



Red Hat Satellite 6.7

Installing Satellite Server from a Connected Network

Installing Red Hat Satellite Server from a Connected Network

Red Hat Satellite 6.7 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

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This 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	4
1.1. SYSTEM REQUIREMENTS	4
1.2. STORAGE REQUIREMENTS	5
1.3. STORAGE GUIDELINES	5
1.4. SUPPORTED OPERATING SYSTEMS	7
1.5. SUPPORTED BROWSERS	7
1.6. PORTS AND FIREWALLS REQUIREMENTS	8
1.7. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER	10
1.8. VERIFYING FIREWALL SETTINGS	11
1.9. VERIFYING DNS RESOLUTION	11
CHAPTER 2. INSTALLING SATELLITE SERVER	13
2.1. REGISTERING TO RED HAT SUBSCRIPTION MANAGEMENT	13
2.2. ATTACHING THE SATELLITE INFRASTRUCTURE SUBSCRIPTION	13
2.3. CONFIGURING REPOSITORIES	15
2.4. INSTALLING THE SATELLITE SERVER PACKAGES	15
2.5. SYNCHRONIZING THE SYSTEM CLOCK WITH CHRONYD	16
2.6. INSTALLING THE SOS PACKAGE ON THE BASE OPERATING SYSTEM	16
2.7. CONFIGURING SATELLITE SERVER	16
2.7.1. Configuring Satellite Manually	17
2.7.2. Configuring Satellite Automatically using an Answer File	18
2.8. IMPORTING A SUBSCRIPTION MANIFEST INTO SATELLITE SERVER	18
CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER	20
3.1. USING RED HAT INSIGHTS WITH SATELLITE SERVER	20
3.2. ENABLING THE SATELLITE TOOLS 6.7 REPOSITORY	20
3.3. SYNCHRONIZING THE SATELLITE TOOLS 6.7 REPOSITORY	21
3.4. CONFIGURING SATELLITE SERVER WITH AN HTTP PROXY	21
3.4.1. Adding a Default HTTP Proxy to Satellite	21
3.4.2. Configuring the HTTP Proxy to Connect to Red Hat CDN	22
3.4.3. Configuring SELinux to Ensure Access to Satellite on Custom Ports	23
3.4.4. Using an HTTP Proxy for all Satellite HTTP Requests	24
3.4.5. Excluding Hosts from Receiving Proxied Requests	24
3.4.6. Resetting the HTTP Proxy	24
3.5. ENABLING POWER MANAGEMENT ON MANAGED HOSTS	25
3.6. CONFIGURING DNS, DHCP, AND TFTP ON SATELLITE SERVER	25
3.7. DISABLING DNS, DHCP, AND TFTP FOR UNMANAGED NETWORKS	26
3.8. CONFIGURING SATELLITE SERVER FOR OUTGOING EMAILS	27
3.9. CONFIGURING SATELLITE SERVER WITH A CUSTOM SSL CERTIFICATE	29
3.9.1. Creating a Custom SSL Certificate for Satellite Server	30
3.9.2. Deploying a Custom SSL Certificate to Satellite Server	31
3.9.3. Deploying a Custom SSL Certificate to Hosts	33
3.10. USING EXTERNAL DATABASES WITH SATELLITE	33
3.10.1. MongoDB as an External Database Considerations	33
3.10.2. PostgreSQL as an External Database Considerations	34
3.10.3. Preparing a Host for External Databases	34
3.10.4. Installing MongoDB	35
3.10.5. Installing PostgreSQL	36
3.10.6. Configuring Satellite to use External Databases	37
3.11. RESTRICTING ACCESS TO MONGOD	38
3.12. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES	38

CHAPTER 4. CONFIGURING SATELLITE SERVER WITH EXTERNAL SERVICES	40
4.1. CONFIGURING SATELLITE SERVER WITH EXTERNAL DNS	40
4.2. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP	41
4.2.1. Configuring an External DHCP Server to Use with Satellite Server	41
4.2.2. Configuring Satellite Server with an External DHCP Server	44
4.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP	45
4.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL IDM DNS	46
4.4.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication	46
4.4.2. Configuring Dynamic DNS Update with TSIG Authentication	50
4.4.3. Reverting to Internal DNS Service	52
APPENDIX A. APPLYING CUSTOM CONFIGURATION TO RED HAT SATELLITE	54
APPENDIX B. RESTORING MANUAL CHANGES OVERWRITTEN BY A PUPPET RUN	55

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

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

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base operating system:

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

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

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

- postgres
- mongoddb
- apache
- qpidd
- qdrouterd
- squid
- foreman
- tomcat
- foreman-proxy
- puppet
- puppetserver

**NOTE**

The Red Hat Satellite Server and Capsule Server versions must match. For example, a Satellite 6.7 Server cannot run an earlier or later version of Capsule Server. Mismatching Satellite Server and Capsule Server versions results in the Capsule Server failing silently.

