Red Hat Satellite 6.6

Installing Capsule Server

Installing Red Hat Satellite Capsule Server

Last Updated: 2020-05-19

Red Hat Satellite 6.6 Installing Capsule Server

Installing Red Hat Satellite Capsule Server

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

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base system:

- x86_64 architecture
- The latest version of Red Hat Enterprise Linux 7 Server
- 4-core 2.0 GHz CPU at a minimum
- A minimum of 12 GB RAM is required for Capsule Server to function. In addition, a minimum of 4 GB RAM of swap space is also recommended. Capsule 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 Capsule Server, ensure that your environment meets the requirements for installation.

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

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

NOTE

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

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

Certified hypervisors

Capsule 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 Capsule 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 Capsule Server Installation

Directory	Installation Size	Runtime Size
/var/cache/pulp/	1 MB	20 GB (Minimum)
/var/lib/pulp/	1 MB	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/opt	500 MB	Not Applicable

1.3. STORAGE GUIDELINES

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

- Because most Capsule Server data is stored within the **/var** directory, mounting **/var** on LVM storage can help the system to scale.
- For the **/var/lib/pulp/** and **/var/lib/mongodb/** directories, use high-bandwidth, low-latency storage, and solid state drives (SSD) rather than hard disk drives (HDD). As Red Hat Satellite has many operations that are I/O intensive, using high latency, low-bandwidth storage causes performance degradation. Ensure your installation has a speed in the range 60 - 80 Megabytes per second. You can use the **fio** tool to get this data. See the Red Hat Knowledgebase solution Impact of Disk Speed on Satellite 6 Operations for more information on using the **fio** tool.

- The **/var/lib/qpidd/** directory uses slightly more than 2 MB per Content Host managed by the **goferd** service. For example, 10 000 Content Hosts require 20 GB of disk space in **/var/lib/qpidd/**.
- Using the same volume for the **/var/cache/pulp/** and **/var/lib/pulp/** directories can decrease the time required to move content from **/var/cache/pulp/** to **/var/lib/pulp/** after synchronizing.

File System Guidelines

- Use the XFS file system for Red Hat Satellite 6 because it does not have the inode limitations that **ext4** does. As Capsule Server uses a lot of symbolic links it is likely that your system may 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 volumes with the following option in the **/etc/fstab** file: **bg**, **nolock**, and **noatime**.
- Do not use NFS for Pulp data storage. Using NFS for Pulp has a negative performance impact on content synchronization.
- Do not use the GFS2 file system as the input-output latency is too high.

SELinux Considerations for NFS Mount

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

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

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

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

Duplicated Packages

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

Temporary Storage

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

ISO Images

For content in ISO format, all ISO files per synchronization task are stored in **/var/cache/pulp/** until the task is complete, after which they are moved to the **/var/lib/pulp/** directory.

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

For example, if you are synchronizing four ISO files, each 4 GB in size, this requires a total of 16 GB in the **/var/cache/pulp/** directory. Consider the number of ISO files you intend synchronizing because the temporary disk space required for them typically exceeds that of RPM content.

Software Collections

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

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

Symbolic links

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

Log Storage

You can view log files at the following locations: **/var/log/messages/**, **/var/log/httpd/**, and **/var/lib/foreman-proxy/openscap/content/**. To manage the size of the log files use the **logrotate** configuration file. For more information, see Log Rotation in the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

1.4. SUPPORTED OPERATING SYSTEMS

You can install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Capsule Server is supported only on the latest versions of Red Hat Enterprise Linux 7 Server that is available at the time when Capsule Server 6.6 is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

Red Hat Capsule 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 Capsule Server first, then create a backup of the system before adding any non-Red Hat software.

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

1.5. PORTS AND FIREWALLS REQUIREMENTS

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

The installation of a Capsule Server fails if the ports between Satellite Server and Capsule Server are not open before installation starts.

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

Integrated Capsule

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

Clients of Capsule

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

Required ports can change based on your configuration.

