Red Hat Satellite 6.15

Installing Capsule Server

Install and configure Capsule

Red Hat Satellite 6.15 Installing Capsule Server

Install and configure Capsule

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

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

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

Prerequisites

- Ensure you have registered a Red Hat account.

Procedure

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

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

3. Click **Create**.

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base operating system:

- x86_64 architecture

- The latest version of Red Hat Enterprise Linux 8

- 4-core 2.0 GHz CPU at a minimum

- A minimum of 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

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

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

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

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

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

> **WARNING**
>
> The version of Capsule must match with the version of Satellite installed. It should not be different. For example, the Capsule version 6.15 cannot be registered with the Satellite version 6.14.

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:

- apache

- foreman-proxy

- postgres

- pulp

- puppet

- redis

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

SELinux mode

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

Synchronized system clock

The system clock on the base operating system where you are installing your Capsule Server must be synchronized across the network. If the system clock is not synchronized, SSL certificate verification might fail. For example, you can use the Chrony suite for timekeeping. For more information, see Configuring time synchronization in *Red Hat Enterprise Linux 8 Configuring basic system settings* .

FIPS mode

You can install Capsule on a Red Hat Enterprise Linux system that is operating in FIPS mode. You cannot enable FIPS mode after the installation of Capsule. For more information, see Switching RHEL to FIPS mode in *Red Hat Enterprise Linux 8 Security hardening* .

> **NOTE**
>
> Satellite supports DEFAULT and FIPS crypto-policies. The FUTURE crypto-policy is not supported for Satellite and Capsule installations. The FUTURE policy is a stricter forward-looking security level intended for testing a possible future policy. For more information, see Using system-wide cryptographic policies in the Red Hat Enterprise Linux guide.

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/lib/pulp	1 MB	300 GB
/var/lib/pgsql	100 MB	20 GB
/usr	3 GB	Not Applicable
/opt/puppetlabs	500 MB	Not Applicable

The size of the PostgreSQL database on your Capsule Server can grow significantly with an increasing number of lifecycle environments, content views, or repositories that are synchronized from your Satellite Server. In the largest Satellite environments, the size of **/var/lib/pgsql** on Capsule Server can grow to double or triple the size of **/var/lib/pgsql** on your Satellite Server.

1.3. STORAGE GUIDELINES

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

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

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

- Use high-bandwidth, low-latency storage for the **/var/lib/pulp/** directories. As Red Hat Satellite has many operations that are I/O intensive, using high latency, low-bandwidth storage causes performance degradation. Ensure your installation has a speed in the range 60 – 80 Megabytes per second.

You can use the **storage-benchmark** script to get this data. For more information on using the **storage-benchmark** script, see Impact of Disk Speed on Satellite Operations .

File system guidelines

- Do not use the GFS2 file system as the input-output latency is too high.

Log file storage

Log files are written to **/var/log/messages/, /var/log/httpd/**, and **/var/lib/foreman-proxy/openscap/content/**. You can manage the size of these files using **logrotate**. For more information, see How to use logrotate utility to rotate log files .

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

SELinux considerations for NFS mount

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

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

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

```
# restorecon -R /var/lib/pulp
```

Duplicated packages

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

Symbolic links

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

Synchronized RHEL ISO

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

1.4. SUPPORTED OPERATING SYSTEMS

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

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

Table 1.2. Operating systems supported by satellite-installer

Operating System	Architecture	Notes
Red Hat Enterprise Linux 8	x86_64 only	

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

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. PORT AND FIREWALL REQUIREMENTS

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

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 operating system on which Capsule Server is running.

Clients of Capsule

Hosts which are clients of Capsules, other than Satellite's integrated Capsule, do not need access to Satellite Server. For more information on Satellite Topology, see Capsule Networking in *Overview, concepts, and deployment considerations*.

Required ports can change based on your configuration.