Certified hypervisors

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

FIPS Mode

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

1.2. STORAGE REQUIREMENTS

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

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

Table 1.1. Storage Requirements for a Satellite Server Installation

Directory	Installation Size	Runtime Size
/var/cache/pulp/	1 MB	20 GB
/var/lib/pulp/	1 MB	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/lib/qpidd/	25 MB	Not Applicable
/var/log/	10 MB	10 GB
/var/lib/pgsql/	100 MB	10 GB
/var/spool/squid/	0 MB	10 GB
/usr	3 GB	Not Applicable
/opt	3 GB	Not Applicable
/opt/puppetlabs	500 MB	Not Applicable

1.3. STORAGE GUIDELINES

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

- If you mount the **/tmp** directory as a separate file system, you must use the **exec** mount option in the **/etc/fstab** file. If **/tmp** is already mounted with the **noexec** option, you must change the option to **exec** and re-mount the file system. This is a requirement for the **puppetserver** service to work.
- Because most Satellite Server data is stored in the **/var** directory, mounting **/var** on LVM storage can help the system to scale.
- 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.
- 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/**.
- Use high-bandwidth, low-latency storage for the **/var/lib/pulp/** and **/var/lib/mongodb/** directories. As Red Hat Satellite has many operations that are I/O intensive, using high latency, low-bandwidth storage causes performance degradation. Ensure your installation has a speed in the range 60 – 80 Megabytes per second. You can use the **fiio** tool to get this data. See the Red Hat Knowledgebase solution [Impact of Disk Speed on Satellite Operations](#) for more information on using the **fiio** tool.

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 Server 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**, **noexec**, and **noatime**.
- Do not use NFS for Pulp data storage. Using NFS for Pulp has a negative performance impact on content synchronization.
- Do not use the GFS2 file system as the input-output latency is too high.

Log File Storage

Log files are written to **/var/log/messages/**, **/var/log/httpd/**, and **/var/lib/foreman-proxy/openscap/content/**. You can manage the size of these files using **logrotate**. For more information, see [Log Rotation](#) in the *Red Hat Enterprise Linux 7 System Administrator's Guide* .

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

SELinux Considerations for NFS Mount

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

```
nfs.example.com:/nfsshare /var/lib/pulp/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. After a full synchronization task is completed, the content is moved to the **/var/lib/pulp/** directory.

For content in RPM format, each RPM file is moved to the **/var/lib/pulp** directory after it is synchronized. A maximum of 5 RPM files are stored in the **/var/cache/pulp/** directory at any time. 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/foreman/** 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.

1.4. SUPPORTED OPERATING SYSTEMS

You can install the operating system from a disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Satellite Server is supported only on the latest versions of Red Hat Enterprise Linux 7 Server that is available at the time when Satellite Server 6.7 is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

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

Install Satellite Server on a freshly provisioned system.

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

1.5. SUPPORTED BROWSERS

Satellite supports recent versions of Firefox and Google Chrome browsers.

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

1.6. PORTS AND FIREWALLS REQUIREMENTS

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

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

Integrated Capsule

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

Clients of Capsule

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

Required ports can change based on your configuration.

A matrix table of ports is available in the Red Hat Knowledgebase solution [Red Hat Satellite List of Network Ports](#).

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

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).

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

Any managed host that is directly connected to Satellite Server is a client in this context because it is a client of the integrated Capsule. This includes the base operating 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.
5900 to 5930	TCP	SSL/TLS	Satellite originated communications, for NoVNC console in web UI to hypervisors.

1.7. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER

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

Use this procedure to configure the host-based firewall on the 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 1.6, “Ports and Firewalls Requirements”](#).

Procedure

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

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

2. Make the changes persistent:

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

1.8. VERIFYING FIREWALL SETTINGS

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

Procedure

To verify the firewall settings, complete the following step:

1. Enter the following command:

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

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

1.9. VERIFYING DNS RESOLUTION

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

Procedure

1. Ensure that the host name and local host resolve correctly:

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

Successful name resolution results in output similar to the following:

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

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

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

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

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

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

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

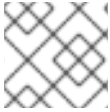


WARNING

Name resolution is critical to the operation of Satellite 6. If Satellite cannot properly resolve its fully qualified domain name, tasks such as content management, subscription management, and provisioning will fail.

CHAPTER 2. INSTALLING SATELLITE SERVER

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

You cannot register Satellite Server to itself.

Use the following procedures to install Satellite Server, perform the initial configuration, and import subscription manifests. For more information on subscription manifests, see [Managing Subscriptions](#) in the *Content Management Guide*.

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/satellite.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. 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 (RHSC), and Red Hat Satellite.

Procedure

- 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
```

2.2. ATTACHING THE SATELLITE INFRASTRUCTURE SUBSCRIPTION

After you have registered Satellite Server, you must identify your subscription Pool ID and attach an available subscription. The Red Hat Satellite Infrastructure subscription provides access to the Red Hat Satellite, Red Hat Enterprise Linux, and Red Hat Software Collections (RHSC) content. This is the only subscription required.