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

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

Table 1.2. Ports for Capsule to Satellite Communication

Port	Protocol	Service	Required For
5646	TCP	AMQP	Capsule's Qpid dispatch router to Qpid dispatch router in Satellite

Table 1.3. Ports for Client to Capsule Communication

Port	Protocol	Service	Required for
80	TCP	HTTP	Anaconda, yum, and for obtaining Katello certificate updates
443	TCP	HTTPS	Anaconda, yum, Telemetry Services, and Puppet
5647	TCP	AMQP	Katello agent to communicate with Capsule's Qpid dispatch router
8000	TCP	HTTPS	Anaconda to download kickstart templates to hosts, and for downloading iPXE firmware
8140	TCP	HTTPS	Puppet agent to Puppet master connections
8443	TCP	HTTPS	Subscription Management Services and Telemetry Services
9090	TCP	HTTPS	Sending SCAP reports to the Smart Proxy in the Capsule and for the discovery image during provisioning

Port	Protocol	Service	Required for
53	TCP and UDP	DNS	Client DNS queries to a Capsule's DNS service (Optional)
67	UDP	DHCP	Client to Capsule broadcasts, DHCP broadcasts for Client provisioning from a Capsule (Optional)
69	UDP	TFTP	Clients downloading PXE boot image files from a Capsule for provisioning (Optional)
5000	TCP	HTTPS	Connection to Katello for the Docker registry (Optional)

Table 1.4. Ports for Capsule to Client Communication

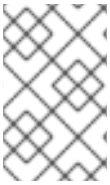
Port	Protocol	Service	Required For
7	TCP and UDP	ICMP	DHCP Capsule to Client network, ICMP ECHO to verify IP address is free (Optional)
68	UDP	DHCP	Capsule to Client broadcasts, DHCP broadcasts for Client provisioning from a Capsule (Optional)
8443	TCP	HTTP	Capsule to Client "reboot" command to a discovered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

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

Table 1.5. Optional Network Ports

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

Port	Protocol	Service	Required For
7911	TCP	DHCP	• Capsule originated commands for orchestration of DHCP records (local or external). • If DHCP is provided by an external service, you must open the port on the external server.



NOTE

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

1.6. ENABLING CONNECTIONS FROM CAPSULE SERVER TO SATELLITE SERVER

On Satellite Server, you must enable the incoming connection from Capsule Server to Satellite Server and make this rule persistent across reboots.

Prerequisites

- Ensure that the firewall rules on Satellite Server are configured to enable connections for client to Satellite communication, because Capsule Server is a client of Satellite Server. For more information, see Enabling Connections from a Client to Satellite Server in *Installing Satellite Server from a Connected Network*.

Procedure

1. On Satellite Server, enter the following command to open the port for Capsule to Satellite communication:

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

2. Make the changes persistent:

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

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

On the base operating system on which you want to install Capsule, you must enable incoming connections from Satellite Server and clients to Capsule Server and make these rules persistent across reboots.

Procedure

1. On the base operating system on which you want to install Capsule, enter the following command to open the ports for Satellite Server and clients communication to Capsule Server:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

2. 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*.

CHAPTER 2. INSTALLING CAPSULE SERVER

Before you install Capsule Server, you must ensure that your environment meets the requirements for installation. For more information, see Section 1.1, “System Requirements”.

2.1. REGISTERING TO SATELLITE SERVER

Use this procedure to register the base system on which you want to install Capsule Server to Satellite Server.

Prerequisites

Before registering it to Satellite Server, ensure that the base system on which you want to install Capsule meets the following conditions:

Subscription Manifest Prerequisites

- On Satellite Server, a manifest must be installed and it must contain the appropriate repositories for the organization you want Capsule to belong to.
- The manifest must contain repositories for the base system on which you want to install Capsule, as well as any clients that you want to connect to Capsule.
- The repositories must be synchronized.

For more information on manifests and repositories, see Managing Subscriptions in the *Red Hat Satellite Content Management Guide*.

Proxy and Network Prerequisites

- The Satellite Server base system must be able to resolve the host name of the Capsule base system and vice versa.
- The base system on which you want to install Capsule Server must not be configured to use a proxy to connect to the Red Hat CDN.
- You must configure the host and network-based firewalls accordingly. For more information, see Section 1.5, “Ports and Firewalls Requirements”.
- You must have a Satellite Server user name and password. For more information, see Configuring External Authentication in *Administering Red Hat Satellite*.

