Red Hat Satellite 6.12

Installing Satellite Server in a Connected Network Environment

Install Red Hat Satellite Server that is deployed inside a network connected to the Internet

Last Updated: 2023-01-24

Red Hat Satellite 6.12 Installing Satellite Server in a Connected Network Environment

Install Red Hat Satellite Server that is deployed inside a network connected to the Internet

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

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. PREPARING YOUR ENVIRONMENT FOR INSTALLATION

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

1.1. SYSTEM REQUIREMENTS

The following requirements apply to the networked base operating system:

- x86_64 architecture
- 4-core 2.0 GHz CPU at a minimum
- A minimum of 20 GB RAM is required for Satellite Server to function. In addition, a minimum of 4 GB RAM of swap space is also recommended. Satellite running with less RAM than the minimum value might not operate correctly.
- A supported operating system installed with all available updates applied
- 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

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 the system locale in *Red Hat Enterprise Linux 8 Configuring basic system settings*.

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

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

- apache
- foreman
- foreman-proxy
- postgres
- pulp
- puppet

- qdrouterd
- qpidd
- tomcat

Certified hypervisors

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

SELinux Mode

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

FIPS Mode

You can install Satellite Server on a Red Hat Enterprise Linux system that is operating in FIPS mode. For more information, see Installing a RHEL 8 system with FIPS mode enabled in *Red Hat Enterprise Linux Security hardening*.

NOTE

Satellite supports DEFAULT and FIPS crypto-policies. The FUTURE crypto-policy is not supported for Satellite and Capsule installations.

Inter-Satellite Synchronization (ISS)

In a scenario with air-gapped Satellite Servers, all your Satellite Servers must be on the same Satellite version for ISS Export Sync to work. ISS Network Sync works across all Satellite versions that support it. For more information, see Synchronizing Content Between Satellite Servers in *Managing Content*.

1.2. STORAGE REQUIREMENTS

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

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

Table 1.1. Storage Requirements for a Satellite Server Installation

Directory	Installation Size	Runtime Size
/var/log	10 MB	10 GB
/var/lib/pgsql	100 MB	20 GB
/usr	5 GB	Not Applicable
/opt/puppetlabs	500 MB	Not Applicable
/var/lib/pulp	1 MB	300 GB

Directory	Installation Size	Runtime Size
/var/lib/qpidd	25 MB	Refer Storage Guidelines

For external database servers: **/var/lib/pgsql** with installation size of 100 MB and runtime size of 20 GB.

For detailed information on partitioning and size, see Partitioning reference in the *Red Hat Enterprise Linux 8 System Design Guide*.

1.3. STORAGE GUIDELINES

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

- If you mount the **/tmp** directory as a separate file system, you must use the **exec** mount option in the **/etc/fstab** file. If **/tmp** is already mounted with the **noexec** option, you must change the option to **exec** and re-mount the file system. This is a requirement for the **puppetserver** service to work.
- Because most Satellite Server data is stored in the **/var** directory, mounting **/var** on LVM storage can help the system to scale.
- The **/var/lib/qpidd/** directory uses slightly more than 2 MB per Content Host managed by the **goferd** service. For example, 10 000 Content Hosts require 20 GB of disk space in **/var/lib/qpidd/**.
- Use high-bandwidth, low-latency storage for the **/var/lib/pulp/** 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

Satellite Server is supported on the latest versions of Red Hat Enterprise Linux 8 that are available at the time when Satellite Server is installed. Previous versions of Red Hat Enterprise Linux including EUS or z-stream are not supported.

You can install the operating system from a disc, local ISO image, kickstart, or any other method that Red Hat supports.

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	

Before you install Satellite, apply all operating system updates if possible.

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

Install Satellite Server on a freshly provisioned system.

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

1.5. SUPPORTED BROWSERS

Satellite supports recent versions of Firefox and Google Chrome browsers.

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

1.6. PORTS AND FIREWALLS REQUIREMENTS

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

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

Integrated Capsule

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

Clients of Capsule

Hosts which are clients of Capsules, other than Satellite's integrated Capsule, do not need access to Satellite Server. For more information on Satellite Topology and an illustration of port connections, see Capsule Networking in *Satellite 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. Satellite Server 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	TCP	HTTPS	Capsule	Red Hat Satellite API	Communication from Capsule
443, 80	TCP	HTTPS, HTTP	Client	Content Retrieval	Content
443, 80	TCP	HTTPS, HTTP	Capsule	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
1883	TCP	MQTT	Client	Pull based REX (optional)	Content hosts for REX job notification (optional)
5646, 5647	TCP	AMQP	Capsule	Katello agent	Forward message to Qpid dispatch router on Satellite (optional)
5910 – 5930	TCP	HTTPS	Browsers	Compute Resource's virtual console	
8000	TCP	HTTP	Client	Provisioning templates	Template retrieval for client installers, iPXE or UEFI HTTP Boot
8000	TCP	HTTPS	Client	PXE Boot	Installation
8140	TCP	HTTPS	Client	Puppet agent	Client updates (optional)
8443	TCP	HTTPS	Client	Content Host registration	Initiation Uploading facts Sending installed packages and traces
9090	TCP	HTTPS	Client	OpenSCAP	Configure Client
9090	TCP	HTTPS	Discovered Node	Discovery	Host discovery and provisioning
9090	TCP	HTTPS	Red Hat Satellite	Capsule API	Capsule functionality