Red Hat Satellite Infrastructure is included with all subscriptions that include Smart Management. For more information, see the Red Hat Knowledgebase solution [Satellite Infrastructure Subscriptions MCT3718 MCT3719](#).

Subscriptions are classified as available if they are not already attached to a system. 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 see if your subscription is being consumed by another system.

Procedure

1. Identify the Pool ID of the Satellite Infrastructure subscription:

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

The command displays output similar to the following:

```
Subscription Name: Red Hat Satellite Infrastructure Subscription
Provides:          Red Hat Satellite
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat CodeReady Linux Builder for x86_64
                  Red Hat Ansible Engine
                  Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                  Red Hat
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Enterprise Linux Server
                  Red Hat Satellite Capsule
                  Red Hat Enterprise Linux for x86_64
                  Red Hat Enterprise Linux High Availability for x86_64
                  Red Hat Satellite
                  Red Hat Satellite 5 Managed DB
                  Red Hat Satellite 6
                  Red Hat Discovery
SKU:               MCT3719
Contract:          11878983
Pool ID:           8a85f99968b92c3701694ee998cf03b8
Provides Management: No
Available:         1
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Subscription Type: Standard
Ends:              03/04/2020
System Type:       Physical
```

2. Make a note of the subscription Pool ID. Your subscription Pool ID is different from the example provided.

3. Attach the Satellite Infrastructure subscription to the base operating system that your Satellite Server is running on:

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

The command displays output similar to the following:

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

4. Optional: Verify that the Satellite Infrastructure subscription is attached:

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

2.3. CONFIGURING REPOSITORIES

Use this procedure to enable the repositories that are required to install Satellite Server.

Procedure

1. Disable all repositories:

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

2. Enable the following repositories:

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-satellite-6.7-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-server-rhsc-7-rpms \
--enable=rhel-7-server-ansible-2.8-rpms
```



NOTE

If you are installing Satellite Server as a virtual machine hosted on Red Hat Virtualization, you must also enable the **Red Hat Common** repository, and install Red Hat Virtualization 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 any metadata:

```
# yum clean all
```

4. Optional: Verify that the required repositories are enabled:

```
# yum repolist enabled
```

2.4. INSTALLING THE SATELLITE SERVER PACKAGES

You must update all packages before installing the Satellite Server packages.

Procedure

1. Update all packages:

```
# yum update
```

2. Install the Satellite Server packages:

```
# yum install satellite
```

2.5. SYNCHRONIZING THE SYSTEM CLOCK WITH CHRONYD

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

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

Procedure

1. Install the **chrony** package:

```
# yum install chrony
```

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

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

2.6. INSTALLING THE SOS PACKAGE ON THE BASE OPERATING SYSTEM

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

Procedure

1. Install the **sos** package:

```
# yum install sos
```

2.7. CONFIGURING SATELLITE SERVER

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

- [Section 2.7.1, "Configuring Satellite Manually"](#). 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.

- [Section 2.7.2, “Configuring Satellite Automatically using an Answer File”](#) . This method is performed by using an answer file to automate the configuration process when running the installation 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.



NOTE

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

2.7.1. Configuring Satellite 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.

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

Considerations for Manual Configuration

- Use the **satellite-installer --scenario satellite --help** command to display the available options and any default values. If you do not specify any values, the default values are used.
- Specify a meaningful value for the option: **--foreman-initial-organization**. This can be your company name. An internal label that matches the value is also created and cannot be changed afterwards. If you do not specify a value, an organization called **Default Organization** with the label **Default_Organization** is created. You can rename the organization name but not the label.
- By default, all configuration files configured by the installer are managed by Puppet. When **satellite-installer** runs, it overwrites any manual changes to the Puppet managed files with the initial values. By default, Satellite Server is installed with the Puppet agent running as a service. If required, you can disable Puppet agent on Satellite Server using the **--puppet-runmode=none** option.
- If you want to manage DNS files and DHCP files manually, use the **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false** options so that Puppet does not manage the files related to the respective services. For more information on how to apply custom configuration on other services, see [Appendix A, Applying Custom Configuration to Red Hat Satellite](#).

Procedure

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

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

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

2.7.2. Configuring Satellite Automatically using an Answer File

You can use answer files to automate installations with customized options. The initial answer file is sparsely populated. After you run the **satellite-installer** script the first time, the answer file is populated with the standard parameter values for installation. You can change the configuration of Satellite Server at any time.

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

Procedure

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
```

2.8. IMPORTING A SUBSCRIPTION MANIFEST INTO SATELLITE SERVER

Use the following procedure to import a Subscription Manifest into Satellite Server.

Prerequisites

- You must have a Subscription Manifest file exported from the Customer Portal. For more information, see [Using Manifests](#) in the *Using Red Hat Subscription Management* guide.