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

Table 1.3. Capsule incoming traffic

Destination Port	Protocol	Service	Source	Required For	Description
53	TCP and UDP	DNS	DNS Servers and clients	Name resolution	DNS (optional)
67	UDP	DHCP	Client	Dynamic IP	DHCP (optional)
69	UDP	TFTP	Client	TFTP Server (optional)	
443, 80	TCP	HTTPS, HTTP	Client	Content Retrieval	Content
443, 80	TCP	HTTPS, HTTP	Client	Content Host Registration	Capsule CA RPM installation
443	TCP	HTTPS	Red Hat Satellite	Content Mirroring	Management
443	TCP	HTTPS	Red Hat Satellite	Capsule API	Smart Proxy functionality

Destination Port	Protocol	Service	Source	Required For	Description
443	TCP	HTTPS	Client	Content Host registration	Initiation Uploading facts Sending installed packages and traces
1883	TCP	MQTT	Client	Pull based REX (optional)	Content hosts for REX job notification (optional)
8000	TCP	HTTP	Client	Provisioning templates	Template retrieval for client installers, iPXE or UEFI HTTP Boot
8000	TCP	HTTP	Client	PXE Boot	Installation
8140	TCP	HTTPS	Client	Puppet agent	Client updates (optional)
8443	TCP	HTTPS	Client	Content Host registration	Deprecated and only needed for Client hosts deployed before upgrades
9090	TCP	HTTPS	Red Hat Satellite	Capsule API	Capsule functionality
9090	TCP	HTTPS	Client	Register Endpoint	Client registration with an external Capsule Server
9090	TCP	HTTPS	Client	OpenSCAP	Configure Client (if the OpenSCAP plugin is installed)
9090	TCP	HTTPS	Discovered Node	Discovery	Host discovery and provisioning (if the discovery plugin is installed)

Destination Port	Protocol	Service	Source	Required For	Description
9090	TCP	HTTPS	Client	Pull based REX (optional)	Content hosts for REX job notification (optional)

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

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

Table 1.4. Capsule outgoing traffic

Destination Port	Protocol	Service	Destination	Required For	Description
	ICMP	ping	Client	DHCP	Free IP checking (optional)
7	TCP	echo	Client	DHCP	Free IP checking (optional)
22	TCP	SSH	Target host	Remote execution	Run jobs
53	TCP and UDP	DNS	DNS Servers on the Internet	DNS Server	Resolve DNS records (optional)
53	TCP and UDP	DNS	DNS Server	Capsule DNS	Validation of DNS conflicts (optional)
68	UDP	DHCP	Client	Dynamic IP	DHCP (optional)
443	TCP	HTTPS	Satellite	Capsule	Capsule Configuration management Template retrieval OpenSCAP Remote Execution result upload

Destination Port	Protocol	Service	Destination	Required For	Description
443	TCP	HTTPS	Red Hat Portal	SOS report	Assisting support cases (optional)
443	TCP	HTTPS	Satellite	Content	Sync
443	TCP	HTTPS	Satellite	Client communication	Forward requests from Client to Satellite
443	TCP	HTTPS	Infoblox DHCP Server	DHCP management	When using Infoblox for DHCP, management of the DHCP leases (optional)
623			Client	Power management	BMC On/Off/Cycle/Status
7911	TCP	DHCP, OMAPI	DHCP Server	DHCP	The DHCP target is configured using **--foreman-proxy-dhcp-server** and defaults to localhost ISC and **remote_isc** use a configurable port that defaults to 7911 and uses OMAPI
8443	TCP	HTTPS	Client	Discovery	Capsule sends reboot command to the discovered host (optional)

NOTE

ICMP to Port 7 UDP and TCP must not be rejected, but can be dropped. The DHCP Capsule sends an ECHO REQUEST to the Client network to verify that an IP address is free. A response prevents IP addresses from being allocated.

1.6. 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. Open the ports for clients on Capsule Server:

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

2. Allow access to services on Capsule Server:

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

3. Make the changes persistent:

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

Verification

- Enter the following command:

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

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

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 Preparing your Environment for Installation .

2.1. REGISTERING TO SATELLITE SERVER

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

Red Hat 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 operating 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 Red Hat Subscriptions in *Managing content*.

Proxy and network prerequisites

- The Satellite Server base operating system must be able to resolve the host name of the Capsule base operating system and vice versa.

- Ensure HTTPS connection using client certificate authentication is possible between Capsule Server and Satellite Server. HTTP proxies between Capsule Server and Satellite Server are not supported.

- You must configure the host and network-based firewalls accordingly. For more information, see Port and firewall requirements in *Installing Capsule Server*. You can register hosts with Satellite using the host registration feature in the Satellite web UI, Hammer CLI, or the Satellite API. For more information, see Registering Hosts in *Managing hosts*.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Register Host**.