Procedure

To register your system to Satellite Server, complete the following steps:

1. Download the **katello-ca-consumer-latest.noarch.rpm** package on the base system on which you want to install Capsule. The consumer RPM configures the host to download content from the content source that is specified in Red Hat Satellite.

```
# curl --insecure --output katello-ca-consumer-latest.noarch.rpm
https://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2. Install the **katello-ca-consumer-latest.noarch.rpm** package:

```
# yum localinstall katello-ca-consumer-latest.noarch.rpm
```

3. Register the Capsule base system with the environments that you want Capsule to belong to. Use an activation key to simplify specifying the environments.

```
# subscription-manager register --org=organization_name --
activationkey=example_activation_key
```

2.2. ATTACHING THE SATELLITE INFRASTRUCTURE SUBSCRIPTION

After you have registered Capsule 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 (RHSCL) 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

To attach the Satellite Infrastructure subscription, complete the following steps:

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 system that your Capsule 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 Capsule Server.

Procedure

To configure the required repositories, complete the following steps:

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-capsule-6.6-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-tools-6.6-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-ansible-2.8-rpms
```

NOTE

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

3. Clear any **yum** metadata:

```
# yum clean all
```

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

```
# yum repolist enabled
```

2.4. SYNCHRONIZING THE SYSTEM CLOCK WITH CHRONYD

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

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

Procedure

To synchronize the system clock, complete the following steps:

1. Install the **chrony** package:

```
# yum install chrony
```

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

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

2.5. INSTALLING CAPSULE SERVER PACKAGES

Before installing the Capsule Server packages, you must update all packages that are installed on the base system.

Procedure

To install Capsule Server, complete the following steps:

1. Update all packages:

```
# yum update
```

2. Install the **satellite-capsule** package:

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

2.6. CONFIGURING CAPSULE SERVER WITH SSL CERTIFICATES

Red Hat Satellite uses SSL certificates to enable encrypted communications between Satellite Server, external Capsule Servers, and all hosts. Depending on the requirements of your organization, you must configure your Capsule Server with a default or custom certificate.

- If you use a default SSL certificate, you must also configure each external Capsule Server with a distinct default SSL certificate. For more information, see Section 2.6.1, "Configuring Capsule Server with a Default SSL Certificate".
- If you use a custom SSL certificate, you must also configure each external Capsule Server with a distinct custom SSL certificate. For more information, see Section 2.6.2, "Configuring Capsule Server with a Custom SSL Certificate".

2.6.1. Configuring Capsule Server with a Default SSL Certificate

Use this section to configure Capsule Server with an SSL certificate that is signed by the Satellite Server default Certificate Authority (CA).

Prerequisites

Before configuring Capsule Server with a default server certificate, ensure that your Capsule Server meets the following conditions:

- Capsule Server is registered to Satellite Server. For more information, see Section 2.1, "Registering to Satellite Server".
- The Capsule Server packages are installed. For more information, see Section 2.5, "Installing Capsule Server Packages".

Procedure

To configure Capsule Server with a default server certificate, complete the following steps:

1. On Satellite Server, to store all the source certificate files for your Capsule Server, create a directory that is accessible only to the **root** user, for example **/root/capsule_cert**:

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

2. On Satellite Server, generate the ***/root/capsule_cert/capsule_certs.tar*** certificate archive for your Capsule Server:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar /root/capsule_cert/capsule_certs.tar
```

Retain a copy of the **satellite-installer** command that the **capsule-certs-generate** command returns for deploying the certificate to your Capsule Server.

Example output of capsule-certs-generate

```
Installing             Done                                [100%]
  Success!

  To finish the installation, follow these steps:

  If you do not have the Capsule registered to the Satellite instance, then please do the
following:

  1. yum -y localinstall http://satellite.example.com.com/pub/katello-ca-consumer-
latest.noarch.rpm
  2. subscription-manager register --org "Default_Organization"
```

Once this is completed run the steps below to start the Capsule installation:

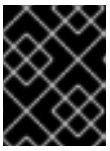
1. Ensure that the satellite-capsule package is installed on the system.
2. Copy the following file */root/capsule_cert/capsule_certs.tar* to the system *capsule.example.com* at the following location */root/capsule_certs.tar*
scp */root/capsule_cert/capsule_certs.tar root@capsule.example.com:/root/capsule_certs.tar*
3. Run the following commands on the Capsule (possibly with the customized parameters, see satellite-installer --scenario capsule --help and documentation for more info on setting up additional services):