Procedure

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

2. Navigate to **Content** > **Subscriptions** and click **Manage Manifest**.
3. In the Manage Manifest window, click **Browse**.
4. Navigate to the location that contains the Subscription Manifest file, then click **Open**. If the Manage Manifest window does not close automatically, click **Close** to return to the Subscriptions window.

For CLI Users

1. Copy the Subscription Manifest file from your client to Satellite Server:

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

2. Log in to Satellite Server as the **root** user and import the Subscription Manifest file:

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

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER

3.1. USING RED HAT INSIGHTS WITH SATELLITE SERVER

You can use Red Hat Insights to diagnose systems and downtime related to security exploits, performance degradation and stability failures. You can use the dashboard to quickly identify key risks to stability, security, and performance. You can sort by category, view details of the impact and resolution, and then determine what systems are affected.

Note that you do not require a Red Hat Insights entitlement in your subscription manifest. For more information about Satellite and Red Hat Insights, see [Red Hat Insights on Satellite Red Hat Enterprise Linux \(RHEL\)](#).

To maintain your Satellite Server, and improve your ability to monitor and diagnose problems you might have with Satellite, install Red Hat Insights on Satellite Server and register Satellite Server with Red Hat Insights.

Scheduling insights-client

Note that you can change the default schedule for running **insights-client** by configuring **insights-client.timer** on Satellite. For more information, see [Changing the insights-client schedule](#) in the *Client Configuration Guide for Red Hat Insights*.

Procedure

1. To install Red Hat Insights on Satellite Server, enter the following command:

```
# satellite-maintain packages install insights-client
```

2. To register Satellite Server with Red Hat Insights, enter the following command:

```
# insights-client --register
```

3.2. ENABLING THE SATELLITE TOOLS 6.7 REPOSITORY

The Satellite Tools 6.7 repository provides the **katello-agent**, **katello-host-tools**, and **puppet** packages for clients registered to Satellite Server.

Procedure

1. In the Satellite web UI, navigate to **Content > Red Hat Repositories**.
2. Use the Search field to enter the following repository name: **Satellite Tools 6.7 (for RHEL 7 Server) (RPMs)**.
3. In the Available Repositories pane, click on **Satellite Tools 6.7 (for RHEL 7 Server) (RPMs)** to expand the repository set.

If the **Satellite Tools 6.7** 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 6.7 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 6.7 repository using the **hammer repository-set enable** command:

```
# hammer repository-set enable --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--basearch='x86_64' \
--name 'Red Hat Satellite Tools 6.7 (for RHEL 7 Server) (RPMs)'
```

3.3. SYNCHRONIZING THE SATELLITE TOOLS 6.7 REPOSITORY

Use this section to synchronize the Satellite Tools 6.7 repository from the Red Hat Content Delivery Network (CDN) to your Satellite. This repository provides the **katello-agent**, **katello-host-tools**, and **puppet** packages for clients registered to Satellite Server.

Procedure

1. In the Satellite web UI, navigate to **Content > Sync Status**.
A list of product repositories available for synchronization is displayed.
2. Click the arrow next to the **Red Hat Enterprise Linux Server** product to view available content.
3. Select **Satellite Tools 6.7 (for RHEL 7 Server) RPMs x86_64**
4. Click **Synchronize Now**.

For CLI Users

- Synchronize your Satellite Tools 6.7 repository using the **hammer repository synchronize** command:

```
# hammer repository synchronize --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--name 'Red Hat Satellite Tools 6.7 for RHEL 7 Server RPMs x86_64' \
--async
```

3.4. CONFIGURING SATELLITE SERVER WITH AN HTTP PROXY

Use the following procedures to configure Satellite with an HTTP proxy.

3.4.1. Adding a Default HTTP Proxy to Satellite

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.

Procedure

1. In the Satellite web UI, navigate to **Infrastructure > HTTP Proxies**.
2. Click **New HTTP Proxy**.
3. In the **Name** field, enter the name for the HTTP proxy.
4. In the **Url** field, enter the URL of the HTTP proxy in the following format:
https://*proxy.example.com*:8080.
5. Optional: If authentication is required, in the **Username** field, enter the username to authenticate with.
6. Optional: If authentication is required, in the **Password** field, enter the password to authenticate with.
7. To test connection to the proxy, click the **Test Connection** button.
8. Click **Submit**.
9. Navigate to **Administer > Settings**, and click the **Content** tab.
10. Set the **Default HTTP Proxy** setting to the created HTTP proxy.

For CLI Users

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. Add an HTTP proxy entry to Satellite:

```
# hammer http-proxy create --name=myproxy \  
--url http://myproxy.example.com:8080 \  
--username=proxy_username \  
--password=proxy_password
```

3. Configure Satellite to use this HTTP proxy by default:

```
# hammer settings set --name=content_default_http_proxy --value=myproxy
```

3.4.2. Configuring the HTTP Proxy to Connect to Red Hat CDN

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

Procedure

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

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

Satellite Server uses SSL to communicate with the Red Hat CDN securely. 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.