2. From the **Activation Keys** list, select the activation keys to assign to your host.

3. Click **Generate** to create the registration command.

4. Click on the *files* icon to copy the command to your clipboard.

5. Connect to your host using SSH and run the registration command.

6. Check the **/etc/yum.repos.d/redhat.repo** file and ensure that the appropriate repositories have been enabled.

CLI procedure

1. Generate the host registration command using the Hammer CLI:

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key"
```

If your hosts do not trust the SSL certificate of Satellite Server, you can disable SSL validation by adding the **--insecure** flag to the registration command.

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key" \
--insecure true
```

2. Connect to your host using SSH and run the registration command.

3. Check the **/etc/yum.repos.d/redhat.repo** file and ensure that the appropriate repositories have been enabled.

API procedure

1. Generate the host registration command using the Satellite API:

```
# curl -X POST https://satellite.example.com/api/registration_commands \
--user "My_User_Name" \
-H 'Content-Type: application/json' \
-d '{ "registration_command": { "activation_keys": ["My_Activation_Key_1,
My_Activation_Key_2"] }}'
```

If your hosts do not trust the SSL certificate of Satellite Server, you can disable SSL validation by adding the **--insecure** flag to the registration command.

```
# curl -X POST https://satellite.example.com/api/registration_commands \
--user "My_User_Name" \
-H 'Content-Type: application/json' \
-d '{ "registration_command": { "activation_keys": ["My_Activation_Key_1,
My_Activation_Key_2"], "insecure": true }}'
```

Use an activation key to simplify specifying the environments. For more information, see Managing Activation Keys in *Managing content*.

To enter a password as a command line argument, use **username:password** syntax. Keep in mind this can save the password in the shell history. Alternatively, you can use a temporary personal access token instead of a password. To generate a token in the Satellite web UI, navigate to **My Account** > **Personal Access Tokens**.

2. Connect to your host using SSH and run the registration command.

3. Check the **/etc/yum.repos.d/redhat.repo** file and ensure that the appropriate repositories have been enabled.

2.2. CONFIGURING REPOSITORIES

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

1. Disable all repositories:

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

2. Enable the following repositories:

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

3. Enable the module:

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

NOTE

If there is any warning about conflicts with Ruby or PostgreSQL while enabling **satellite-capsule:el8** module, see Troubleshooting DNF modules. For more information about modules and lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux Application Streams Lifecycle.

NOTE

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

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

```
# dnf repolist enabled
```

2.3. OPTIONAL: USING FAPOLICYD ON CAPSULE SERVER

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

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

2.3.1. Installing fapolicyd on Capsule Server

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

Prerequisites

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

Procedure

1. For a new installation, install fapolicyd:

```
# dnf install fapolicyd
```

2. For an existing installation, install fapolicyd using satellite-maintain packages install:

```
# satellite-maintain packages install fapolicyd
```

3. Start the **fapolicyd** service:

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

Verification

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

```
# systemctl status fapolicyd
```

New Satellite Server or Capsule Server installations

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

Additional resources

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

2.4. INSTALLING CAPSULE SERVER PACKAGES

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

Procedure

To install Capsule Server, complete the following steps:

1. Update all packages:

```
# dnf upgrade
```

2. Install the Capsule Server packages:

```
# dnf install satellite-capsule
```

2.5. 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 2.5.1, "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 2.5.2, "Configuring Capsule Server with a custom SSL certificate".

2.5.1. Configuring Capsule Server with a default SSL certificate

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

Prerequisites

- Capsule Server is registered to Satellite Server. For more information, see Registering to Satellite Server.

- Capsule Server packages are installed. For more information, see Installing Capsule Server Packages.

Procedure

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.example.com*-**certs.tar** certificate archive for your Capsule Server:

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

   ```
   output omitted
   satellite-installer --scenario capsule \
   --certs-tar-file "/root/capsule_cert/capsule.example.com-certs.tar" \
   --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"
   ```

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.

2.5.2. 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 2.5.2.1, "Creating a custom SSL certificate for Capsule Server"

2. Section 2.5.2.2, "Deploying a custom SSL certificate to Capsule Server"

3. Section 2.5.2.3, "Deploying a custom SSL certificate to hosts"

2.5.2.1. 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.

