provisioned Hosts in *Administering Red Hat Satellite* .

Determining where to install the IdM Client

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

4.5.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication

In this example, Satellite Server has the following settings.

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

The IdM server has the following settings.

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

Before you Begin.

1. Confirm the IdM server is deployed and the host-based firewall has been configured correctly. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.
2. Obtain an account on the IdM server with permissions to create zones on the IdM server.
3. Confirm if the Satellite or an external Capsule is managing DNS for a domain.
4. Confirm that the Satellite or external Capsule are currently working as expected.
5. In the case of a newly installed system, complete the installation procedures in this guide first. In particular, DNS and DHCP configuration should have been completed.
6. Make a backup of the answer file in case you have to revert the changes. See Specifying Installation Options for more information.

Create a Kerberos Principal on the IdM Server.

1. Ensure you have a Kerberos ticket.

```
# kinit idm_user
```

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

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

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

Install and Configure the IdM Client.

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

1. Install the IdM client package.

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

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

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

3. Ensure you have a Kerberos ticket.

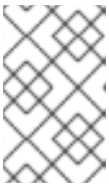
```
# kinit admin
```

4. Remove any preexisting keytab.

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

5. Get the keytab created for this system.

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



NOTE

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

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

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

7. If required, check the keytab is valid.

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

Configure DNS Zones in the IdM web UI.

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

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

   e. Ensure **Dynamic update** is set to **True**.
   f. Enable **Allow PTR sync**.
   g. Select **Save** to save the changes.
2. Create and Configure the reverse zone.
   a. Navigate to **Network Services** > **DNS** > **DNS Zones**.
   b. Select **Add**.
   c. Select **Reverse zone IP network** and add the network address in CIDR format to enable reverse lookups.
   d. Click **Add and Edit**
   e. On the **Settings** tab, in the **BIND update policy** box, add an entry as follows to the semi-colon separated list:

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

   f. Ensure **Dynamic update** is set to **True**.
   g. Select **Save** to save the changes.

Configure the Satellite or Capsule Server Managing the DNS Service for the Domain.

- On a Satellite Server's Base System.

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

- On a Capsule Server's Base System.

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

Restart the Satellite or Capsule's Proxy Service.

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

Update the Configuration in Satellite web UI.

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

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

4.5.2. Configuring Dynamic DNS Update with TSIG Authentication

In this example, Satellite Server has the following settings.

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

The IdM server has the following settings.

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

Before you Begin

1. Confirm the IdM Server is deployed and the host-based firewall has been configured correctly. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.
2. Obtain **root** user privileges on the IdM server.
3. Confirm if the Satellite or an external Capsule is managing DNS for a domain.
4. Confirm that the Satellite or external Capsule are currently working as expected.
5. In the case of a newly installed system, complete the installation procedures in this guide first. In particular, DNS and DHCP configuration should have been completed.
6. Make a backup of the answer file in case you have to revert the changes. See Specifying Installation Options for more information.

Enabling External Updates to the DNS Zone in the IdM Server

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

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

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

```
# systemctl reload named
```

3. In the IdM web UI, go to **Network Services** > **DNS** > **DNS Zones**. Select the name of the zone. On the **Settings** tab:
   a. Add the following in the **BIND update policy** box.

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

   b. Ensure **Dynamic update** is set to **True**.
   c. Click **Update** to save the changes.

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

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

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

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

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

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="192.168.25.2" \
--foreman-proxy-keyfile=/etc/rndc.key \
--foreman-proxy-dns-ttl=86400
```

Testing External Updates to the DNS Zone in the IdM Server

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

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

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

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

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

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

4. On Satellite Server, test the DNS entry.

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

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

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

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

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

7. Confirm that the DNS entry was removed.

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

The above **nslookup** command fails and outputs the SERVFAIL error message if the record was successfully deleted.

4.5.3. Reverting to Internal DNS Service

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

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

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

```
# satellite-installer
```

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

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

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

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

Update the Configuration in Satellite web UI.

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

1. Navigate to **Infrastructure** > **Capsules**.
2. For each Capsule to be updated, from the **Actions** drop-down menu, select **Refresh**.
3. Configure the domain:
   a. Go to **Infrastructure** > **Domains** and select the domain name.
   b. On the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.

4. Configure the subnet:

    a. Go to **Infrastructure** > **Subnets** and select the subnet name.

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

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

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

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

CHAPTER 5. UNINSTALLING SATELLITE SERVER

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

Uninstalling Satellite Server erases all applications used on the target system. If you use any applications or application data for purposes other than Satellite Server, you should back up the information before the removal process.

Before you Begin

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

WARNING

This script erases many packages and config files, such as the following important packages:

- httpd (apache)
- mongodb
- tomcat6
- puppet
- ruby
- rubygems
- All Katello and Foreman Packages

Uninstall Satellite Server

1. Uninstall Satellite Server.