2. 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 =
```

3.4.3. Configuring SELinux to Ensure Access to Satellite on 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.

Procedure

1. On Satellite, 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
```

3.4.4. 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.

Procedure

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.

For CLI Users

- Enter the following command:

```
# hammer settings set --name=http_proxy --value=Proxy_URL
```

3.4.5. 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.

Procedure

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.

For CLI Users

- Enter the following command:

```
# hammer settings set --name=http_proxy_except_list --value=[hostname1.hostname2...]
```

3.4.6. Resetting the HTTP Proxy

If you want to reset the current HTTP proxy setting, unset the **Default HTTP Proxy** setting.

Procedure

1. Navigate to **Administer** > **Settings**, and click the **Content** tab.

2. Set the **Default HTTP Proxy** setting to **no global default**.

For CLI Users

- Set the **content_default_http_proxy** setting to an empty string:

```
# hammer settings set --name=content_default_http_proxy --value=""
```

3.5. ENABLING POWER MANAGEMENT ON MANAGED HOSTS

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

Prerequisites

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

Procedure

- To enable BMC, enter the following command:

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

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

To configure the DNS, DHCP, and TFTP services on Satellite Server, use the **satellite-installer** command with the options appropriate for your environment. To view a complete list of configurable options, enter the **satellite-installer --scenario satellite --help** command.

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

To use external DNS, DHCP, and TFTP services instead, see [Chapter 4, Configuring Satellite Server with External Services](#).

Adding Multihomed DHCP details

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

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

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

If this file does not exist already, create it.

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

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

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

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

Prerequisites

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

Procedure

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

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

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

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

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

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

Procedure

1. On Satellite Server, enter the following command:

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

2. In the Satellite web UI, navigate to **Infrastructure** > **Subnets** and select a subnet.
3. Click the **Capsules** tab and clear the **DHCP Capsule**, **TFTP Capsule**, and **Reverse DNS Capsule** fields.
4. 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.

3.8. 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
```

Procedure

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

Table 3.1. Using an SMTP server as a delivery method

Name	Example value
Delivery method	SMTP
SMTP address	<i>smtp.example.com</i>
SMTP authentication	login
SMTP HELO/EHLO domain	<i>example.com</i>
SMTP password	<i>password</i>
SMTP port	25
SMTP username	<i>user@example.com</i>

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

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

Table 3.2. Using gmail.com as an SMTP server

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

- 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*.

3.9. CONFIGURING SATELLITE SERVER WITH A CUSTOM SSL CERTIFICATE

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

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

1. [Section 3.9.1, "Creating a Custom SSL Certificate for Satellite Server"](#)
2. [Section 3.9.2, "Deploying a Custom SSL Certificate to Satellite Server"](#)
3. [Section 3.9.3, "Deploying a Custom SSL Certificate to Hosts"](#)

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

3.9.1. Creating a Custom SSL Certificate for Satellite Server

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

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

- You must use the Privacy-Enhanced Mail (PEM) encoding for the SSL certificates.
- You cannot use the same certificate for both Satellite Server and Capsule Server.
- The same Certificate Authority must sign certificates for Satellite Server and Capsule Server.

Procedure

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

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

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

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

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

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

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

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

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

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

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

[ alt_names ]
DNS.1 = satellite.example.com 3
```

- 1 In the `[req_distinguished_name]` section, enter information about your organization.
- 2 Set the certificate's Common Name **CN** to match the fully qualified domain name (FQDN) of your Satellite Server. To confirm a FQDN, on that Satellite Server, enter the **hostname -f** command. This is required to ensure that the **katello-certs-check** command validates the certificate correctly.
- 3 Set the Subject Alternative Name (SAN) **DNS.1** to match the fully qualified domain name (FQDN) of your server.

4. Generate the Certificate Signing Request (CSR):

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

- 1 Path to the private key.
 - 2 Path to the configuration file.
 - 3 Path to the CSR to generate.
5. Send the certificate signing request to the Certificate Authority. The same Certificate Authority must sign certificates for Satellite Server and Capsule Server. When you submit the request, specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the Certificate Authority for the preferred method. In response to the request, you can expect to receive a Certificate Authority bundle and a signed certificate, in separate files.

3.9.2. Deploying a Custom SSL Certificate to Satellite Server

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

Procedure

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

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

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

- 1 Path to the Satellite Server certificate file that is signed by a Certificate Authority.
- 2 Path to the private key that was used to sign the Capsule Server certificate.
- 3 Path to the Certificate Authority bundle.

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

Example output of **katello-certs-check**

Validation succeeded.

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

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

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

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

2. From the output of the **katello-certs-check** command, depending on your requirements, enter the **satellite-installer** command that installs a new Satellite with custom SSL certificates or updates certificates on a currently running Satellite.
If you are unsure which command to run, you can verify that Satellite is installed by checking if the file **/etc/foreman-installer/scenarios.d/installed** exists. If the file exists, run the second **satellite-installer** command that updates certificates.