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

A DHCP Capsule performs ICMP ping or 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. Satellite Server 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
22, 16514	TCP	SSH SSH/TLS	Compute Resource	Satellite originated communications, for compute resources in libvirt	
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)
53	TCP and UDP	DNS	DNS Server	Orchestration	Validation of DNS conflicts
68	UDP	DHCP	Client	Dynamic IP	DHCP (optional)
80	TCP	HTTP	Remote repository	Content Sync	Remote yum repository

Destination Port	Protocol	Service	Destination	Required For	Description
389, 636	TCP	LDAP, LDAPS	External LDAP Server	LDAP	LDAP authentication, necessary only if external authentication is enabled. The port can be customized when **LDAPAuthSource** is defined
443	TCP	HTTPS	Satellite	Capsule	Capsule Configuration management Template retrieval OpenSCAP Remote Execution result upload
443	TCP	HTTPS	Amazon EC2, Azure, Google GCE	Compute resources	Virtual machine interactions (query/create/destroy) (optional)
443	TCP	HTTPS	console.redhat.com	Red Hat Cloud plugin API calls	
443	TCP	HTTPS	Red Hat Portal	SOS report	Assisting support cases (optional)
443	TCP	HTTPS	Red Hat CDN	Content Sync	Red Hat CDN
443	TCP	HTTPS	cert-api.access.redhat.com	Telemetry data upload and report	
443	TCP	HTTPS	Capsule	Content mirroring	Initiation
443	TCP	HTTPS	Infoblox DHCP Server	DHCP management	When using Infoblox for DHCP, management of the DHCP leases (optional)

Destination Port	Protocol	Service	Destination	Required For	Description
623			Client	Power management	BMC On/Off/Cycle/Status
5000	TCP	HTTPS	OpenStack Compute Resource	Compute resources	Virtual machine interactions (query/create/destroy) (optional)
5646	TCP	AMQP	Satellite Server	Katello agent	Forward message to Qpid dispatch router on Capsule (optional)
5671			Qpid	Remote install	Send install command to client
5671			Dispatch router (hub)	Remote install	Forward message to dispatch router on Satellite
5671			Satellite Server	Remote install for Katello agent	Send install command to client
5671			Satellite Server	Remote install for Katello agent	Forward message to dispatch router on Satellite
5900 – 5930	TCP	SSL/TLS	Hypervisor	noVNC console	Launch noVNC console
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

Destination Port	Protocol	Service	Destination	Required For	Description
8443	TCP	HTTPS	Client	Discovery	Capsule sends reboot command to the discovered host (optional)
9090	TCP	HTTPS	Capsule	Capsule API	Management of Capsules

1.7. ENABLING CONNECTIONS FROM A CLIENT TO SATELLITE SERVER

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

Use this procedure to configure the host-based firewall on the system that Satellite is installed on, to enable incoming connections from Clients, and to make the configuration persistent across system reboots. For more information on the ports used, see Ports and Firewalls Requirements.

Procedure

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

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

2. Make the changes persistent:

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

1.8. VERIFYING FIREWALL SETTINGS

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

Procedure

1. Enter the following command:

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

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

1.9. VERIFYING DNS RESOLUTION

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

Procedure

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

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

Successful name resolution results in output similar to the following:

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

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

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

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

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

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

For more information, see the Changing a hostname using hostnamectl in the *Red Hat Enterprise Linux 8 Configuring and managing networking*.

WARNING

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

CHAPTER 2. PREPARING YOUR ENVIRONMENT FOR SATELLITE INSTALLATION IN AN IPV6 NETWORK

You can install and use Satellite in an IPv6 network. Before installing Satellite in an IPv6 network, view the limitations and ensure that you meet the requirements.

To provision hosts in an IPv6 network, after installing Satellite, you must also configure Satellite for the UEFI HTTP boot provisioning. For more information, see Section 4.5, “Configuring Satellite for UEFI HTTP Boot Provisioning in an IPv6 Network”.

2.1. LIMITATIONS OF SATELLITE INSTALLATION IN AN IPV6 NETWORK

Satellite installation in an IPv6 network has the following limitations:

- You can install Satellite and Capsules in IPv6-only systems, dual-stack installation is not supported.
- Although Satellite provisioning templates include IPv6 support for PXE and HTTP (iPXE) provisioning, the only tested and certified provisioning workflow is the UEFI HTTP Boot provisioning. This limitation only relates to users who plan to use Satellite to provision hosts.

2.2. REQUIREMENTS FOR SATELLITE INSTALLATION IN AN IPV6 NETWORK

Before installing Satellite in an IPv6 network, ensure that you meet the following requirements:

- You must deploy an external DHCP IPv6 server as a separate unmanaged service to bootstrap clients into GRUB2, which then configures IPv6 networking either using DHCPv6 or or assigning static IPv6 address. This is required because the DHCP server in Red Hat Enterprise Linux (ISC DHCP) does not provide an integration API for managing IPv6 records, therefore the Capsule DHCP plug-in that provides DHCP management is limited to IPv4 subnets.
- You must deploy an external IPv4 HTTP proxy server. This is required because Red Hat Content Delivery Network distributes content only over IPv4 networks, therefore you must use this proxy to pull content into the Satellite on your IPv6 network.
- You must configure Satellite to use this IPv4 HTTP proxy server as the default proxy. For more information, see Adding a Default HTTP Proxy to Satellite .