```
# katello-remove
```

CHAPTER 6. RUNNING RED HAT SATELLITE ON AMAZON WEB SERVICES

Use this guide to ensure that you make all the necessary preparations for installing Red Hat Satellite Server and Capsules in Amazon Web Services (AWS) Elastic Compute Cloud (Amazon EC2).

Use the Deployment Scenarios section to understand the different architecture setups that are available for Satellite and Capsule installation on AWS.

Use the Prerequisites section to prepare your Red Hat and Amazon Web resources for the Red Hat Satellite installation.

Subscriptions

Not all subscriptions are eligible to run in public cloud environments. For more information about subscription eligibility, see the Cloud Access Page. You can create additional organizations and then import additional manifests to the organizations. For more information, see Creating an Organization in the *Content Management Guide*.

6.1. USE CASE CONSIDERATIONS

Because Amazon Web Services is an image-only service, there are common Satellite use cases that do not work, or require extra configuration in an Amazon Web Service environment. If you plan to use Satellite on AWS, ensure that the use case scenarios that you want to use are available in an AWS environment.

6.1.1. Use Cases Known to Work

You can perform the following Red Hat Satellite use cases on AWS:

- Subscription Management
- Content Management
- Errata Management
- Configuring Hosts
- Red Hat Insights
- Provisioning Containers
- Realm Integration via IdM
- OpenSCAP
- Remote Execution

Multi-homed Satellite and Capsule

If you want Satellite to use multiple interfaces with distinct host names, you must perform additional configuration of the Satellite Server and Satellite Capsule Server CA certificates. If you want to deploy Satellite in this configuration, contact Red Hat.