```
satellite-installer \
--scenario capsule \
--certs-tar-file                    "/root/capsule_certs.tar"\
--foreman-proxy-content-parent-fqdn    "satellite.example.com"\
--foreman-proxy-register-in-foreman    "true"\
--foreman-proxy-foreman-base-url       "https://satellite.example.com"\
--foreman-proxy-trusted-hosts          "satellite.example.com"\
--foreman-proxy-trusted-hosts          "capsule.example.com"\
--foreman-proxy-oauth-consumer-key     "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f"\
--foreman-proxy-oauth-consumer-secret  "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"\
--puppet-server-foreman-url          "https://satellite.example.com"
```

3. On Satellite Server, copy the certificate archive file to your Capsule Server:

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

4. On Capsule Server, to deploy the certificate, enter the **satellite-installer** command that the **capsule-certs-generate** command returns.
When network connections or ports to Satellite are not yet open, you can set the **--foreman-proxy-register-in-foreman** option to **false** to prevent Capsule from attempting to connect to Satellite and reporting errors. Run the installer again with this option set to **true** when the network and firewalls are correctly configured.



IMPORTANT

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

2.6.2. Configuring Capsule Server with a Custom SSL Certificate

If you configure Satellite Server to use a custom SSL certificate, you must also configure each of your external Capsule Servers with a distinct custom SSL certificate.

To configure your Capsule Server with a custom certificate, complete the following procedures on each Capsule Server:

1. Section 2.6.2.1, “Creating a Custom SSL Certificate for Capsule Server”
2. Section 2.6.2.2, “Deploying a Custom SSL Certificate to Capsule Server”
3. Section 2.6.2.3, “Deploying a Custom SSL Certificate to Hosts”

2.6.2.1. Creating a Custom SSL Certificate for Capsule Server

On Satellite Server, create a custom certificate for your Capsule Server. If you already have a custom SSL certificate for Capsule Server, skip this procedure.

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

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

Procedure

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

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

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

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

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

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

3. Create the **/root/capsule_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 ] 1
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 = capsule.example.com 2

[ 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 = capsule.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 Capsule Server. To confirm a FQDN, on that Capsule 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/capsule_cert/capsule_cert_key.pem \ 1
-config /root/capsule_cert/openssl.cnf \ 2
-out /root/capsule_cert/capsule_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.

2.6.2.2. Deploying a Custom SSL Certificate to Capsule Server

Use this procedure to configure your Capsule Server with a custom SSL certificate signed by a Certificate Authority. The **satellite-installer** command, which the **capsule-certs-generate** command returns, is unique to each Capsule Server. Do not use the same command on more than one Capsule Server.

Prerequisites

Before configuring Capsule Server with a custom server certificate, ensure that your Satellite and Capsule Servers meet the following conditions:

- Satellite Server is configured with a custom certificate. For more information, see Configuring Satellite Server with a Custom Server Certificate in *Installing Satellite Server from a Connected Network*.

- Capsule Server is registered to Satellite Server. For more information, see Section 2.1, "Registering to Satellite Server".
- The Capsule Server packages are installed. For more information, see Section 2.5, "Installing Capsule Server Packages"

Procedure

To configure your Capsule Server with a custom SSL certificate, complete the following steps:

1. On Satellite Server, validate the custom SSL certificate input files:

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

1. Path to the Capsule 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 **capsule-certs-generate** commands, one of which you must use to generate the certificate archive file for your Capsule Server.

Example output of katello-certs-check

```
Validation succeeded.

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

capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
    --certs-tar  "~/$CAPSULE-certs.tar" \
    --server-cert "/root/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \

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

  capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
    --certs-tar  "~/$CAPSULE-certs.tar" \
    --server-cert "/root/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \
    --certs-update-server
```

2. On Satellite Server, from the output of the **katello-certs-check** command, depending on your requirements, enter the **capsule-certs-generate** command that generates a certificate for a new or existing Capsule. In this command, change **$CAPSULE** to the FQDN of your Capsule Server.
3. Retain a copy of the **satellite-installer** command that the **capsule-certs-generate** command returns for deploying the certificate to your Capsule Server.

Example output of capsule-certs-generate

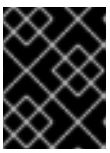
Example output of capsule-certs-generate

```
Installing              Done                                [100%]
  Success!

