Red Hat Satellite 6.5-beta

Installing Capsule Server

Installing Red Hat Satellite Capsule Server

Last Updated: 2019-05-09

Red Hat Satellite 6.5-beta 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

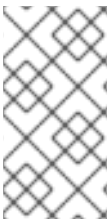
1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base system:

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

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

Satellite Server must be installed on a freshly provisioned system that serves no other function except to run Satellite Server.



NOTE

The Red Hat Satellite Server and Capsule Server versions must match. For example, a Satellite 6.2 Server cannot run a 6.5-beta Capsule Server and a Satellite 6.5-beta Server cannot run a 6.2 Capsule Server. Mismatching Satellite Server and Capsule Server versions results in the Capsule Server failing silently.

NOTE

Self-registered Satellites are not supported.

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

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

Certified hypervisors

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

FIPS Mode

You can install Satellite Server and 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 AND GUIDELINES

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

1.2.1. Storage Requirements

The following table details storage requirements for specific directories. These values are based on expected use case scenarios and can vary according to individual environments. Pay attention to your specific use case when reading the table. For example, you can have a Capsule Server without Pulp enabled, in which case you do not need the same level of storage requirements for directories related to Pulp such as **/var/lib/pulp/**.

In the following table, the runtime size was measured with Red Hat Enterprise Linux 5, 6, and 7 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	500 GB
/var/lib/mongodb/	3.5 GB	50 GB
/opt	500 MB	Not Applicable

1.2.2. 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 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*.

1.3. SUPPORTED OPERATING SYSTEMS

You can install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports. Red Hat Satellite Server and Red Hat Satellite Capsule Server are supported only on the latest versions of Red Hat Enterprise Linux 7 Server that is available at the time when Satellite 6.5-beta is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

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

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

1.4. PORTS AND FIREWALLS REQUIREMENTS

To enable the components of Satellite architecture to communicate, specific network ports and network-based firewalls must be open and free on the base operating system that you want to install Capsule on. The installation of a Capsule Server fails if the ports between Satellite Server and Capsule Server are not open before installation starts.

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

Integrated Capsule

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

Clients of Capsule

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

Required ports can change based on your configuration.

Table 1.2. Ports for 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
5000	TCP	HTTPS	Connection to Katello for the Docker registry
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)

Table 1.3. Ports for Capsule to Satellite Communication

Port	Protocol	Service	Required For
80	TCP	HTTP	Anaconda, yum, and for obtaining Katello certificate updates
443	TCP	HTTPS	Connections to Katello, Foreman, Foreman API, and Pulp
5646	TCP	amqp	Capsule's Qpid dispatch router to Qpid dispatch router in Satellite
5647	TCP	amqp	Katello agent to communicate with Satellite's Qpid dispatch router
5000	TCP	HTTPS	Connection to Katello for the Docker registry

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.

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

You can enable incoming connections from Satellite Server and clients to Capsule Server and make these rules persistent during reboots. If you do not use an external Capsule Server, you do not need to enable this connection.

For more information on the ports used, see Ports and Firewalls Requirements.

1. Configure the firewall on the base system on which you want to install Capsule:

```
# 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
```

1.6. VERIFYING FIREWALL SETTINGS

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

To verify firewall settings:

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

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

CHAPTER 2. INSTALLING CAPSULE SERVER

Before you install Capsule Server, you should ensure that your environment meets the requirements for installation. Capsule Server has the same requirements for installation as Satellite Server, with the additional requirement that you have not configured it to use a proxy to connect to the Red Hat CDN. 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 to Satellite Server.

Subscription Manifest Considerations

- The Satellite Server must have a manifest installed with the appropriate repositories for the organization you want the future 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 the 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 Considerations

- The Satellite Server's base system must be able to resolve the host name of the base system on which you want to install Capsule and vice versa.
- You must revert any changes related to the use of proxies which prevent access to Red Hat Satellite.
- You must have configured host and network-based firewalls. For more information, see Section 1.4, "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* .

Register to Satellite Server

1. Install the Satellite Server's CA certificate on the base system on which you want to install Capsule.

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

2. Register the base system on which you want to install Capsule with the environments that you want the future 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. IDENTIFYING AND ATTACHING THE CAPSULE SERVER SUBSCRIPTION

After you have registered the Capsule Server, you must identify your Capsule Server subscription Pool ID. The Pool ID enables you to attach the required subscription to your Capsule Server. The Capsule Server subscription provides access to the Capsule Server content, as well as Red Hat Enterprise Linux, Red Hat Software Collections (RHSCL), and Red Hat Satellite. This is the only subscription required.