You must do this when Satellite Server or Capsule Server has different internal and external DNS host names and there is no site-to-site VPN connection between the locations where you deploy Satellite Server and Capsule Server.

On demand content sources

You can use the **On demand** download policy to reduce the storage footprint of the Red Hat Enterprise Linux server that runs Satellite. When you set the download policy to **On Demand**, content syncs to the Satellite Server or Capsule Server when a content host requests it.

For more information, see Importing Red Hat Content in the *Content Management Guide*.

6.1.2. Use Cases that Do Not Work

In AWS, you cannot manage the DHCP. Because of this, most of Satellite Server's kickstart and PXE provisioning models are unusable. This includes:

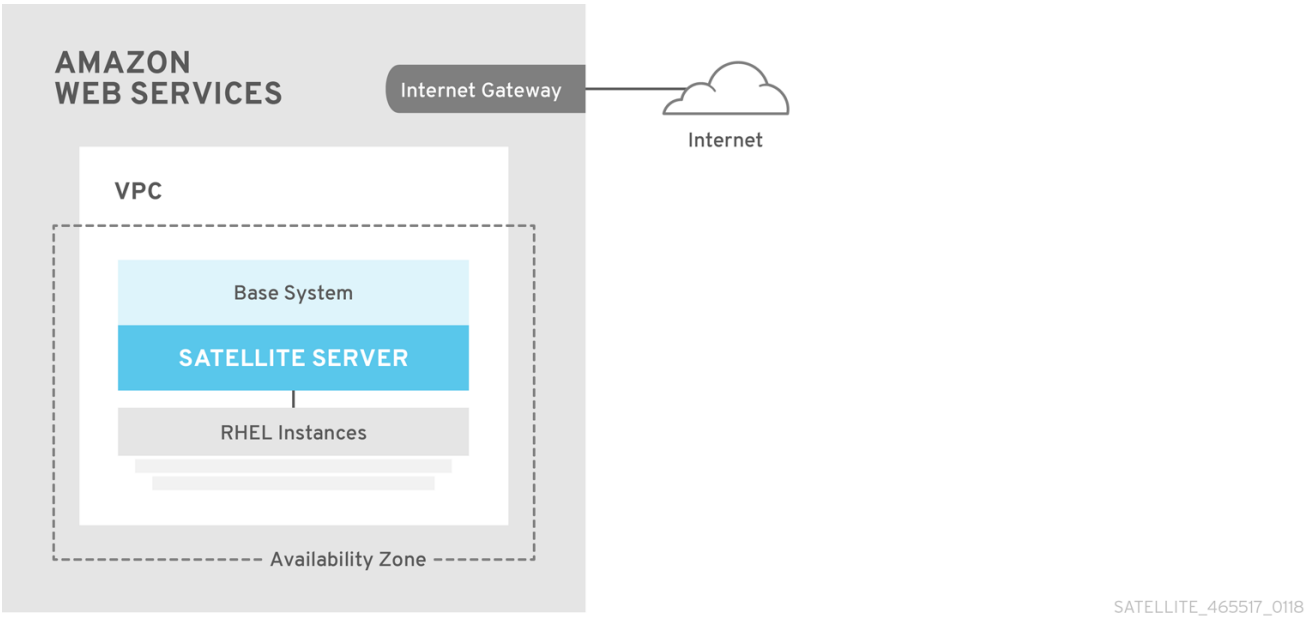
- PXE Provisioning
- Discovery and Discovery Rules
- ISO Provisioning methods.
  - PXE-Less Discovery (iPXE)
  - Per-host ISO
  - Generic ISO
  - Full-host ISO

6.2. DEPLOYMENT SCENARIOS

There are three deployment scenarios for Red Hat Satellite in Amazon Web Services:

1. One region setup
2. Connecting on-premise and AWS region
3. Connecting different regions

Scenario 1: One region setup



The least complex configuration of Satellite Server in Amazon Web Services consists of both the Satellite Server and the content hosts residing within the same region and within the Virtual Private Cloud (VPC).

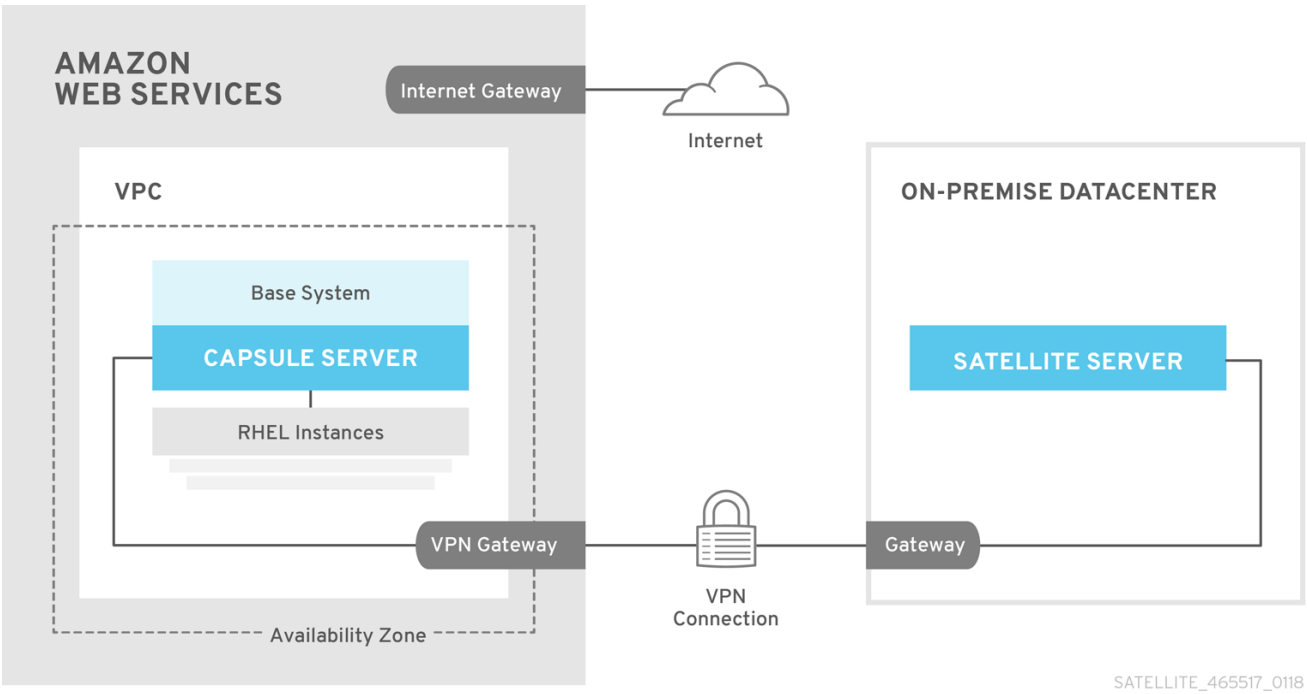
You can also use a different availability zone.

Scenario 2: Connecting on-premise and AWS region

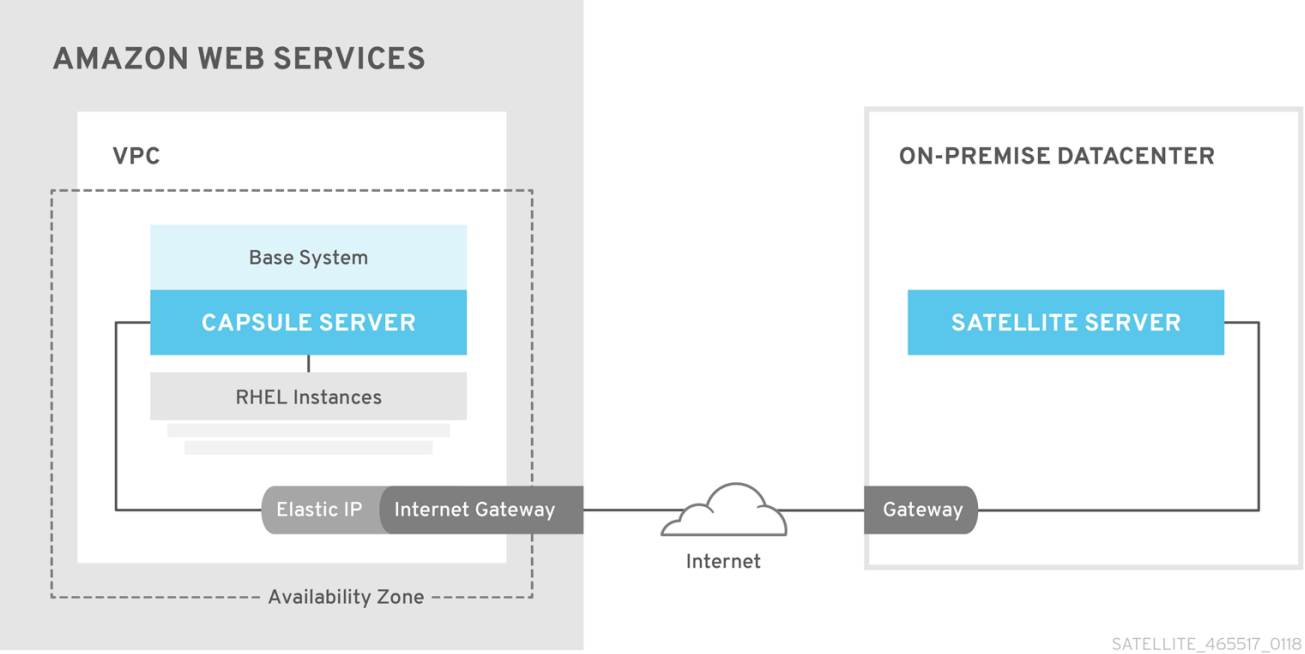
Create a VPN connection between the on-premise location and the AWS region where the Capsule resides.

It is also possible to use the external host name of Satellite Server when you register the instance which runs Capsule Server.