  To finish the installation, follow these steps:

  If you do not have the Capsule registered to the Satellite instance, then please do the
following:

  1. yum -y localinstall http://satellite.example.com.com/pub/katello-ca-consumer-
latest.noarch.rpm
  2. subscription-manager register --org "Default_Organization"

  Once this is completed run the steps below to start the Capsule installation:

  1. Ensure that the satellite-capsule package is installed on the system.
  2. Copy the following file /root/capsule_cert/capsule_certs.tar to the system
capsule.example.com at the following location /root/capsule_certs.tar
  scp /root/capsule_cert/capsule_certs.tar root@capsule.example.com:/root/capsule_certs.tar
  3. Run the following commands on the Capsule (possibly with the customized
     parameters, see satellite-installer --scenario capsule --help and
     documentation for more info on setting up additional services):

satellite-installer \
--scenario capsule \
--certs-tar-file                "/root/capsule_certs.tar"\
--foreman-proxy-content-parent-fqdn     "satellite.example.com"\
--foreman-proxy-register-in-foreman     "true"\
--foreman-proxy-foreman-base-url        "https://satellite.example.com"\
--foreman-proxy-trusted-hosts           "satellite.example.com"\
--foreman-proxy-trusted-hosts           "capsule.example.com"\
--foreman-proxy-oauth-consumer-key      "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f"\
--foreman-proxy-oauth-consumer-secret   "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"\
--puppet-server-foreman-url             "https://satellite.example.com"
```

4. On Satellite Server, copy the certificate archive file to your Capsule Server:

```
# scp /root/capsule_cert/capsule.example.com-certs.tar \
root@capsule.example.com:/root/capsule.example.com-certs.tar
```

5. On Capsule Server, to deploy the certificate, enter the **satellite-installer** command that the **capsule-certs-generate** command returns.
When network connections or ports to Satellite are not yet open, you can set the **--foreman-proxy-register-in-foreman** option to **false** to prevent Capsule from attempting to connect to Satellite and reporting errors. Run the installer again with this option set to **true** when the network and firewalls are correctly configured.



IMPORTANT

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

2.6.2.3. Deploying a Custom SSL Certificate to Hosts

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

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

Procedure

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

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

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

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

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

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON CAPSULE SERVER

Use this chapter to configure additional settings on your Capsule Server.

3.1. INSTALLING THE KATELLO AGENT

To remotely update Satellite clients, you must install the Katello agent.

The **katello-agent** package depends on the **gofer** package that provides the **goferd** service. This service must be enabled so that Satellite Server or Capsule Server can provide information about errata that are applicable for content hosts.

Prerequisites

Before installing the Katello agent, ensure the following conditions are met:

- You have enabled the Satellite Tools repository on Satellite Server. For more information, see Enabling the Satellite Tools Repository in *Installing Satellite Server from a Connected Network* .
- You have synchronized the Satellite Tools repository on Satellite Server. For more information, see Synchronizing the Satellite Tools Repository in *Installing Satellite Server from a Connected Network*.
- You have enabled the Satellite Tools repository on the client. For example, to ensure that the repository is enabled on the Red Hat Enterprise Linux 7 client, enter the following command on the client:

```
# subscription-manager repos --enable rhel-7-server-satellite-tools-6.6-rpms
```

Procedure

To install the Katello agent, complete the following steps:

1. Install the **katello-agent** package:

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

2. Start the **goferd** service :

```
# systemctl start goferd
```

3.2. ENABLING REMOTE EXECUTION ON CAPSULE SERVER

To run commands on hosts that are registered to Capsule Server, you must enable the remote execution feature on your Capsule.

Remote execution on external Capsules is disabled by default.

Procedure

- To enable remote execution on Capsule Server, enter the following command:

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

3.3. ENABLING OPENSCAP ON EXTERNAL CAPSULES

On Satellite Server and the integrated Capsule of your Satellite Server, OpenSCAP is enabled by default.

To use the OpenSCAP plug-in and content on an external Capsule, you must enable OpenSCAP on each Capsule.

Procedure

- To enable OpenSCAP, enter the following command:

```
# satellite-installer --scenario capsule \
--enable-foreman-proxy-plugin-openscap
```

3.4. ADDING LIFE CYCLE ENVIRONMENTS TO CAPSULE SERVERS

If your Capsule Server has the content functionality enabled, you must add an environment so that Capsule can synchronize content from Satellite Server and provide content to host systems.

Do not assign the *Library* lifecycle environment to your Capsule Server because it triggers an automated Capsule sync every time the CDN updates a repository. This might consume multiple system resources on Capsules, network bandwidth between Satellite and Capsules, and available disk space on Capsules.

You can use Hammer CLI on Satellite Server or the Satellite web UI.

Procedure

To add a life cycle environment to Capsule Server, complete the following steps:

1. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**, and select the Capsule that you want to add a life cycle to.
2. Click **Edit** and click the **Life Cycle Environments** tab.
3. From the left menu, select the life cycle environments that you want to add to Capsule and click **Submit**.
4. To synchronize the content on the Capsule, click the **Overview** tab and click **Synchronize**.
5. Select either **Optimized Sync** or **Complete Sync**.
   For definitions of each synchronization type, see Recovering a Repository in the *Content Management Guide*.

For CLI Users

1. To display a list of all Capsule Servers, on Satellite Server, enter the following command:

```
# hammer capsule list
```

   Note the Capsule ID of the Capsule that you want to add a life cycle to.

2. Using the ID, verify the details of your Capsule:

```
# hammer capsule info --id capsule_id
```

3. Display the life cycle environments that are available and note the environment ID:

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

4. To view the life cycle environments available for your Capsule Server, enter the following command and note the ID and the organization name:

```
# hammer capsule content available-lifecycle-environments --id capsule_id
```

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

```
# hammer capsule content add-lifecycle-environment \
--id capsule_id --organization "My_Organization" \
--environment-id environment_id
```

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

6. Synchronize the content from Satellite to Capsule.

- To synchronize all content from your Satellite Server environment to Capsule Server, enter the following command:

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

- To synchronize a specific life cycle environment from your Satellite Server to Capsule Server, enter the following command:

```
# hammer capsule content synchronize --id external_capsule_id \
--environment-id environment_id
```

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 Capsule Server.

Prerequisites

- All managed hosts must have a network interface of BMC type. Capsule 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 --scenario capsule \
--foreman-proxy-bmc "true" \
--foreman-proxy-bmc-default-provider "freeipmi"
```

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

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

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

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

Prerequisites

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

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

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

3.7. 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 Capsule 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
```

CHAPTER 4. CONFIGURING EXTERNAL SERVICES

Use this section to configure your Red Hat Satellite Capsule Server to work with external DNS, DHCP and TFTP services.

4.1. CONFIGURING CAPSULE SERVER WITH EXTERNAL DNS

You can configure Capsule Server with external DNS. Capsule 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

Before you can configure Capsule Server with external DNS, ensure that the following conditions are met:

- You must have a configured external DNS server.

Procedure

To configure Capsule Server with external DNS, complete the following steps:

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

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

2. Copy the **/etc/rndc.key** file from the external DNS server to Capsule 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. 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
```

6. Restart the foreman-proxy service:

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

7. Log in to the Satellite Server web UI and navigate to **Infrastructure** > **Capsules**.

8. Locate the Capsule Server that you want to configure with external DNS and from the list in the **Actions** column, select **Refresh**.

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

4.2. CONFIGURING CAPSULE SERVER WITH EXTERNAL DHCP

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

1. Section 4.2.1, "Configuring an External DHCP Server to Use with Capsule Server"
2. Section 4.2.2, "Configuring Capsule Server with an External DHCP Server"

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

To configure an external DHCP server to use with Capsule 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 Capsule Server. The example in this procedure uses the distributed Network File System (NFS) protocol to share the DHCP configuration and lease files.

Procedure

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

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

```
# yum install dhcp bind
```

2. Generate a security token:

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

As a result, a key pair that consists of two files is created in the current directory.