1. Identify your Capsule Server subscription.

```
# subscription-manager list --all --available
```

The command displays output similar to the following:

```
+-------------------------------------------+
   Available Subscriptions
+-------------------------------------------+

Subscription Name: Red Hat Satellite Capsule Server
Provides:          Red Hat Satellite Proxy
                   Red Hat Satellite Capsule
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux Server
                   Red Hat Enterprise Linux High Availability (for RHEL Server)
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Enterprise Linux Load Balancer (for RHEL Server)
SKU:               MCT0369
Pool ID:           9e4cc4e9b9fb407583035861bb6be501
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2022
System Type:       Physical
```

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

3. Attach your subscription to your Capsule Server, using your Pool ID:

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

The outputs displays something similar to the following:

```
Successfully attached a subscription for: Red Hat Capsule Server
```

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

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

2.3. CONFIGURING REPOSITORIES

1. Disable all existing repositories.

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

2. Enable the Red Hat Satellite Capsule, Red Hat Enterprise Linux, and Red Hat Software Collections repositories.
   The Red Hat Software Collections repository provides a later version of Ruby required by some Red Hat Satellite Capsule features, including the Remote Execution feature.

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

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

```
# yum clean all
```

4. Verify that the repositories have been enabled.

```
# yum repolist enabled
```

2.4. SYNCHRONIZING TIME

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

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

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

Synchronizing Time using chronyd

1. Install chronyd.

```
# yum install chrony
```

2. Start and enable the chronyd service.

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

2.5. INSTALLING CAPSULE SERVER

1. Update all packages.

```
# yum update
```

2. Install the installation package.

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

2.6. PERFORMING INITIAL CONFIGURATION OF CAPSULE SERVER

This section demonstrates a default installation of Capsule Server, including use of default certificates, DNS, and DHCP configuration. For details of more advanced configuration options, see Performing Additional Configuration on Capsule Server.

2.6.1. Configuring Capsule Server with a Default Server Certificate

You can use the default certificate authority (CA) that comes with Capsule Server, which is used by both the server and the client SSL certificates for the authentication of subservices.

If you configured Satellite Server to use a custom SSL certificate, proceed to Section 3.6, "Configuring Capsule Server with a Custom Server Certificate".

Before You Begin

- Ensure that Capsule is installed and **satellite-installer** package is available on Capsule Server.
- You must have configured host and network-based firewalls. For more information, see Section 1.4, "Ports and Firewalls Requirements".
- You must have installed the **katello-ca-consumer-latest** package. For more information, see Section 2.1, "Registering to Satellite Server".
- You must have registered your Capsule Server to the Satellite Server.
- You must have attached the required subscription to the Capsule Server.

Configure Capsule Server with a Default Server Certificate

1. On Satellite Server, create the certificates archive:

```
# capsule-certs-generate \
--foreman-proxy-fqdn mycapsule.example.com \
--certs-tar mycapsule.example.com-certs.tar
```

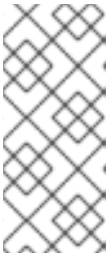
   Retain a copy of the **satellite-installer** command that is output by the **capsule-certs-generate** command for installing the Capsule Server certificates.

2. Copy the generated archive **.tar** file from Satellite Server to Capsule Server.

3. On Capsule Server, run the **satellite-installer** command that the **capsule-certs-generate** command outputs to install Capsule Server certificates:

```
# satellite-installer --scenario capsule \
--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 mycapsule.example.com \
```

```
--foreman-proxy-oauth-consumer-key UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV \
--foreman-proxy-oauth-consumer-secret ZhH8p7M577ttNU3WmUGWASag3JeXKgUX \
--foreman-proxy-content-certs-tar mycapsule.example.com-certs.tar \
--puppet-server-foreman-url "https://satellite.example.com"
```



NOTE

When network connections or ports to the 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.

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON CAPSULE SERVER

3.1. INSTALLING THE KATELLO AGENT

Installing the katello agent is recommended to allow remote updates of clients. The base system of a Capsule Server is a client of Satellite Server and therefore should also have the katello agent installed.