Option 1: Site-to-Site VPN connection between the AWS region and the On-Premise Datacenter



Option 2: Direct connection using the External DNS host name



Scenario 3: Connecting different regions

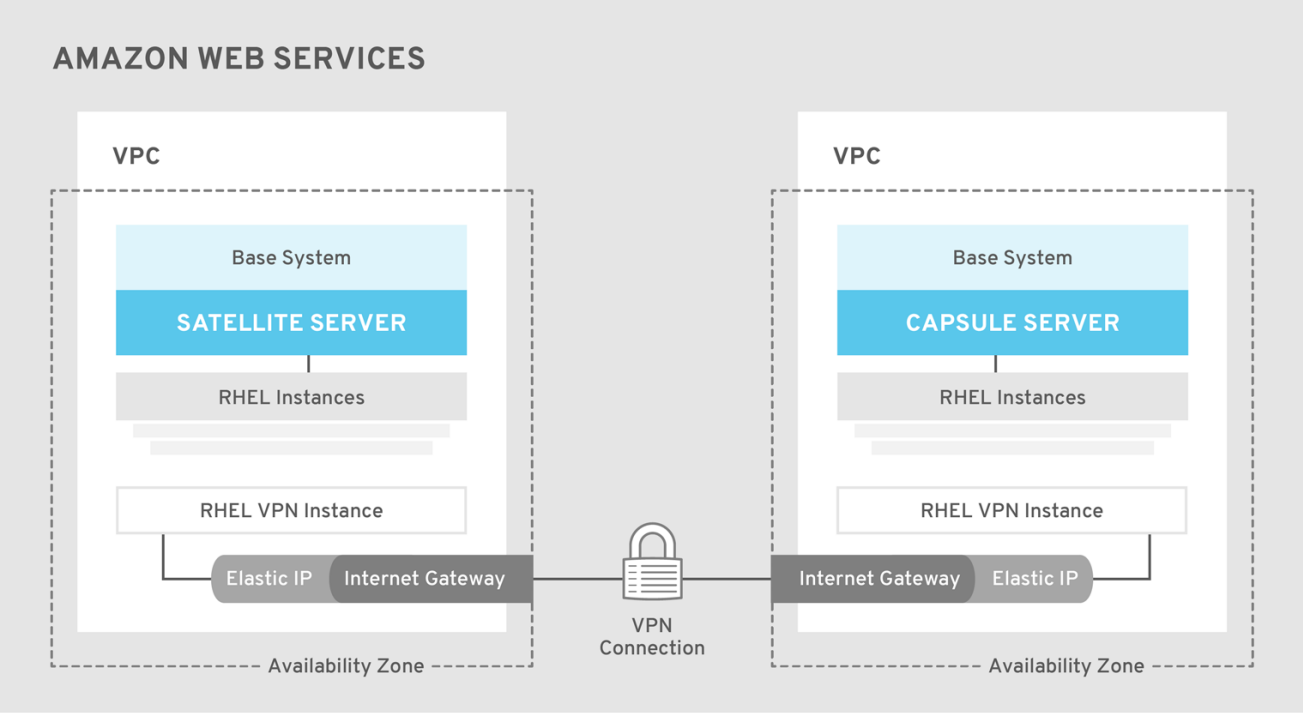
Create a site-to-site VPN connection between the different regions so that you can use the Internal DNS host name when you register the instance that runs Capsule Server to the Satellite Server.

If you do not establish a site-to-site VPN connection, use the external DNS host name when you register the instance that runs Capsule Server to the Satellite Server.

NOTE

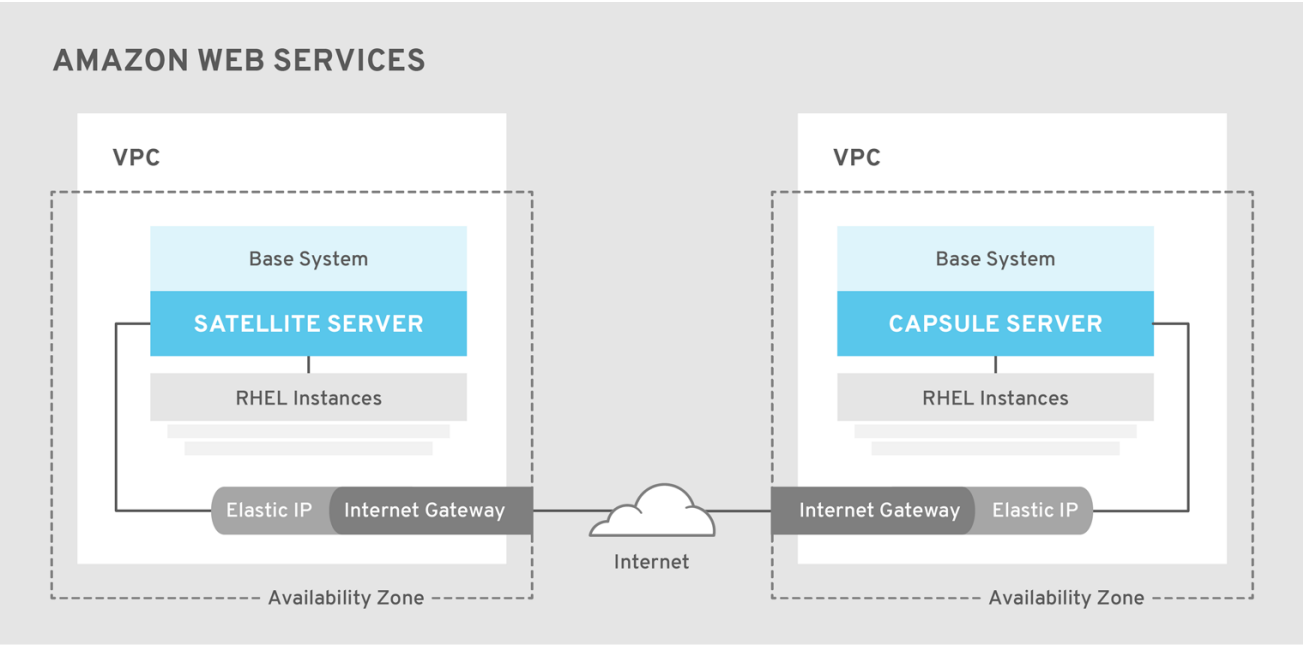
Most Public Cloud Providers do not charge for data being transferred into a region, or between availability zones within a single region; however, they do charge for data leaving the region to the Internet.

Option 1: Site-to-Site VPN connection between AWS regions



SATELLITE_465517_0118

Option 2: Direct connection using the External DNS host name



SATELLITE_465517_0118

6.3. PREREQUISITES

Before you can install and register Red Hat Satellite and Capsule, you must set up accounts with Amazon Web Services (AWS) and create and start Red Hat Enterprise Linux instances on AWS.

6.3.1. Amazon Web Service Assumptions

To use this guide, you must have a working knowledge of the following aspects of Amazon Web Services:

- Creating and accessing Red Hat Enterprise Linux images in AWS

- Editing network access in AWS Security
- Creating EC2 instances and how to create EBS volumes
- Launching instances
- Importing and exporting virtual machines in AWS.
- Using AWS Direct Connect

To install Satellite and Capsule in an AWS environment, you must ensure that your AWS set up meets the Section 1.1, "System Requirements" for Satellite and Capsule.

For more information about Amazon Web Services and terminology, see Amazon Elastic Compute Cloud Documentation.