CHAPTER 3. INSTALLING SATELLITE SERVER

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

NOTE

You cannot register Satellite Server to itself.

Use the following procedures to install Satellite Server, perform the initial configuration, and import subscription manifests. For more information on subscription manifests, see Managing Red Hat Subscriptions in *Managing Content*.

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

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

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

You can restore the previous file as follows:

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

3.1. CONFIGURING THE HTTP PROXY TO CONNECT TO RED HAT CDN

Prerequisites

Your network gateway and the HTTP proxy must allow access to the following hosts:

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

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

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

To configure the subscription-manager with the HTTP proxy, follow the procedure below.

Procedure

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

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

# port for http proxy server
proxy_port = 8080

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

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

3.2. REGISTERING TO RED HAT SUBSCRIPTION MANAGEMENT

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

Procedure

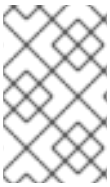
- Register your system with the Red Hat Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

The command displays output similar to the following:

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

3.3. ATTACHING THE SATELLITE INFRASTRUCTURE SUBSCRIPTION



NOTE

This step is **optional** if you have SCA enabled in the Red Hat Customer Portal. There is no requirement of attaching the Red Hat Satellite Infrastructure Subscription to the Satellite Server using subscription-manager.

After you have registered Satellite Server, you must identify your subscription Pool ID and attach an available subscription. The Red Hat Satellite Infrastructure subscription provides access to the Red Hat Satellite and Red Hat Enterprise Linux content. For Red Hat Enterprise Linux 7, it also provides access to Red Hat Software Collections (RHSCL). This is the only subscription required.

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

Subscriptions are classified as available if they are not already attached to a system. If you are unable to find an available Satellite subscription, see the Red Hat Knowledgebase solution How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager? to run a script to see if your subscription is being consumed by another system.

Procedure

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

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

The command displays output similar to the following:

```
Subscription Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Satellite
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Satellite Capsule
                     Red Hat Ansible Engine
                     Red Hat Satellite with Embedded Oracle
                     Red Hat Satellite 5 Managed DB
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat Beta
                     Red Hat Software Collections Beta (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Satellite Proxy
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Discovery
SKU:                 MCT3718
Contract:
Pool ID:             8aca43dd771bf31101771c0231f906a5
Provides Management: Yes
Available:           10
Suggested:           1
Service Type:        L1-L3
Roles:
Service Level:       Premium
Usage:
Add-ons:
Subscription Type:   Standard
Starts:              11/11/2020
Ends:                11/11/2023
Entitlement Type:    Physical
```

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

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

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

The command displays output similar to the following:

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

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

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

3.4. CONFIGURING REPOSITORIES

Use this procedure to enable the repositories that are required to install Satellite 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-6.12-for-rhel-8-x86_64-rpms \
--enable=satellite-maintenance-6.12-for-rhel-8-x86_64-rpms
```

3. Enable the module:

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

NOTE

Enablement of the module **satellite:el8** warns about a conflict with **postgresql:10** and **ruby:2.5** as these modules are set to the default module versions on Red Hat Enterprise Linux 8. The module **satellite:el8** has a dependency for the modules **postgresql:12** and **ruby:2.7** that will be enabled with the **satellite:el8** module. These warnings do not cause installation process failure, hence can be ignored safely. For more information about modules and lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux Application Streams Life Cycle.

3.5. INSTALLING SATELLITE SERVER PACKAGES

Procedure

1. Update all packages:

```
# dnf update
```

2. Install Satellite Server packages:

```
# dnf install satellite
```

3.6. SYNCHRONIZING THE SYSTEM CLOCK WITH CHRONYD

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

For more information about the **chrony** suite, see Using the Chrony suite to configure NTP in *Red Hat Enterprise Linux 8 Configuring basic system settings*.

Procedure

1. Install the **chrony** package:

```
# dnf install chrony
```

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

```
# systemctl enable --now chronyd
```

3.7. INSTALLING THE SOS PACKAGE ON THE BASE OPERATING SYSTEM

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

Procedure

1. Install the **sos** package:

```
# dnf install sos
```

3.8. CONFIGURING SATELLITE SERVER

Install Satellite Server using the **satellite-installer** installation script.

This method is performed by running the installation script with one or more command options. The command options override the corresponding default initial configuration options and are recorded in the Satellite answer file. You can run the script as often as needed to configure any necessary options.

NOTE

Depending on the options that you use when running the Satellite installer, the configuration can take several minutes to complete.

3.8.1. Configuring Satellite Installation

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

The installation process can take tens of minutes to complete. If you are connecting remotely to the system, use a utility such as **tmux** that allows suspending and reattaching a communication session so that you can check the installation progress in case you become disconnected from the remote system. If you lose connection to the shell where the installation command is running, see the log at **/var/log/foreman-installer/satellite.log** to determine if the process completed successfully.

Considerations

- Use the **satellite-installer --scenario satellite --help** command to display the available options and any default values. If you do not specify any values, the default values are used.
- Specify a meaningful value for the option: **--foreman-initial-organization**. This can be your company name. An internal label that matches the value is also created and cannot be changed afterwards. If you do not specify a value, an organization called **Default Organization** with the label **Default_Organization** is created. You can rename the organization name but not the label.
- Remote Execution is the primary method of managing packages on Content Hosts. If you want to use the deprecated Katello Agent instead of Remote Execution SSH, use the **--foreman-proxy-content-enable-katello-agent=true** option to enable it. The same option should be given on any Capsule Server as well as Satellite Server.
- By default, all configuration files configured by the installer are managed by Puppet. When **satellite-installer** runs, it overwrites any manual changes to the Puppet managed files with the initial values. By default, Satellite Server is installed with the Puppet agent running as a service. If required, you can disable Puppet agent on Satellite Server using the **--puppet-runmode=none** option.
- If you want to manage DNS files and DHCP files manually, use the **--foreman-proxy-dns-managed=false** and **--foreman-proxy-dhcp-managed=false** options so that Puppet does not manage the files related to the respective services. For more information on how to apply custom configuration on other services, see Applying Custom Configuration to Satellite .

Procedure

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

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

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

3.9. IMPORTING A RED HAT SUBSCRIPTION MANIFEST INTO SATELLITE SERVER

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

Prerequisites

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

Procedure

1. In the Satellite web UI, ensure the context is set to the organization you want to use.
2. In the Satellite web UI, navigate to **Content** > **Subscriptions** and click **Manage Manifest**.
3. In the Manage Manifest window, click **Browse**.
4. Navigate to the location that contains the Red Hat Subscription Manifest file, then click **Open**. If the Manage Manifest window does not close automatically, click **Close** to return to the Subscriptions window.

CLI procedure

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

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

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

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

You can now enable repositories and import Red Hat content. For more information, see Importing Content in *Managing Content*.

CHAPTER 4. PERFORMING ADDITIONAL CONFIGURATION ON SATELLITE SERVER

4.1. USING RED HAT INSIGHTS WITH SATELLITE SERVER

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

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

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

Scheduling insights-client

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

Procedure

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

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

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

```
# satellite-installer --register-with-insights
```

4.2. DISABLING REGISTRATION TO RED HAT INSIGHTS

After you install or upgrade Satellite, you can choose to unregister or register Red Hat Insights as needed. For example, if you need to use Satellite in a disconnected environment, you can unregister **insights-client** from Satellite Server.

Prerequisites

1. You have registered Satellite to Red Hat Customer Portal.

Procedure

1. Optional: To unregister Red Hat Insights from Satellite Server, enter the following command:

```
# insights-client --unregister
```

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

```
# satellite-installer --register-with-insights
```

4.3. ENABLING THE SATELLITE CLIENT 6 REPOSITORY

The Satellite Client 6 repository provides the **katello-agent**, **katello-host-tools**, and **puppet** packages for clients registered to Satellite Server.

To use the CLI instead of the Satellite web UI, see the CLI procedure.

Procedure

1. Use the Search field to enter the following repository name: **Satellite Client 6 (for RHEL 8) (RPMs)**.
2. In the Available Repositories pane, click on **Satellite Client 6 (for RHEL 8) (RPMs)**to expand the repository set.
   If the **Satellite Client 6** items are not visible, it may be because they are not included in the Red Hat Subscription Manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the Red Hat Subscription Manifest and import it into Satellite. For more information, see Managing Red Hat Subscriptions in *Managing Content*.
3. For the **x86_64** entry, click the **Enable** icon to enable the repository.

Enable the Satellite Client 6 repository for every supported major version of Red Hat Enterprise Linux running on your hosts. After enabling a Red Hat repository, a Product for this repository is automatically created.

CLI procedure

- Enable the Satellite Client 6 repository using the **hammer repository-set enable** command:

```
# hammer repository-set enable \
--basearch='x86_64' \
--name 'Red Hat Satellite Client 6 for RHEL 8 x86_64 (RPMs)' \
--organization "My_Organization" \
--product 'Red Hat Enterprise Linux for x86_64'
```

4.4. SYNCHRONIZING THE SATELLITE CLIENT 6 REPOSITORY

Use this section to synchronize the Satellite Client 6 repository from the Red Hat Content Delivery Network (CDN) to your Satellite. This repository provides the **katello-agent**, **katello-host-tools**, and **puppet** packages for clients registered to Satellite Server.

Procedure

1. In the Satellite web UI, navigate to **Content** > **Sync Status**.
   A list of product repositories available for synchronization is displayed.
2. Click the arrow next to the **Red Hat Enterprise Linux Server** product to view available content.
3. Select **Satellite Client 6 (for RHEL 8) RPMs x86_64**
4. Click **Synchronize Now**.

CLI procedure

- Synchronize your Satellite Client 6 repository using the **hammer repository synchronize** command:

```
# hammer repository synchronize \
--name 'Red Hat Satellite Client 6 for RHEL 8 x86_64 RPMs' \
--organization "My_Organization" \
--product 'Red Hat Enterprise Linux for x86_64'
```

4.5. CONFIGURING SATELLITE FOR UEFI HTTP BOOT PROVISIONING IN AN IPV6 NETWORK

Use this procedure to configure Satellite to provision hosts in an IPv6 network with UEFI HTTP Boot provisioning.

Prerequisites

- Ensure that your clients can access DHCP and HTTP servers.
- Ensure that the UDP ports 67 and 68 are accessible by clients so clients can send DHCP requests and receive DHCP offers.
- Ensure that the TCP port 8000 is open for clients to download files and Kickstart templates from Satellite and Capsules.
- Ensure that the host provisioning interface subnet has an HTTP Boot Capsule, and Templates Capsule set. For more information, see Adding a Subnet to Satellite Server in *Provisioning Hosts*.
- In the Satellite web UI, navigate to **Administer** > **Settings** > **Provisioning** and ensure that the **Token duration** setting is not set to **0**. Satellite cannot identify clients that are booting from the network by a remote IPv6 address because of unmanaged DHCPv6 service, therefore provisioning tokens must be enabled.

Procedure

1. You must disable DHCP management in the installer or not use it.
2. For all IPv6 subnets created in Satellite, set the **DHCP Capsule** to blank.
3. Optional: If the host and the DHCP server are separated by a router, configure the DHCP relay agent and point to the DHCP server.
4. On Satellite or Capsule from which you provision, update the **grub2-efi** package to the latest version:

```
# satellite-maintain packages update grub2-efi
```

5. Synchronize the Red Hat Enterprise Linux 8 kickstart repository.

4.6. CONFIGURING SATELLITE SERVER WITH AN HTTP PROXY

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

4.6.1. Adding a Default HTTP Proxy to Satellite

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

The following procedure configures a proxy only for downloading content for Satellite. To use the CLI instead of the Satellite web UI, see the CLI procedure.

Procedure

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

CLI procedure

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

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

2. Add an HTTP proxy entry to Satellite:

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

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

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

4.6.2. Configuring SELinux to Ensure Access to Satellite on Custom Ports

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

Procedure

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

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

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

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

4.6.3. Using an HTTP Proxy for all Satellite HTTP Requests

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

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

Procedure

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

CLI procedure

- Enter the following command:

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

4.6.4. Excluding Hosts from Receiving Proxied Requests

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

Procedure

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

3. Click the tick icon to save your changes.

CLI procedure

- Enter the following command:

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

4.6.5. Resetting the HTTP Proxy

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

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **Content** tab.
2. Set the **Default HTTP Proxy** setting to **no global default**.

CLI procedure

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

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

4.7. ENABLING POWER MANAGEMENT ON MANAGED HOSTS

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

Prerequisites

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

Procedure

- To enable BMC, enter the following command:

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

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

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

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

To use external DNS, DHCP, and TFTP services instead, see Chapter 6, *Configuring Satellite Server with External Services*.

Adding Multihomed DHCP details

If you want to use Multihomed DHCP, you must inform the installer.

Prerequisites

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

Procedure

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

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns true \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-reverse 2.0.192.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-additional-interfaces eth1 \
--foreman-proxy-dhcp-additional-interfaces eth2 \
--foreman-proxy-dhcp-range "192.0.2.100 192.0.2.150" \
--foreman-proxy-dhcp-gateway 192.0.2.1 \
--foreman-proxy-dhcp-nameservers 192.0.2.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername 192.0.2.3
```

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

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

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

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

Procedure

1. On Satellite Server, enter the following command:

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

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

```
Option 66: IP address of Satellite or Capsule
Option 67: /pxelinux.0
```

   For more information about DHCP options, see RFC 2132.

NOTE

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

4.10. CONFIGURING SATELLITE SERVER FOR OUTGOING EMAILS

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

Prerequisite

- Some SMTP servers with anti-spam protection or grey-listing features are known to cause problems. To setup outgoing email with such a service either install and configure a vanilla SMTP service on Satellite Server for relay or use the **sendmail** command instead.

Procedure

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

2. Click the **Email** tab and set the configuration options to match your preferred delivery method. The changes have an immediate effect.

   a. The following example shows the configuration options for using an SMTP server:

   Table 4.1. Using an SMTP server as a delivery method

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

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

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

   Table 4.2. Using gmail.com as an SMTP server

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

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

Table 4.3. Using sendmail as a delivery method

Name	Example value
Delivery method	Sendmail
Sendmail location	/usr/sbin/sendmail
Sendmail arguments	-i

For security reasons, both Sendmail location and Sendmail argument settings are read-only and can be only set in **/etc/foreman/settings.yaml**. Both settings currently cannot be set via **satellite-installer**. For more information see the **sendmail 1** man page.

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

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

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

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

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

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

NOTE

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

4.11. CONFIGURING AN ALTERNATE CNAME FOR SATELLITE

You can configure an alternate CNAME for Satellite. This might be useful if you want to deploy the Satellite web interface on a different domain name than the one that is used by client systems to connect to Satellite. You must plan the alternate CNAME configuration in advance prior to installing Capsules and registering hosts to Satellite to avoid redeploying new certificates to hosts.