Before You Begin

- You must have enabled the Satellite Tools repositories in Satellite Server.
- You must have synchronized the Satellite Tools repositories in Satellite Server.

To Install katello-agent:

1. Log into the system.
2. Enable the Satellite tools repository for this version of Satellite.

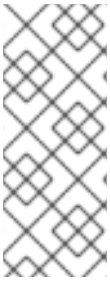
```
# subscription-manager repos \
--enable=rhel-7-server-satellite-tools-6-beta-rpms
```

3. Install the package.

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

3.2. ENABLING REMOTE EXECUTION ON CAPSULE SERVER

If you want to run commands on a Capsule Server's hosts, ensure that you enable the remote execution.



NOTE

Remote execution on external Capsules is disabled by default. To use remote execution on a Capsule Server you need to enable it by running the following command:

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

3.3. 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 step:

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 then click **Submit**.
4. To synchronize Capsule's content, click the **Overview** tab, and then click **Synchronize**.
5. Select either **Optimized Sync** or **Complete Sync**.

For CLI Users

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

```
# hammer capsule list
```

   Note the ID that returns.

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

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

3. Verify the life cycle environments 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.

   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
```

3.4. ENABLING POWER MANAGEMENT ON MANAGED HOSTS

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

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

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

Before You Begin

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

Enable Power Management on Managed Hosts

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

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

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

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

You can also configure Capsule Server to use external DNS and DHCP services. See Configuring External Services for more information.

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

Before You Begin

- 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.

Configure DNS, DHCP, and TFTP on Capsule Server

1. Run capsule installer with the options applicable to your environment.
   The following example shows 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-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername $(hostname)
```

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

3.6. CONFIGURING CAPSULE SERVER WITH A CUSTOM SERVER CERTIFICATE

Red Hat Satellite 6 includes default SSL certificates to enable encrypted communications between the Satellite Server, Capsule Servers, and all hosts. You can replace the default certificates with custom certificates if required. For example, your company's security policy might dictate that SSL certificates must be obtained from a specific Certificate Authority.

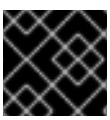
Prerequisites

- Satellite Server configured with custom certificates. For more information, see Configuring Satellite Server with a Custom Server Certificate in *Installing Satellite Server from a Connected Network*.
- Capsule Server installed and registered to the Satellite Server. For more information, see Chapter 2, *Installing Capsule Server*.

To use custom certificates on each Capsule Server, complete these procedures:

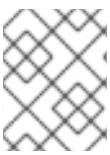
1. Section 3.6.1, "Obtain an SSL Certificate for Capsule Server"
2. Section 3.6.2, "Validate the Capsule Server's SSL Certificate"
3. Section 3.6.3, "Create the Capsule Server's Certificate Archive File"
4. Section 3.6.4, "Install the Capsule Server's Custom Certificate"
5. Section 3.6.5, "Install the Capsule Server's New Certificate on All Hosts"

3.6.1. Obtain an SSL Certificate for Capsule Server



IMPORTANT

Use PEM encoding for the SSL Certificates.



NOTE

- Do **not** use the Satellite Server's certificate on any Capsule Server as each server's certificate is unique.

Procedure

On Satellite Server, obtain custom SSL certificates for Capsule Server:

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

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

   In these examples, the directory is **/root/capsule_cert**. If you have multiple Capsule Servers, name the directory to match. For example, if you have Capsule Servers named **capsule_apac** and **capsule_emea**, you might create directories named *capsule_apac* and *capsule_emea* respectively. This is not *required*, but reduces the risk of using files from one Capsule Server on another Capsule Server.

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

NOTE

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

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

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

NOTE

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

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

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

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

[ v3_req ]
# Extensions to add to a certificate request
```

```
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
```

4. Generate the Certificate Signing Request (CSR):

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

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

3.6.2. Validate the Capsule Server's SSL Certificate

On the Satellite Server, validate the Capsule Server's certificate input files with the **katello-certs-check** command. This process requires that you have copied the Capsule Server key, CSR, and SSL certificate from Capsule Server to Satellite Server.

```
# 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. Capsule Server certificate file, provided by your Certificate Authority
2. Capsule Server's private key, used to sign the certificate
3. Certificate Authority bundle, provided by your Certificate Authority

If the certificate is successfully validated, the output contains the following information.

```
Validation succeeded
```

Retain a copy of the example **capsule-certs-generate** command that is output by the **katello-certs-check** command for use in the following procedure.