For more information about Amazon Web Services Direct Connect, see What is AWS Direct Connect?

6.3.2. Red Hat Cloud prerequisites

To use this guide, you must complete the following steps:

- Register with Red Hat Cloud Access.
- Migrate any Red Hat subscriptions that you want to use.
- Create an AWS instance and deploy a Red Hat Enterprise Linux virtual machine to the instance.
- Ensure that your subscriptions are eligible for transfer to Red Hat Cloud. For more information, see Red Hat Cloud Access Program Details .

For more information about deploying Red Hat Enterprise Linux in AWS, see How to Locate Red Hat Cloud Access Gold Images on AWS EC2.

6.3.3. Red Hat Satellite-specific prerequisites

- Ensure that the Amazon EC2 instance type meets or exceeds the system and storage requirements in Chapter 1, *Preparing your environment for installation*. For the best performance, use an AWS Storage Optimized instance.
- Use Chapter 1, *Preparing your environment for installation* to understand and assign the correct storage to your AWS EBS volumes.
- Store the synced content on an EBS volume that is separate to the boot volume.
- Mount the synced content EBS volume separately in the operating system.
- Optional: store other data, for example, the **mongodb** directory on a separate EBS volume.
- If you want the Satellite Server and Capsule Server to communicate using external DNS hostnames, open the required ports for communication in the AWS Security Group that is associated with the instance.

6.3.4. Preparing for the Red Hat Satellite Installation

In your AWS environment, complete the following steps:

1. Launch an EC2 instance of a Red Hat Enterprise Linux AMI
2. Connect to the newly created instance.

If you use a Red Hat Gold Image, remove the RHUI client and set the **enabled** parameter in the **product-id.conf** to **1**.