Procedure

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 CSR and include the following content:

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

   [ req_distinguished_name ]
   commonName = capsule.example.com

   [ v3_req ]
   ```

```
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
subjectAltName = @alt_names

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

For more information about the **[v3_req]** parameters and their purpose, see RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

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

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

1 Two letter code

2 Full name

3 Full name (example: New York)

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

5. Generate CSR:

```
# openssl req -new \
-key /root/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

6. Send the certificate signing request to the certificate authority (CA). The same CA must sign certificates for Satellite Server and Capsule Server.
When you submit the request, specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the CA for the preferred method. In response to the request, you can expect to receive a CA bundle and a signed certificate, in separate files.

2.5.2.2. 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

- Satellite Server is configured with a custom certificate. For more information, see Configuring Satellite Server with a Custom SSL Certificate in *Installing Satellite Server in a connected network environment*.

- Capsule Server is registered to Satellite Server. For more information, see Registering to Satellite Server.

- Capsule Server packages are installed. For more information, see Installing Capsule Server Packages.

Procedure

1. On your Satellite Server, generate a certificate bundle:

   ```
   # capsule-certs-generate \
   --foreman-proxy-fqdn capsule.example.com \
   --certs-tar ~/capsule.example.com-certs.tar \
   --server-cert /root/capsule_cert/capsule_cert.pem \     1
   --server-key /root/capsule_cert/capsule_cert_key.pem \  2
   --server-ca-cert /root/capsule_cert/ca_cert_bundle.pem  3
   ```

   1   Path to Capsule Server certificate file that is signed by a Certificate Authority.

   2   Path to the private key that was used to sign Capsule Server certificate.

   3   Path to the Certificate Authority bundle.

2. 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

   ```
   output omitted
   satellite-installer --scenario capsule \
   --certs-tar-file "/root/capsule.example.com-certs.tar" \
   --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 "My_OAuth_Consumer_Key" \
   --foreman-proxy-oauth-consumer-secret "My_OAuth_Consumer_Secret"
   ```

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

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

4. On your Capsule Server, to deploy the certificate, enter the **satellite-installer** command that the **capsule-certs-generate** command returns.

   If 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.

2.5.2.3. Deploying a custom SSL certificate to hosts

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

Procedure

- Update the SSL certificate on each host:

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

2.6. ASSIGNING THE CORRECT ORGANIZATION AND LOCATION TO CAPSULE SERVER IN THE SATELLITE WEB UI

After installing Capsule Server packages, if there is more than one organization or location, you must assign the correct organization and location to Capsule to make Capsule visible in the Satellite web UI.

> **NOTE**
>
> Assigning a Capsule to the same location as your Satellite Server with an embedded Capsule prevents Red Hat Insights from uploading the Insights inventory. To enable the inventory upload, synchronize SSH keys for both Capsules.

Procedure

1. Log into the Satellite web UI.

2. From the **Organization** list in the upper-left of the screen, select **Any Organization**.

3. From the **Location** list in the upper-left of the screen, select **Any Location**.

4. In the Satellite web UI, navigate to **Hosts** > **All Hosts** and select Capsule Server.

5. From the **Select Actions** list, select **Assign Organization**.

6. From the **Organization** list, select the organization where you want to assign this Capsule.

7. Click **Fix Organization on Mismatch**.

8. Click **Submit**.

9. Select Capsule Server. From the **Select Actions** list, select **Assign Location**.

10. From the **Location** list, select the location where you want to assign this Capsule.

11. Click **Fix Location on Mismatch**.

12. Click **Submit**.

13. In the Satellite web UI, navigate to **Administer** > **Organizations** and click the organization to which you have assigned Capsule.

14. Click **Capsules** tab and ensure that Capsule Server is listed under the **Selected items** list, then click **Submit**.

15. In the Satellite web UI, navigate to **Administer** > **Locations** and click the location to which you have assigned Capsule.

16. Click **Capsules** tab and ensure that Capsule Server is listed under the **Selected items** list, then click **Submit**.

Verification

Optionally, you can verify if Capsule Server is correctly listed in the Satellite web UI.

1. Select the organization from the **Organization** list.

2. Select the location from the **Location** list.

3. In the Satellite web UI, navigate to **Hosts** > **All Hosts**.

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

CHAPTER 3. PERFORMING ADDITIONAL CONFIGURATION ON CAPSULE SERVER

After installation, you can configure additional settings on your Capsule Server.

3.1. CONFIGURING CAPSULE FOR HOST REGISTRATION AND PROVISIONING

Use this procedure to configure Capsule so that you can register and provision hosts using your Capsule Server instead of your Satellite Server.

Procedure

- On Satellite Server, add the Capsule to the list of trusted proxies.
  This is required for Satellite to recognize hosts' IP addresses forwarded over the **X-Forwarded-For** HTTP header set by Capsule. For security reasons, Satellite recognizes this HTTP header only from localhost by default. You can enter trusted proxies as valid IPv4 or IPv6 addresses of Capsules, or network ranges.

> ⚠️ **WARNING**
>
> Do not use a network range that is too wide, because that poses a potential security risk.

  Enter the following command. Note that the command overwrites the list that is currently stored in Satellite. Therefore, if you have set any trusted proxies previously, you must include them in the command as well:

  ```
  # satellite-installer \
  --foreman-trusted-proxies "127.0.0.1/8" \
  --foreman-trusted-proxies "::1" \
  --foreman-trusted-proxies "My_IP_address" \
  --foreman-trusted-proxies "My_IP_range"
  ```

  The localhost entries are required, do not omit them.

Verification

1. List the current trusted proxies using the full help of Satellite installer:

   ```
   # satellite-installer --full-help | grep -A 2 "trusted-proxies"
   ```

2. The current listing contains all trusted proxies you require.

3.2. CONFIGURING PULL-BASED TRANSPORT FOR REMOTE EXECUTION

By default, remote execution uses push-based SSH as the transport mechanism for the Script provider. If your infrastructure prohibits outgoing connections from Capsule Server to hosts, you can use remote execution with pull-based transport instead, because the host initiates the connection to Capsule Server. The use of pull-based transport is not limited to those infrastructures.

The pull-based transport comprises **pull-mqtt** mode on Capsules in combination with a pull client running on hosts.

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

The mode is configured per Capsule Server. Some Capsule Servers can be configured to use **pull-mqtt** mode while others use SSH. If this is the case, it is possible that one remote job on a given host will use the pull client and the next job on the same host will use SSH. If you wish to avoid this scenario, configure all Capsule Servers to use the same mode.

Procedure

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

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

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

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

3. Make the changes persistent:

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

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

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

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

Next steps

- Configure your hosts for the pull-based transport. For more information, see Transport modes for remote execution in *Managing hosts*.

3.3. ENABLING OPENSCAP ON CAPSULE SERVERS

On Satellite Server and the integrated Capsule of your Satellite Server, OpenSCAP is enabled by default. To use the OpenSCAP plugin and content on external Capsules, you must enable OpenSCAP on each Capsule.

Procedure

- To enable OpenSCAP, enter the following command:

```
# satellite-installer \
--enable-foreman-proxy-plugin-openscap \
--foreman-proxy-plugin-openscap-ansible-module true \
--foreman-proxy-plugin-openscap-puppet-module true
```

If you want to use Puppet to deploy compliance policies, you must enable it first. For more information, see *Managing configurations using Puppet integration* .

3.4. ADDING LIFECYCLE 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

1. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**, and select the Capsule that you want to add a lifecycle to.

2. Click **Edit** and click the **Lifecycle Environments** tab.

3. From the left menu, select the lifecycle 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 .

CLI procedure

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 to which you want to add a lifecycle.

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

```
# hammer capsule info \
--id My_capsule_ID
```

3. To view the lifecycle 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 My_capsule_ID
```

4. Add the lifecycle environment to your Capsule Server:

```
# hammer capsule content add-lifecycle-environment \
--id My_capsule_ID \
--lifecycle-environment-id My_Lifecycle_Environment_ID
--organization "My_Organization"
```

Repeat for each lifecycle environment you want to add to Capsule Server.