4.11.1. Configuring Satellite with an Alternate CNAME

Use this procedure to configure Satellite with an alternate CNAME. Note that the procedures for users of a default Satellite certificate and custom certificate differ.

For Default Satellite Certificate Users

- If you have installed Satellite with a default Satellite certificate and want to configure Satellite with an alternate CNAME, enter the following command on Satellite to generate a new default Satellite SSL certificate with an additional CNAME.

```
# satellite-installer --certs-cname alternate_fqdn --certs-update-server
```

- If you have not installed Satellite, you can add the **--certs-cname *alternate_fqdn*** option to the **satellite-installer** command to install Satellite with an alternate CNAME.

For Custom Certificate Users

If you use Satellite with a custom certificate, when creating a custom certificate, include the alternate CNAME records to the custom certificate. For more information, see Creating a Custom SSL Certificate for Satellite Server.

4.11.2. Configuring Hosts to Use an Alternate Satellite CNAME for Content Management

If Satellite is configured with an alternate CNAME, you can configure hosts to use the alternate Satellite CNAME for content management. To do this, you must point hosts to the alternate Satellite CNAME prior to registering the hosts to Satellite. You can do this using the bootstrap script or manually.

Configuring Hosts with the bootstrap Script

On the host, run the bootstrap script with the **--server *alternate_fqdn.example.com*** option to register the host to the alternate Satellite CNAME:

```
# ./bootstrap.py --server alternate_fqdn.example.com
```

Configuring Hosts Manually

On the host, edit the **/etc/rhsm/rhsm.conf** file to update **hostname** and **baseurl** settings to point to the alternate host name, for example:

```
[server]
# Server hostname:
hostname = alternate_fqdn.example.com

content omitted

[rhsm]
# Content base URL:
baseurl=https://alternate_fqdn.example.com/pulp/content/
```

Now you can register the host with the **subscription-manager**.

4.12. CONFIGURING SATELLITE SERVER WITH A CUSTOM SSL CERTIFICATE

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

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

1. Section 4.12.1, “Creating a Custom SSL Certificate for Satellite Server”
2. Section 4.12.2, “Deploying a Custom SSL Certificate to Satellite Server”
3. Section 4.12.3, “Deploying a Custom SSL Certificate to Hosts”
4. If you have external Capsule Servers registered to Satellite Server, you must configure them with custom SSL certificates. The same Certificate Authority must sign certificates for Satellite Server and Capsule Server. For more information, see Configuring Capsule Server with a Custom SSL Certificate in *Installing Capsule Server*.

4.12.1. Creating a Custom SSL Certificate for Satellite Server

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

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

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

Procedure

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

```
# mkdir /root/satellite_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 Satellite Server, skip this step.

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

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

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

[ req_distinguished_name ] (1)
C  = Country Name (2 letter code)
ST = State or Province Name (full name)
L  = Locality Name (eg, city)
O  = Organization Name (eg, company)
OU = The division of your organization handling the certificate
CN = satellite.example.com (2)
```

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

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

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

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

4. Generate the Certificate Signing Request (CSR):

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

1. Path to the private key.
2. Path to the configuration file.
3. Path to the CSR to generate.

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

4.12.2. Deploying a Custom SSL Certificate to Satellite Server

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

Procedure

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

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

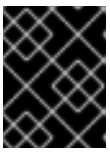
1. Path to Satellite Server certificate file that is signed by a Certificate Authority.
2. Path to the private key that was used to sign Satellite Server certificate.
3. Path to the Certificate Authority bundle.

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

Example output of katello-certs-check

```
Validation succeeded.

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

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

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

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

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



IMPORTANT

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

3. On a computer with network access to Satellite Server, navigate to the following URL: **https://satellite.example.com**.

4. In your browser, view the certificate details to verify the deployed certificate.

4.12.3. Deploying a Custom SSL Certificate to Hosts

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

Procedure

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

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

4.13. USING EXTERNAL DATABASES WITH SATELLITE

As part of the installation process for Red Hat Satellite, the **satellite-installer** command installs PostgreSQL databases on the same server as Satellite. In certain Satellite deployments, using external databases instead of the default local databases can help with the server load.

Red Hat does not provide support or tools for external database maintenance. This includes backups, upgrades, and database tuning. You must have your own database administrator to support and maintain external databases.

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

1. Section 4.13.2, “Preparing a Host for External Databases” . Prepare a Red Hat Enterprise Linux 8 server to host the external databases.
2. Section 4.13.3, “Installing PostgreSQL”. Prepare PostgreSQL with databases for Satellite, Candlepin and Pulp with dedicated users owning them.
3. Section 4.13.4, “Configuring Satellite Server to use External Databases” . Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**.

4.13.1. PostgreSQL as an External Database Considerations

Foreman, Katello, and Candlepin use the PostgreSQL database. If you want to use PostgreSQL as an external database, the following information can help you decide if this option is right for your Satellite configuration. Satellite supports PostgreSQL version 12.

Advantages of External PostgreSQL

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

Disadvantages of External PostgreSQL

- Increase in deployment complexity that can make troubleshooting more difficult

- The external PostgreSQL server is an additional system to patch and maintain
- If either Satellite or the PostgreSQL database server suffers a hardware or storage failure, Satellite is not operational
- If there is latency between the Satellite server and database server, performance can suffer

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

4.13.2. Preparing a Host for External Databases

Install a freshly provisioned system with the latest Red Hat Enterprise Linux 8 to host the external databases.

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

Prerequisite

- The prepared host must meet Satellite's Storage Requirements.

Procedure

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