IMPORTANT

Do not delete the certificate archive file after you deploy the certificate. It is required, for example, when upgrading Satellite Server.

3. On a computer with network access to Satellite Server, navigate to the following URL:
<https://satellite.example.com>.
4. In your browser, view the certificate details to verify the deployed certificate.

3.9.3. Deploying a Custom SSL Certificate to Hosts

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

Procedure

- On each host, install the **katello-ca-consumer** package:

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

3.10. 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 instead of the default local databases can help with the server load. 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. You must have your own database administrator to support and maintain external databases.

Use [MongoDB as an External Database Considerations](#) and [PostgreSQL as an External Database Considerations](#) to decide if you want to use external databases for your Satellite deployment.

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

1. [Section 3.10.3, “Preparing a Host for External Databases”](#). Prepare a Red Hat Enterprise Linux 7 server to host the external databases.
2. [Section 3.10.4, “Installing MongoDB”](#). Prepare MongoDB with user **pulp** owning the **pulp_database**
3. [Section 3.10.5, “Installing PostgreSQL”](#). Prepare PostgreSQL with databases for Satellite and Candlepin and dedicated users owning them.
4. [Section 3.10.6, “Configuring Satellite to use External Databases”](#). Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**.

3.10.1. 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 decide if this option is right for your Satellite configuration. Satellite supports MongoDB version 3.4.

Advantages of External MongoDB

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

Disadvantages of External MongoDB

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

FIPS-related Restrictions

- You cannot use an external MongoDB with Satellite in FIPS mode.

3.10.2. 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 decide if this option is right for your Satellite configuration. Satellite supports PostgreSQL version 9.2.

Advantages of External PostgreSQL:

- Increase in free memory and free CPU on Satellite
- Flexibility to set **shared_buffers** on the PostgreSQL database to a high number without the risk of interfering with other services on Satellite
- Flexibility to tune the PostgreSQL server's system without adversely affecting Satellite operations

Disadvantages of External PostgreSQL

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

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

3.10.3. Preparing a Host for External Databases

Install a freshly provisioned system with the latest Red Hat Enterprise Linux 7 server to host the external databases.

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

Prerequisites

- The Red Hat Enterprise Linux 7 server must meet Satellite's [Storage Requirements](#).

Procedure

1. Use the instructions in [Attaching the Satellite Infrastructure Subscription](#) to attach a Satellite subscription to your server.
2. Disable all repositories and enable only the following repositories:

```
# subscription-manager repos --disable '*'
# subscription-manager repos --enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-rpms
```

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

Procedure

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"
```

3.10.5. Installing PostgreSQL

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

Procedure

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_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
```

- Switch to the **postgres** user and start the PostgreSQL client:

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

- 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;
```

- Exit the **postgres** user:

```
# \q
```

- 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"
```

3.10.6. Configuring Satellite to use External Databases

Use the **satellite-installer** command to configure Satellite to connect to external MongoDB and PostgreSQL databases.

Prerequisites

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

Procedure

- To configure the external databases 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
```

2. Verify the status of the databases:

- For PostgreSQL, enter the following command:

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

- For MongoDB, enter the following command:

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

3.11. RESTRICTING ACCESS TO MONGOD

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

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

Procedure

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

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

2. Make the changes persistent:

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

3.12. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES

If your Satellite deployment includes more than 5000 hosts, you can use predefined tuning profiles to improve the performance of Satellite Server.

You can choose one of the profiles depending on the number of hosts your Satellite manages and available hardware resources. The tuning profiles are available in the `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes` directory.

default

Number of managed hosts: 0-5000

RAM: 20G

Number of CPU cores: 4

medium

Number of managed hosts: 5001-10000

RAM: 32G

Number of CPU cores: 8

large

Number of managed hosts: 10001-20000

RAM: 64G

Number of CPU cores: 16

extra-large

Number of managed hosts: 20001-60000

RAM: 128G

Number of CPU cores: 32

extra-extra-large

Number of managed hosts: 60000+

RAM: 256G

Number of CPU cores: 48+

Procedure

- To configure a tuning profile for your Satellite deployment, enter the **satellite-installer** command with the **--tuning** option. For example, to apply the medium tuning profile settings, enter the following command:

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

CHAPTER 4. CONFIGURING SATELLITE SERVER WITH EXTERNAL SERVICES

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

4.1. CONFIGURING SATELLITE SERVER WITH EXTERNAL DNS

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

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

Prerequisites

- You must have a configured external DNS server.

Procedure

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

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

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

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

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

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

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

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

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

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

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

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

- Restart the foreman-proxy service:

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

- Log in to the Satellite Server web UI.
- Navigate to **Infrastructure > Capsules**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.
- Associate the DNS service with the appropriate subnets and domain.

4.2. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP

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

- [Section 4.2.1, "Configuring an External DHCP Server to Use with Satellite Server"](#)
- [Section 4.2.2, "Configuring Satellite Server with an External DHCP Server"](#)

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

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



NOTE

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

Procedure

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