5. 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 My_capsule_ID
```

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

```
# hammer capsule content synchronize \
--id My_capsule_ID
--lifecycle-environment-id My_Lifecycle_Environment_ID
```

- To synchronize all content from your Satellite Server to your Capsule Server without checking metadata:

```
# hammer capsule content synchronize \
--id My_capsule_ID \
--skip-metadata-check true
```

This equals selecting **Complete Sync** in the Satellite web UI.

3.5. ENABLING POWER MANAGEMENT ON HOSTS

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

Prerequisites

- All 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 \
--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.

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

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

You can monitor the progress of the **satellite-installer** command displayed in your prompt. You can view the logs in **/var/log/foreman-installer/satellite.log**.

Additional resources

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

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

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

CHAPTER 4. CONFIGURING CAPSULE SERVER WITH EXTERNAL SERVICES

If you do not want to configure the DNS, DHCP, and TFTP services on Capsule Server, use this section to configure your 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 Server uses the **nsupdate** utility to update DNS records on the remote server.

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

Prerequisites

- You must have a configured external DNS server.

- This guide assumes you have an existing installation.

Procedure

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

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

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

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

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

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

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

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

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

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

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

4.2. CONFIGURING CAPSULE SERVER WITH EXTERNAL DHCP

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

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

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

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

Procedure

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

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

2. Generate a security token:

   ```
   # tsig-keygen -a hmac-md5 omapi_key
   ```

3. Edit the **dhcpd** configuration file for all subnets and add the key generated by **tsig-keygen**. The following is an example:

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

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

```
omapi-port 7911;
key omapi_key {
 algorithm hmac-md5;
 secret "My_Secret";
};
omapi-key omapi_key;
```

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

4. 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.

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

```
# firewall-cmd --add-service dhcp
```

6. Make the changes persistent:

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

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

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

8. 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
```

9. 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
```

10. Enable and start the DHCP service:

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

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

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

12. 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
```

13. 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
```

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

```
# mount -a
```

15. 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.

16. Reload the NFS server:

```
# exportfs -rva
```

17. Configure the firewall for DHCP omapi port 7911:

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

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

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

19. Make the changes persistent:

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

4.2.2. Configuring Satellite Server with an external DHCP server

You can configure Capsule Server with an external DHCP server.

Prerequisites

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

Procedure

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

   ```
   # satellite-maintain packages install nfs-utils
   ```

2. Create the DHCP directories for NFS:

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

3. Change the file owner:

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

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

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

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

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

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

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

   ```
   # mount -a
   ```

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

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

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

   ```
   # satellite-installer \
   --enable-foreman-proxy-plugin-dhcp-remote-isc \
   --foreman-proxy-dhcp-provider=remote_isc \
   --foreman-proxy-dhcp-server=My_DHCP_Server_FQDN \
   --foreman-proxy-dhcp=true \
   --foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
   ```

```
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-secret=My_Secret \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911
```

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

4.3. CONFIGURING CAPSULE SERVER WITH EXTERNAL TFTP

You can configure Capsule Server with external TFTP services.

Procedure

1. Create the TFTP directory for NFS:

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

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

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

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

   ```
   # mount -a
   ```

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

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

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

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

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

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

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

4.4. CONFIGURING CAPSULE SERVER WITH EXTERNAL IDM DNS

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

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

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

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

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

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

- Section 4.4.3, "Reverting to internal DNS service"

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

4.4.1. Configuring dynamic DNS update with GSS-TSIG authentication

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

Prerequisites

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

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

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

Procedure

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

Creating a Kerberos principal on the IdM server

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

   ```
   # kinit idm_user
   ```

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

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

Installing and configuring the idM client

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

   > # satellite-maintain packages install ipa-client

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

   > # ipa-client-install

3. Obtain a Kerberos ticket:

   > # kinit admin

4. Remove any preexisting **keytab**:

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

5. Obtain the **keytab** for this system:

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

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

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

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

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

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

Configuring DNS zones in the IdM web UI

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

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

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

   c. Click **Add and Edit**.

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

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

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

    f. Enable **Allow PTR sync**.

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

2. Create and configure the reverse zone:

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

    b. Click **Add**.

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

    d. Click **Add and Edit**.

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

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

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

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

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

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

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

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

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

    b. Configure the domain:

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

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

c. Configure the subnet:

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

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

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

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

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

4.4.2. Configuring dynamic DNS update with TSIG authentication

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

Prerequisites

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

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

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

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

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

Procedure

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

Enabling external updates to the DNS zone in the IdM server

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

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

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

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

```
# systemctl reload named
```

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

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

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

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

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

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

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

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

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

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

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

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

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

Testing external updates to the DNS zone in the IdM server

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

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

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

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

3. On Satellite Server, test the DNS entry:

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

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

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

5. If resolved successfully, remove the test DNS entry:

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

6. Confirm that the DNS entry was removed:

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

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

4.4.3. Reverting to internal DNS service

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

Procedure

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

Configuring Satellite or Capsule as a DNS server

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