```
# subscription-manager repos --disable '*'
# subscription-manager repos \
--enable=satellite-6.12-for-rhel-8-x86_64-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms
```

3. Enable the following modules:

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

NOTE

Enablement of the module **satellite:el8** warns about a conflict with **postgresql:10** and **ruby:2.5** as these modules are set to the default module versions on Red Hat Enterprise Linux 8. The module **satellite:el8** has a dependency for the modules **postgresql:12** and **ruby:2.7** that will be enabled with the **satellite:el8** module. These warnings do not cause installation process failure, hence can be ignored safely. For more information about modules and lifecycle streams on Red Hat Enterprise Linux 8, see Red Hat Enterprise Linux Application Streams Life Cycle.

4.13.3. Installing PostgreSQL

You can install only the same version of PostgreSQL that is installed with the **satellite-installer** tool during an internal database installation. Satellite supports PostgreSQL version 12.

Procedure

1. To install PostgreSQL, enter the following command:

```
# dnf install postgresql-server postgresql-evr
```

2. To initialize PostgreSQL, enter the following command:

```
# postgresql-setup initdb
```

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

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

   Note that the default configuration of external PostgreSQL needs to be adjusted to work with Satellite. The base recommended external database configuration adjustments are as follows:

   - checkpoint_completion_target: 0.9
   - max_connections: 500
   - shared_buffers: 512MB
   - work_mem: 4MB

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

```
listen_addresses = '*'
```

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

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

6. Add the following line to the file:

```
host  all   all   Satellite_ip/32   md5
```

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

```
# systemctl enable --now postgresql
```

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

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

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

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

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

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

11. Exit the **postgres** user:

```
# \q
```

12. From Satellite Server, test that you can access the database. If the connection succeeds, the commands return **1**.

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com  -p 5432 -U foreman
-d foreman -c "SELECT 1 as ping"
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U
candlepin -d candlepin -c "SELECT 1 as ping"
# PGPASSWORD='Pulpcore_Password' psql -h postgres.example.com -p 5432 -U pulp -d
pulpcore -c "SELECT 1 as ping"
```

4.13.4. Configuring Satellite Server to use External Databases

Use the **satellite-installer** command to configure Satellite to connect to an external PostgreSQL database.

Prerequisite

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

Procedure

1. To configure the external databases for Satellite, enter the following command:

```
satellite-installer --scenario satellite \
  --foreman-db-host postgres.example.com \
  --foreman-db-password Foreman_Password \
  --foreman-db-database foreman \
  --foreman-db-manage false \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password Candlepin_Password \
  --katello-candlepin-manage-db false \
  --foreman-proxy-content-pulpcore-manage-postgresql false \
  --foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \
  --foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \
  --foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password
  --foreman-proxy-content-pulpcore-postgresql-user pulp
```

To enable the Secure Sockets Layer (SSL) protocol for these external databases, add the following options:

```
--foreman-db-sslmode verify-full
--foreman-db-root-cert <path_to_CA>
--katello-candlepin-db-ssl true
--katello-candlepin-db-ssl-verify true
--katello-candlepin-db-ssl-ca <path_to_CA>
--foreman-proxy-content-pulpcore-postgresql-ssl true
--foreman-proxy-content-pulpcore-postgresql-ssl-root-ca <path_to_CA>
```

4.14. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES

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

Note that you cannot use tuning profiles on Capsules.

You can choose one of the profiles depending on the number of hosts your Satellite manages and available hardware resources.

The tuning profiles are available in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes** directory.

When you run the **satellite-installer** command with the **--tuning** option, deployment configuration settings are applied to Satellite in the following order:

1. The default tuning profile defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** file
2. The tuning profile that you want to apply to your deployment and is defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** directory
3. Optional: If you have configured a **/etc/foreman-installer/custom-hiera.yaml** file, Satellite applies these configuration settings.

Note that the configuration settings that are defined in the **/etc/foreman-installer/custom-hiera.yaml** file override the configuration settings that are defined in the tuning profiles.

Therefore, before applying a tuning profile, you must compare the configuration settings that are defined in the default tuning profile in **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml**, the tuning profile that you want to apply and your **/etc/foreman-installer/custom-hiera.yaml** file, and remove any duplicated configuration from the **/etc/foreman-installer/custom-hiera.yaml** file.

default

Number of managed hosts: 0 – 5000
RAM: 20G

Number of CPU cores: 4

medium

Number of managed hosts: 5001 – 10000
RAM: 32G

Number of CPU cores: 8

large

Number of managed hosts: 10001 – 20000
RAM: 64G

Number of CPU cores: 16

extra-large

Number of managed hosts: 20001 – 60000
RAM: 128G

Number of CPU cores: 32

extra-extra-large

Number of managed hosts: 60000+
RAM: 256G

Number of CPU cores: 48+

Procedure

1. Optional: If you have configured the **custom-hiera.yaml** file on Satellite Server, back up the **/etc/foreman-installer/custom-hiera.yaml** file to **custom-hiera.original**. You can use the backup file to restore the **/etc/foreman-installer/custom-hiera.yaml** file to its original state if it becomes corrupted:

```
# cp /etc/foreman-installer/custom-hiera.yaml \
/etc/foreman-installer/custom-hiera.original
```