```
# yum install dhcp bind
```

- Generate a security token:

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

-
- As a result, a key pair that consists of two files is created in the current directory.
3. Copy the secret hash from the key:

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

4. Edit the **dhcpd** configuration file for all of the subnets and add the key. The following is an example:

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

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

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

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

5. Delete the two key files from the directory that they were created in.
6. On Satellite Server, define each subnet. Do not set DHCP Capsule for the defined Subnet yet. To prevent conflicts, set up the lease and reservation ranges separately. For example, if the lease range is 192.168.38.10 to 192.168.38.100, in the Satellite web UI define the reservation range as 192.168.38.101 to 192.168.38.250.
7. Configure the firewall for external access to the DHCP server:

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

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

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

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

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

10. To ensure that the configuration files are accessible, restore the read and execute flags:

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

11. Start the DHCP service:

```
# systemctl start dhcpd
```

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

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

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

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

14. To create mount points for the created directories, add the following line to the **/etc/fstab** file:

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

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

```
# mount -a
```

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

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)
/exports/etc/dhcp 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
/exports/var/lib/dhcpd 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

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

17. Reload the NFS server:

```
# exportfs -rva
```

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

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

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

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

4.2.2. Configuring Satellite Server with an External DHCP Server

You can configure Satellite Server with an external DHCP server.

Prerequisite

- Ensure that you have configured an external DHCP server and that you have shared the DHCP configuration and lease files with Satellite Server. For more information, see [Section 4.2.1, "Configuring an External DHCP Server to Use with Satellite Server"](#).

Procedure

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

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

2. Create the DHCP directories for NFS:

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

3. Change the file owner:

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

4. Verify communication with the NFS server and the Remote Procedure Call (RPC) communication paths:

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

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

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

DHCP_Server_FQDN:/exports/var/lib/dhcpd /mnt/nfs/var/lib/dhcpd nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcpd_state_t:s0" 0 0
```

6. Mount the file systems on **/etc/fstab**:

```
# mount -a
```

7. To verify that the **foreman-proxy** user can access the files that are shared over the network, display the DHCP configuration and lease files:


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

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

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

- Restart the **foreman-proxy** service:

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

- Log in to the Satellite Server web UI.
- Navigate to **Infrastructure > Capsules**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.
- Associate the DHCP service with the appropriate subnets and domain.

4.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

You can configure Satellite Server with external TFTP services.

Procedure

- Create the TFTP directory for NFS:

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

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

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

- Mount the file systems in **/etc/fstab**:

```
# mount -a
```

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

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

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

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

- Log in to the Satellite Server web UI.
- Navigate to **Infrastructure > Capsules**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.
- Associate the TFTP service with the appropriate subnets and domain.

4.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL IDM DNS

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

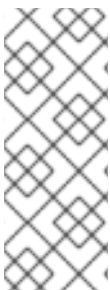
Satellite Server can be configured to use a Red Hat Identity Management (IdM) server to provide DNS service. For more information about Red Hat Identity Management, see the [Linux Domain Identity, Authentication, and Policy Guide](#).

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

- [Section 4.4.1, "Configuring Dynamic DNS Update with GSS-TSIG Authentication"](#)
- [Section 4.4.2, "Configuring Dynamic DNS Update with TSIG Authentication"](#)

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

- [Section 4.4.3, "Reverting to Internal DNS Service"](#)



NOTE

You are not required to use Satellite Server to manage DNS. When you are using the realm enrollment feature of Satellite, where provisioned hosts are enrolled automatically to IdM, the **ipa-client-install** script creates DNS records for the client. Configuring Satellite Server with external IdM DNS and realm enrollment are mutually exclusive. For more information about configuring realm enrollment, see [External Authentication for Provisioned Hosts](#) in *Administering Red Hat Satellite*.

4.4.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication

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

Prerequisites

Prerequisites

- You must ensure the IdM server is deployed and the host-based firewall is configured correctly. For more information, see [Port Requirements](#) in the *Linux Domain Identity, Authentication, and Policy Guide*.
- You must contact the IdM server administrator to ensure that you obtain an account on the IdM server with permissions to create zones on the IdM server.
- You must confirm whether Satellite Server or Capsule Server is configured to provide DNS service for your deployment.
- You must configure DNS, DHCP and TFTP services on the base operating system of either the Satellite or Capsule that is managing the DNS service for your deployment.
- You must create a backup of the answer file. You can use the backup to restore the answer file to its original state if it becomes corrupted. For more information, see [Configuring Satellite Server](#).