  ```
  # satellite-installer
  ```

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

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

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

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

Updating the configuration in the Satellite web UI

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

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

3. Configure the domain:

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

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

4. Configure the subnet:

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

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

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

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

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

CHAPTER 5. MANAGING DHCP USING CAPSULE

Satellite can integrate with a DHCP service using your Capsule. A Capsule has multiple DHCP providers that you can use to integrate Satellite with your existing DHCP infrastructure or deploy a new one. You can use the DHCP module of Capsule to query for available IP addresses, add new, and delete existing reservations. Note that your Capsule cannot manage subnet declarations.

Available DHCP providers

- **dhcp_infoblox** – For more information, see Using Infoblox as DHCP and DNS Providers in *Provisioning hosts*.

- **dhcp_isc** – ISC DHCP server over OMAPI. For more information, see Configuring DNS, DHCP, and TFTP on Capsule Server in *Installing Capsule Server*.

- **dhcp_remote_isc** – ISC DHCP server over OMAPI with leases mounted through networking. For more information, see Configuring an External DHCP Server to Use with Capsule Server in *Installing Capsule Server*.

5.1. SECURING THE DHCPD API

Capsule interacts with DHCP daemon using the dhcpd API to manage DHCP. By default, the dhcpd API listens to any host without access control. You can add an **omapi_key** to provide basic security.

Procedure

1. On your Capsule, install the required packages:

   ```
   # satellite-maintain packages install bind-utils
   ```

2. Generate a key:

   ```
   # dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
   # cat Komapi_key.+*.private | grep ^Key|cut -d ' ' -f2-
   ```

3. Use **satellite-installer** to secure the dhcpd API:

   ```
   # satellite-installer \
   --foreman-proxy-dhcp-key-name "My_Name" \
   --foreman-proxy-dhcp-key-secret "My_Secret"
   ```

CHAPTER 6. MANAGING DNS USING CAPSULE

Satellite can manage DNS records using your Capsule. DNS management contains updating and removing DNS records from existing DNS zones. A Capsule has multiple DNS providers that you can use to integrate Satellite with your existing DNS infrastructure or deploy a new one.

After you have enabled DNS, your Capsule can manipulate any DNS server that complies with RFC 2136 using the **dns_nsupdate** provider. Other providers provide more direct integration, such as **dns_infoblox** for Infoblox.

Available DNS providers

- **dns_infoblox** – For more information, see Using Infoblox as DHCP and DNS Providers in *Provisioning hosts*.

- **dns_nsupdate** – Dynamic DNS update using nsupdate. For more information, see Using Infoblox as DHCP and DNS Providers in *Provisioning hosts*.

- **dns_nsupdate_gss** – Dynamic DNS update with GSS-TSIG. For more information, see Section 4.4.1, "Configuring dynamic DNS update with GSS-TSIG authentication".

APPENDIX A. CAPSULE SERVER SCALABILITY CONSIDERATIONS

The maximum number of Capsule Servers that Satellite Server can support has no fixed limit. It was tested that a Satellite Server can support 17 Capsule Servers with 2 vCPUs. 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. 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 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 overloading Satellite Server. 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

APPENDIX B. TROUBLESHOOTING DNF MODULES

If DNF modules fails to enable, it can mean an incorrect module is enabled. In that case, you have to resolve dependencies manually as follows. List the enabled modules:

```
# dnf module list --enabled
```

B.1. RUBY

If Ruby module fails to enable, it can mean an incorrect module is enabled. In that case, you have to resolve dependencies manually as follows:

List the enabled modules:

```
# dnf module list --enabled
```

If the Ruby 2.5 module has already been enabled, perform a module reset:

```
# dnf module reset ruby
```

B.2. POSTGRESQL

If PostgreSQL module fails to enable, it can mean an incorrect module is enabled. In that case, you have to resolve dependencies manually as follows:

List the enabled modules:

```
# dnf module list --enabled
```

If the PostgreSQL 10 module has already been enabled, perform a module reset:

```
# dnf module reset postgresql
```

If a database was previously created using PostgreSQL 10, perform an upgrade:

1. Enable the DNF modules:

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

2. Install the PostgreSQL upgrade package:

   ```
   # dnf install postgresql-upgrade
   ```

3. Perform the upgrade:

   ```
   # postgresql-setup --upgrade
   ```