Proceed to Section 3.6.3, “Create the Capsule Server’s Certificate Archive File” .

3.6.3. Create the Capsule Server’s Certificate Archive File

The Capsule Server’s installer requires the server certificates to be in an archive file. To create this file, use the **capsule-certs-generate** command on the Satellite Server.

The **capsule-certs-generate** command must be run once for every external Capsule Server. In these examples, **capsule.example.com** is the example FQDN and **capsule_certs.tar** the example archive file’s name. Replace these with values appropriate to your environment, taking care not to overwrite an existing certificate archive file. For example, if you have Capsule Servers named **capsule1** and **capsule2**, you can name the certificate archive files **capsule1_certs.tar** and **capsule2_certs.tar**.

The **capsule-certs-generate** command, including parameters, is output by the **katello-certs-check** when run on Satellite Server. For more information, see Configuring Satellite Server with a Custom Server Certificate in *Installing Satellite Server from a Connected Network* .

1. In an editor, prepare a copy of the **capsule-certs-generate** command.
2. Edit the values for **--foreman-proxy-fqdn** to match the Capsule Server’s FQDN, and **--certs-tar** to the file path and name for the certificate archive file.
3. If the Capsule Server has not already been installed, remove the **--certs-update-server** parameter. This is used only to **update** an existing Capsule Server’s certificate.
4. Copy the modified **capsule-certs-generate** command from the text editor to the terminal.
5. Run the modified **capsule-certs-generate** command.

   Example capsule-certs-generate command

   ```
   # capsule-certs-generate --foreman-proxy-fqdn capsule.example.com \
   --certs-tar  /root/capsule_cert/capsule_certs.tar \
   --server-cert /root/capsule_cert/capsule_cert.pem \
   --server-key /root/capsule_cert/capsule_cert_key.pem \
   --server-ca-cert /root/sat_cert/ca_cert_bundle.pem \
   --certs-update-server
   ```

6. On the Satellite Server, copy the certificate archive file to the Capsule Server, providing the **root** user’s password when prompted.
   In this example the archive file is copied to the **root** user’s home directory, but you may prefer to copy it elsewhere.

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

Retain a copy of the example **satellite-installer** command that is output by the **capsule-certs-generate** command for use in the following procedure.

Proceed to Section 3.6.4, “Install the Capsule Server’s Custom Certificate” .

3.6.4. Install the Capsule Server’s Custom Certificate

WARNING

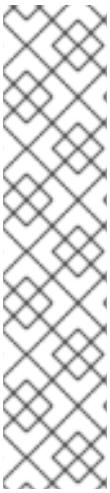
Complete this procedure on the Capsule Server.

To install the Capsule Server's custom certificates, run the **satellite-installer** script with custom parameters. The command, including parameters, is output by the **capsule-certs-generate** command in Section 3.6.3, "Create the Capsule Server's Certificate Archive File" .

1. In an editor, prepare a copy of the **satellite-installer** command.
2. Edit the value for **--foreman-proxy-content-certs-tar** to match the location of the certificates archive file.
3. If you want to enable additional features on the Capsule Server, append their parameters to the **satellite-installer** command. For a description of all the installer's parameters, enter the command **satellite-installer --scenario capsule --help**.
4. Copy the modified **satellite-installer** command from the text editor to the terminal.
5. Run the modified **satellite-installer** command.

Example custom satellite-installer command

```
# satellite-installer --scenario capsule \
--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 "FeQsbASvCjvvaqE6duKH6SoYZWg4jwjg" \
--foreman-proxy-oauth-consumer-secret "7UhPXFPDBongvdTbNixbsWR5WFZsKEgF" \
--foreman-proxy-content-certs-tar "/root/capsule_certs.tar" \
--puppet-server-foreman-url "https://satellite.example.com"
```



NOTE

The **satellite-installer** command, as output by the **capsule-certs-generate** command, is unique to each Capsule Server. Do **not** use the same command on more than one Capsule Server.