```
# yum -y remove rh-amazon-rhui-client*
# yum clean all
# cat << EOF > /etc/yum/pluginconf.d/product-id.conf
> [main]
> enabled=1
> EOF
```

6.4. INSTALLING SATELLITE SERVER ON AWS

On your AWS environment, complete the following steps:

1. Connect to the new instance.
2. Use Installing Satellite Server to install Satellite Server.

6.5. INSTALLING CAPSULE ON AWS

On your AWS environment, complete the following steps:

1. Connect to the new instance.
2. Install Capsule Server. For more information, see Installing Capsule Server.

6.6. REGISTERING HOSTS TO SATELLITE USING THE BOOTSTRAP SCRIPT

When you install Satellite Server and Capsule Server, you must then register the content hosts on EC2 instances to Satellite with the bootstrap script.

For more information about using the bootstrap script, see Registering Hosts to Satellite Using The Bootstrap Script in the *User Guide*.

Install the Katello Agent. For more information, see Installing the katello Agent.

APPENDIX A. LARGE DEPLOYMENT CONSIDERATIONS

Increasing the Maximum Number of File Descriptors for Apache

With more than 800 content hosts registered, Apache can reach several system-level limits, resulting in new content host registration failure. To avoid this, file descriptor limits must be increased before deploying a large number of content hosts.

1. Create the **/etc/systemd/system/httpd.service.d/limits.conf** file and insert the following text:

```
[Service]
LimitNOFILE=65536
```

2. Apply the changes to the unit.

```
# systemctl daemon-reload
```

3. Restart Satellite services.

```
# foreman-maintain service restart
```

Increasing the Maximum Number of File Descriptors for qpid

With more than 1100 content hosts with goferd running for errata updates, the qpid reach system-level limits, resulting in registration failures. To avoid this, file descriptors limits must be increased before deploying a large number of content hosts.

Increasing the Maximum Number of File Descriptors for qpid

1. Create the **/etc/systemd/system/qpidd.service.d/limits.conf** file and insert the following text:

```
[Service]
LimitNOFILE=65536
```

2. Apply the changes to the unit.

```
# systemctl daemon-reload
# systemctl restart qpidd.service
```

Increasing the Shared Buffer and Work Memory

You can increase the **shared_buffer** and **work_mem** to **256M** and **4M** respectively.

1. On Red Hat Enterprise Linux 7, create the **/var/lib/pgsql/data/postgresql.conf** file and insert the following text:

```
work_mem = 4MB
shared_buffers = 256MB
```

2. Restart postgresql services.

```
# systemctl restart postgresql
```

Increasing Concurrent Content Host Registrations

To avoid reaching system-level limits, you can increase the global passenger queue limit to accommodate up to 250 concurrent content hosts.

To increase the global passenger queue limit, complete the following steps:

1. Adjust the maximum passenger pool size to 1.5 times the physical CPU cores available to the Satellite Server.
   For example, if you have a Satellite Server with 16 cores, then the maximum passenger pool size is 24. This number is referenced as an example and you should use the number applicable to your environment.

2. Edit the **/etc/httpd/conf.d/passenger.conf** file, updating the **IfModule** stanza to match the following text:

```
<IfModule mod_passenger.c>
  PassengerRoot /usr/share/gems/gems/passenger-
4.0.18/lib/phusion_passenger/locations.ini
  PassengerRuby /usr/bin/ruby
  PassengerMaxPoolSize 24
  PassengerMaxRequestQueueSize 200
  PassengerStatThrottleRate 120
</IfModule>
```

3. Edit the Foreman Passenger application configuration file **/etc/httpd/conf.d/05-foreman-ssl.conf**, updating the stanza starting with **PassengerAppRoot** to match the following text:

```
PassengerAppRoot /usr/share/foreman
PassengerRuby /usr/bin/tfm-ruby
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com
```

4. Edit the Puppet Passenger application configuration file **/etc/httpd/conf.d/25-puppet.conf**, adding the following text to the end of the virtual host definition:

```
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com:8140
```

5. Change the maximum connections in the **/var/lib/pgsql/data/postgresql.conf** file.

```
max_connections = 500
```

6. Restart postgresql services.

```
# systemctl restart postgresql
```

Increasing the maximum number of open files for qdrouterd

With more than 1000 content hosts registered, **qdrouterd** can reach the default maximum number of open files. To avoid this, increase the maximum number of open files on the Satellite Server and all external Capsule Servers.

1. Calculate the required maximum number of open files, using the following equation.

```
(3 x number of content hosts) + 100
```

   For example, with 1020 content hosts, the new maximum should be set to 3160 ((3 x 1020) + 100).

2. Create the file **/etc/systemd/system/qdrouterd.service.d/limits.conf** and add the following text.