Procedure

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

Creating a Kerberos Principal on the IdM Server

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

```
# kinit idm_user
```

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

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

Installing and Configuring the IdM Client

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

- On Satellite Server, enter the following command:

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

- On Capsule Server, enter the following command:

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

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

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

3. Obtain a Kerberos ticket:

```
# kinit admin
```

4. Remove any preexisting **keytab**:

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

- Obtain the **keytab** for this system:

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



NOTE

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

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

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

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

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

Configuring DNS Zones in the IdM web UI

- Create and configure the zone that you want to manage:
 - Navigate to **Network Services > DNS > DNS Zones**.
 - Select **Add** and enter the zone name. For example, **example.com**.
 - Click **Add and Edit**
 - Click the Settings tab and in the **BIND update policy** box, add the following to the semi-colon separated list:

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

- Set **Dynamic update** to **True**.
 - Enable **Allow PTR sync**.
 - Click **Save** to save the changes.
- Create and configure the reverse zone:
 - Navigate to **Network Services > DNS > DNS Zones**.
 - Click **Add**.
 - Select **Reverse zone IP network** and add the network address in CIDR format to enable reverse lookups.
 - Click **Add and Edit**

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

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

- f. Set **Dynamic update** to **True**.
- g. Click **Save** to save the changes.

Configuring the Satellite or Capsule Server that Manages the DNS Service for the Domain

1. Use the **satellite-installer** command to configure the Satellite or Capsule that manages the DNS Service for the domain:

- On Satellite, enter the following command:

```
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 Capsule, enter the following command:

```
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
```

2. Restart the Satellite or Capsule's Proxy Service.

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

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

Updating the Configuration in the Satellite web UI

1. Navigate to **Infrastructure > Capsules**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.
2. Configure the domain:
 - a. Navigate to **Infrastructure > Domains** and select the domain name.

- b. In the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.
3. Configure the subnet:
 - a. Navigate to **Infrastructure** > **Subnets** and select the subnet name.
 - b. In the **Subnet** tab, set **IPAM** to **None**.
 - c. In the **Domains** tab, select the domain that you want to manage using the IdM server.
 - d. In the **Capsules** tab, ensure **Reverse DNS Capsule** is set to the Capsule where the subnet is connected.
 - e. Click **Submit** to save the changes.

4.4.2. Configuring Dynamic DNS Update with TSIG Authentication

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

Prerequisites

- You must ensure the IdM server is deployed and the host-based firewall is configured correctly. For more information, see [Port Requirements](#) in the *Linux Domain Identity, Authentication, and Policy Guide*.
- You must obtain **root** user access on the IdM server.
- You must confirm whether Satellite Server or Capsule Server is configured to provide DNS service for your deployment.
- You must configure DNS, DHCP and TFTP services on the base operating system of either the Satellite or Capsule that is managing the DNS service for your deployment.
- You must create a backup of the answer file. You can use the backup to restore the answer file to its original state if it becomes corrupted. For more information, see [Configuring Satellite Server](#).

Procedure

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

Enabling External Updates to the DNS Zone in the IdM Server

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

```
include "/etc/rndc.key"; controls { inet IdM_Server_IP_Address port 953 allow {  
Satellite_IP_Address; } keys { "rndc-key"; }; };
```

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

```
# systemctl reload named
```

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

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


```
grant "rndc-key" zonesub ANY;
```
 - b. Set **Dynamic update** to **True**.
 - c. Click **Update** to save the changes.
4. Copy the **/etc/rndc.key** file from the IdM server to the base operating system of your Satellite Server. Enter the following command:

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

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

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

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

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

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

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

Testing External Updates to the DNS Zone in the IdM Server

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

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

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

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

- 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
```

- 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
```

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

- Confirm that the DNS entry was removed:

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

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

4.4.3. Reverting to Internal DNS Service

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

Procedure

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

Configuring Satellite or Capsule as a DNS Server

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

```
# satellite-installer
```

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


```
# 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
```

For more information, see [Configuring DNS, DHCP, and TFTP on Capsule Server](#).

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

Updating the Configuration in the Satellite web UI

1. Navigate to **Infrastructure** > **Capsules**.
2. For each Capsule that you want to update, from the **Actions** list, select **Refresh**.
3. Configure the domain:
 - a. Navigate to **Infrastructure** > **Domains** and click the domain name that you want to configure.
 - b. In the **Domain** tab, set **DNS Capsule** to the Capsule where the subnet is connected.
4. Configure the subnet:
 - a. Navigate to **Infrastructure** > **Subnets** and select the subnet name.
 - b. In the **Subnet** tab, set **IPAM** to **DHCP** or **Internal DB**.
 - c. In the **Domains** tab, select the domain that you want to manage using Satellite or Capsule.
 - d. In the **Capsules** tab, set **Reverse DNS Capsule** to the Capsule where the subnet is connected.
 - e. Click **Submit** to save the changes.

APPENDIX A. APPLYING CUSTOM CONFIGURATION TO RED HAT SATELLITE

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

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

Common examples include:

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

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

- To turn off the Apache server signature entirely:

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

- 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.

APPENDIX B. RESTORING MANUAL CHANGES OVERWRITTEN BY A PUPPET RUN

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

Procedure

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

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

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

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

3. Restore the overwritten file:

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

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