Do **NOT** delete the certificates archive file (the .tar file) even after the certificates have been deployed to all relevant hosts. It is required, for example, when upgrading the Capsule Server. If the certificates archive file is not found by the installer, it fails with a message similar to the following:

```
[ERROR YYYY-MM-DD hh:mm:ss main]  tar -xzf /var/tmp/srvcapsule01.tar returned 2
instead of one of [0]
```

Proceed to Section 3.6.5, "Install the Capsule Server's New Certificate on All Hosts" .

3.6.5. Install the Capsule Server's New Certificate on All Hosts

Hosts which connect to an external Capsule Server require that server's custom certificate. Run the following command on all the Capsule Server's hosts.

NOTE

Use the Capsule Server's host name, **not** that of the Satellite Server.

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

3.7. RESTRICTING ACCESS TO MONGOD

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

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

1. Configure the Firewall.

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

2. Make the changes persistent:

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

CHAPTER 4. CONFIGURING EXTERNAL SERVICES

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

1. On the Red Hat Enterprise Linux Server, install the ISC DNS Service.

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

   Ensure that the **nsupdate** utility was installed. The Capsule uses the **nsupdate** utility to update DNS records on the remote server.

2. Copy the **/etc/rndc.key** file from the services server to the Capsule Server.

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

3. 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
```

4. Test the **nsupdate** utility by adding a host remotely.

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

5. Run the **satellite-installer** script 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="192.168.38.2" \
--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.

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

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 have previously configured a DHCP server, and shared the DHCP configuration and lease files via NFS.

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

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

```
# yum install dhcp bind
```

2. Generate a security token in an empty directory.

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

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

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

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

3. Copy the secret hash from the key.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11. Start the DHCP service.

```
# systemctl start dhcpd
```

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

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

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

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

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

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

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

```
# mount -a
```

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

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

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

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

17. Reload the NFS server.

```
# exportfs -rva
```

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

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

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

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

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

To Configure Capsule Server with External DHCP

1. Install the NFS client.

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

2. Create the DHCP directories for NFS.

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

3. Change the file owner.

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

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

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

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

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

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

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

```
# mount -a
```

7. Read the relevant files.

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

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

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

9. Restart the foreman-proxy service.

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

10. Log in to the Satellite Server web UI.

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

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

4.3. CONFIGURING CAPSULE SERVER WITH EXTERNAL TFTP

1. Create the TFTP directory to prepare for NFS.

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

2. Add the following line in the **/etc/fstab** file:

```
192.168.38.2:/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. Run the **satellite-installer** script 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 that server.

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

This updates all configuration files with the new value.

6. Log in to the Satellite Server web UI.

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

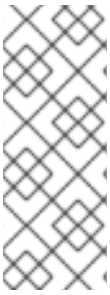
8. Associate the TFTP service with the appropriate subnets and domain.

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 IdM client package.

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

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

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

3. Ensure you have a Kerberos ticket.

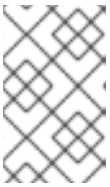
```
# kinit admin
```

4. Remove any preexisting keytab.

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

5. Get the keytab created for this system.

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



NOTE

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

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

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

7. If required, check the keytab is valid.

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

Configure DNS Zones in the IdM web UI.

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

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

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

f. Enable **Allow PTR sync**.

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

2. Create and Configure the reverse zone.

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

b. Select **Add**.

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

d. Click **Add and Edit**

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

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

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

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

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

- On a Satellite Server's Base System.

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

- On a Capsule Server's Base System.

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

Restart the Satellite or Capsule's Proxy Service.

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

Update the Configuration in Satellite web UI.

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

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

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

Before you Begin

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

WARNING

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

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

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **All Hosts** and select **Delete** from the **Edit** list to the right of the Capsule Server instance.
2. Navigate to **Infrastructure** > **Capsule** and select **Delete** from the **Edit** list to the right of the Capsule Server instance.
3. On Capsule Server, enter the **katello-remove** command to uninstall Capsule Server:

```
# katello-remove
```

For CLI Users

1. On Satellite Server, list all Capsule Servers to find the FQDN and ID of the Capsule Server instance you want to remove:

```
# hammer capsule list
```

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

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

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

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

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

```
# katello-remove
```

APPENDIX A. CAPSULE SERVER SCALABILITY CONSIDERATIONS

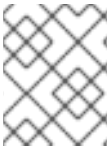
The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit. The tested limit is 17 Capsule Servers with 2 vCPUs on a Satellite Server with Red Hat Enterprise Linux 7. However, scalability is highly variable, especially when managing Puppet clients.

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