```
[Service]
LimitNOFILE=maximum_number_of_files
```

   a. Apply the changes to the unit.

```
# systemctl daemon-reload
```

   b. Restart Satellite services.

```
# foreman-maintain service restart
```

Reducing delays in host registration

Communication between Satellite and each registered host is secured by use of certificates. When a host is registered, Satellite creates two certificates, an identity certificate and a Puppet certificate. The algorithm used to create each certificate requires random data from the Red Hat Enterprise Linux kernel. If not enough entropy is available when a host is registered, there is a delay until a suitable level of entropy is available. In very large environments, with more than 10,000 hosts, the rate of host registration is likely to be slowed by the lack of entropy. Several methods can be used to improve the availability of entropy to the Linux kernel, and so reduce the risk to performance of Satellite.

By default, the Linux kernel uses the **/dev/random** device as the source of random numbers. This is a blocking device, which means it stops supplying numbers when it determines that the amount of entropy is insufficient for generating a properly random output. It is this wait time which causes the delay in registering hosts. To resolve this issue, use the **/dev/urandom** device, as this is a non-blocking device.

Some hardware servers have processors which include hardware random number generators. For those that are supported by the Red Hat Enterprise Linux kernel, you can use that as the source of random numbers. For more information, see the hardware vendor's documentation.

If Satellite is hosted on a virtual machine, note that some hypervisors make the hardware server's random number generator available to the virtual machines it hosts. Red Hat Virtualization features the virtio RNG (Random Number Generator) device that provides KVM virtual machines access to entropy from the Red Hat Virtualization Host. On guests running Red Hat Enterprise Linux 7.0, you must install and configure **rngd**. On guests running Red Hat Enterprise Linux 7.1 and later, the guest kernel fetches entropy from the host as required. If a host's random number generator is shared by guests, use of a hardware random number generator is recommended.

For more information about random number generators for guests, see Random Number Generator Device in the Red Hat Enterprise Linux 7 *Virtualization Deployment and Administration Guide*. For other hypervisors, see the vendor's documentation.

For more information about the random number generator daemon, **rngd**, see Using the Random Number Generator in the Red Hat Enterprise Linux 7 *Security Guide*.

APPENDIX B. APPLYING CUSTOM CONFIGURATION TO RED HAT SATELLITE

When you install and configure Satellite for the first time using **satellite-installer**, you can specify that the DNS and DHCP configuration files are not to be managed by Puppet using the installer flags **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false**. If these flags are not specified during the initial installer run, rerunning of the installer overwrites all manual changes, for example, rerun for upgrade purposes. If changes are overwritten, you must run the restore procedure to restore the manual changes. For more information, see How to Restore Manual Changes Overwritten by a Puppet Run in *Installing Satellite Server from a Connected Network* .

To view all installer flags available for custom configuration, run **satellite-installer --scenario satellite --full-help**. Some Puppet classes are not exposed to the Satellite installer. To manage them manually and prevent the installer from overwriting their values, specify the configuration values by adding entries to configuration file **/etc/foreman-installer/custom-hiera.yaml**. This configuration file is in YAML format, consisting of one entry per line in the format of **<puppet class>::<parameter name>: <value>**. Configuration values specified in this file persist across installer reruns.

Common examples include:

- For Apache, to set the ServerTokens directive to only return the Product name:

```
apache::server_tokens: Prod
```

- To turn off the Apache server signature entirely:

```
apache::server_signature: Off
```

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

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

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

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

B.1. HOW TO RESTORE MANUAL CHANGES OVERWRITTEN BY A PUPPET RUN

If your manual configuration has been overwritten by a Puppet run, you can restore the files to the previous state. The following example shows you how to restore a DHCP configuration file overwritten by a Puppet run.

1. Copy the file you intend to restore. This allows you to compare the files to check for any mandatory changes required by the upgrade. This is not common for DNS or DHCP services.

```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.backup
```

2. Check the log files to note down the md5sum of the overwritten file. For example:

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

3. Restore the overwritten file:

```
# puppet filebucket restore --local --bucket \
/var/lib/puppet/clientbucket /etc/dhcp/dhcpd.conf \ 622d9820b8e764ab124367c68f5fa3a1
```

4. Compare the backup file and the restored file, and edit the restored file to include any mandatory changes required by the upgrade.