3. Copy the secret hash from the key:

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

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

```
# cat /etc/dhcp/dhcpd.conf
```

```
default-lease-time 604800;
max-lease-time 2592000;
log-facility local7;

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

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

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

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

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

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

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

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

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

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

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

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

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

11. Start the DHCP service:

```
# systemctl start dhcpd
```

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

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

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

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

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

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

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

```
# mount -a
```

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

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

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

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

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

17. Reload the NFS server:

```
# exportfs -rva
```

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

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

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

- Use the **firewalld** NFS service to configure the firewall:

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

-

4.2.2. Configuring Capsule Server with an External DHCP Server

You can configure Capsule Server with an external DHCP server.

Prerequisite

- Ensure that you have configured an external DHCP server and that you have shared the DHCP configuration and lease files with Capsule Server. For more information, see Section 4.2.1, “Configuring an External DHCP Server to Use with Capsule Server”.

Procedure

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

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

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

2. Create the DHCP directories for NFS:

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

3. Change the file owner:

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

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

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

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

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

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

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

```
# mount -a
```

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

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

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

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=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
```

9. Restart the foreman-proxy service:

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

10. Log in to the Satellite Server web UI.

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

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

4.3. CONFIGURING CAPSULE SERVER WITH EXTERNAL TFTP

You can configure Capsule Server with external TFTP services.

Procedure

To configure Capsule Server with external TFTP, complete the following steps:

1. Create the TFTP directory for NFS:

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

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

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

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

```
# mount -a
```

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

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

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

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

6. Log in to the Satellite Server web UI.

7. Navigate to **Infrastructure** > **Capsules**. Locate the appropriate Capsule Server and from the list in the **Actions** column, select **Refresh**.

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

4.4. CONFIGURING SATELLITE OR CAPSULE WITH EXTERNAL IDM DNS

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

The first method is to install the IdM client which automates the process with the *generic security service algorithm for secret key transaction* (GSS-TSIG) technology defined in RFC3645. This method requires installing the IdM client on the Satellite Server or Capsule's base system and having an account created by the IdM server administrator for use by the Satellite administrator. See Section 4.4.1, "Configuring Dynamic DNS Update with GSS-TSIG Authentication" to use this method.

The second method, *secret key transaction authentication for DNS* (TSIG), uses an **rndc.key** for authentication. It requires root access to the IdM server to edit the BIND configuration file, installing the **BIND** utility on the Satellite Server's base system, and coping the **rndc.key** to between the systems. This technology is defined in RFC2845. See Section 4.4.2, "Configuring Dynamic DNS Update with TSIG Authentication" to use this method.

NOTE

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

Determining where to install the IdM Client

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

4.4.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication

In this example, Satellite Server has the following settings.

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

The IdM server has the following settings.

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

Before you Begin.

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

Create a Kerberos Principal on the IdM Server.

1. Ensure you have a Kerberos ticket.

```
# kinit idm_user
```

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

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

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

Install and Configure the IdM Client.

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

1. Install the **ipa-client** package on Satellite Server or Capsule Server:
   - On Satellite Server, enter the following command:

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

   - On Capsule Server, enter the following command:

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

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

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

3. Ensure you have a Kerberos ticket.

```
# kinit admin
```

4. Remove any preexisting keytab.

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

5. Get the keytab created for this system.

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

NOTE

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

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

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

7. If required, check the keytab is valid.

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

Configure DNS Zones in the IdM web UI.

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

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

   e. Ensure **Dynamic update** is set to **True**.
   f. Enable **Allow PTR sync**.

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

2. Create and Configure the reverse zone.

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

b. Select **Add**.

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

d. Click **Add and Edit**

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

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

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

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

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

- On a Satellite Server's Base System.

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

- On a Capsule Server's Base System.

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

Restart the Satellite or Capsule's Proxy Service.

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

Update the Configuration in Satellite web UI.

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

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

4.4.2. Configuring Dynamic DNS Update with TSIG Authentication

In this example, Satellite Server has the following settings.

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

The IdM server has the following settings.

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

Before you Begin

1. Confirm the IdM Server is deployed and the host-based firewall has been configured correctly. For more information, see Port Requirements in the *Linux Domain Identity, Authentication, and Policy Guide*.
2. Obtain **root** user privileges on the IdM server.
3. Confirm if the Satellite or an external Capsule is managing DNS for a domain.

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

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

6. Make a backup of the answer file in case you have to revert the changes. See Specifying Installation Options for more information.

Enabling External Updates to the DNS Zone in the IdM Server

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

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

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

```
# systemctl reload named
```

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

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

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

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

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

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

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

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

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

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

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

Testing External Updates to the DNS Zone in the IdM Server

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

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

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

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

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

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

4. On Satellite Server, test the DNS entry.

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

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

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

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

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

7. Confirm that the DNS entry was removed.

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

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

4.4.3. Reverting to Internal DNS Service

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

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

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

```
# satellite-installer
```

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

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

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

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

Update the Configuration in Satellite web UI.

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

1. Navigate to **Infrastructure** > **Capsules**.
2. For each Capsule to be updated, from the **Actions** drop-down menu, select **Refresh**.
3. Configure the domain:
   a. Go to **Infrastructure** > **Domains** and select the domain name.
   b. On the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.
4. Configure the subnet:
   a. Go to **Infrastructure** > **Subnets** and select the subnet name.
   b. On the **Subnet** tab, set **IPAM** to **DHCP** or **Internal DB**.
   c. On the **Domains** tab, ensure the domain to be managed by the Satellite or Capsule is selected.
   d. On the **Capsules** tab, ensure **Reverse DNS Capsule** is set to the Capsule where the subnet is connected.
   e. Click **Submit** to save the changes.

CHAPTER 5. UNINSTALLING CAPSULE SERVER

Uninstalling Capsule Server deletes all applications that are used on the target system. If you use any applications or application data on the target system for purposes other than running Capsule Server, you must back up the information before uninstalling Capsule.

To uninstall Capsule, use the **katello-remove** command. Before deleting all packages and configuration files in the system, the **katello-remove** command displays two warnings that require your confirmation.

The **katello-remove** command deletes the following packages and configuration files:

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

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **All Hosts**.
2. From the **Edit** list to the right of the Capsule Server that you want to uninstall, select **Delete** .
3. Navigate to **Infrastructure** > **Capsule**.
4. From the **Edit** list to the right of the Capsule Server that you want to uninstall, select **Delete**.
5. On Capsule Server, enter the **katello-remove** command:

```
# katello-remove
```

For CLI Users

1. On Satellite Server, list all Capsule Servers and note the FQDN and ID of the Capsule Server that you want to uninstall:

```
# hammer capsule list
```

2. On Satellite Server, to delete Capsule Server from the Satellite hosts list, enter the **hammer host delete** command and specify the Capsule Server FQDN with the **--name** option:

```
# hammer host delete --name Capsule_Server_FQDN
```

3. On Satellite Server, to delete Capsule Server from the Satellite Capsules list, enter the **hammer capsule delete** command and specify the Capsule Server ID with the **--id** option:

```
# hammer capsule delete --id Capsule_Server_ID
```

4. On Capsule Server, enter the **katello-remove** command:

```
# katello-remove
```

APPENDIX A. CAPSULE SERVER SCALABILITY CONSIDERATIONS

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit. The tested limit is 17 Capsule Servers with 2 vCPUs on a Satellite Server with Red Hat Enterprise Linux 7. However, scalability is highly variable, especially when managing Puppet clients.

Capsule Server scalability when managing Puppet clients depends on the number of CPUs, the run-interval distribution, and the number of Puppet managed resources. The Capsule Server has a limitation of 100 concurrent Puppet agents running at any single point in time. Running more than 100 concurrent Puppet agents results in a 503 HTTP error.

For example, assuming that Puppet agent runs are evenly distributed with less than 100 concurrent Puppet agents running at any single point during a run-interval, a Capsule Server with 4 CPUs has a maximum of 1250-1600 Puppet clients with a moderate workload of 10 Puppet classes assigned to each Puppet client. Depending on the number of Puppet clients required, the Satellite installation can scale out the number of Capsule Servers to support them.

If you want to scale your Capsule Server when managing Puppet clients, the following assumptions are made:

- There are no external Puppet clients reporting directly to the Satellite 6 integrated Capsule.
- All other Puppet clients report directly to an external Capsule.
- There is an evenly distributed run-interval of all Puppet agents.

NOTE

Deviating from the even distribution increases the risk of filling the passenger request queue. The limit of 100 concurrent requests applies.

The following table describes the scalability limits using the recommended 4 CPUs.

Table A.1. Puppet Scalability Using 4 CPUs

Puppet Managed Resources per Host	Run-Interval Distribution
1	3000-2500
10	2400-2000
20	1700-1400

The following table describes the scalability limits using the minimum 2 CPUs.

Table A.2. Puppet Scalability Using 2 CPUs

Puppet Managed Resources per Host	Run-Interval Distribution
1	1700-1450

Puppet Managed Resources per Host	Run-Interval Distribution
10	1500-1250
20	850-700