2. Optional: If you have configured the **custom-hiera.yaml** file on Satellite Server, review the definitions of the default tuning profile in **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** and the tuning profile that you want to apply in **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/**. Compare the configuration entries against the entries in your **/etc/foreman-installer/custom-hiera.yaml** file and remove any duplicated configuration settings in your **/etc/foreman-installer/custom-hiera.yaml** file.

3. Enter the **satellite-installer** command with the **--tuning** option for the profile that you want to apply. For example, to apply the medium tuning profile settings, enter the following command:

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

CHAPTER 5. CONFIGURING EXTERNAL AUTHENTICATION

By using external authentication you can derive user and user group permissions from user group membership in an external identity provider. When you use external authentication, you do not have to create these users and maintain their group membership manually on Satellite Server. In case the external source does not provide email, it will be requested during the first login through Satellite web UI.

Important User and Group Account Information

All user and group accounts must be local accounts. This is to ensure that there are no authentication conflicts between local accounts on your Satellite Server and accounts in your Active Directory domain.

Your system is not affected by this conflict if your user and group accounts exist in both **/etc/passwd** and **/etc/group** files. For example, to check if entries for **puppet**, **apache**, **foreman** and **foreman-proxy** groups exist in both **/etc/passwd** and **/etc/group** files, enter the following commands:

```
# cat /etc/passwd | grep 'puppet\|apache\|foreman\|foreman-proxy'
# cat /etc/group | grep 'puppet\|apache\|foreman\|foreman-proxy'
```

Scenarios for Configuring External Authentication

Red Hat Satellite supports the following general scenarios for configuring external authentication:

- Using *Lightweight Directory Access Protocol* (LDAP) server as an external identity provider. LDAP is a set of open protocols used to access centrally stored information over a network. With Satellite, you can manage LDAP entirely through the Satellite web UI. For more information, see Section 5.1, “Using LDAP”. Though you can use LDAP to connect to a Red Hat Identity Management or AD server, the setup does not support server discovery, cross-forest trusts, or single sign-on with Kerberos in Satellite’s web UI.

- Using a Red Hat Identity Management server as an external identity provider. Red Hat Identity Management deals with the management of individual identities, their credentials and privileges used in a networking environment. Configuration using Red Hat Identity Management cannot be completed using only the Satellite web UI and requires some interaction with the CLI. For more information see Section 5.2, “Using Red Hat Identity Management”.

- Using *Active Directory* (AD) integrated with Red Hat Identity Management through cross-forest Kerberos trust as an external identity provider. For more information see Section 5.3.5, “Active Directory with Cross-Forest Trust”.

- Using Red Hat Single Sign-On as an OpenID provider for external authentication to Satellite. For more information, see Section 5.8, “Configuring Satellite with Red Hat Single Sign-On Authentication”.

- Using Red Hat Single Sign-On as an OpenID provider for external authentication to Satellite with TOTP. For more information, see Section 5.9, “Configuring Red Hat Single Sign-On Authentication with TOTP”.

As well as providing access to Satellite Server, hosts provisioned with Satellite can also be integrated with Red Hat Identity Management realms. Red Hat Satellite has a realm feature that automatically manages the life cycle of any system registered to a realm or domain provider. For more information, see Section 5.7, “External Authentication for Provisioned Hosts”.

Table 5.1. Authentication Overview

Type	Authentication	User Groups
Red Hat Identity Management	Kerberos or LDAP	Yes
Active Directory	Kerberos or LDAP	Yes
POSIX	LDAP	Yes

5.1. USING LDAP

Satellite supports LDAP authentication using one or multiple LDAP directories.

If you require Red Hat Satellite to use **TLS** to establish a secure LDAP connection (LDAPS), first obtain certificates used by the LDAP server you are connecting to and mark them as trusted on the base operating system of your Satellite Server as described below. If your LDAP server uses a certificate chain with intermediate certificate authorities, all of the root and intermediate certificates in the chain must be trusted, so ensure all certificates are obtained. If you do not require secure LDAP at this time, proceed to Section 5.1.2, "Configuring Red Hat Satellite to use LDAP" .

Using SSSD Configuration

Though direct LDAP integration is covered in this section, Red Hat recommends that you use SSSD and configure it against Red Hat Identity Management, AD, or an LDAP server. SSSD improves the consistency of the authentication process. For more information about the preferred configurations, see Section 5.3, "Using Active Directory" . You can also cache the SSSD credentials and use them for LDAP authentication. For more information on SSSD, see Configuring SSSD in the *Red Hat Enterprise Linux 8 Configuring Authentication and Authorization in RHEL Guide*.

5.1.1. Configuring TLS for Secure LDAP

Use the Satellite CLI to configure TLS for secure LDAP (LDAPS).

Procedure

1. Obtain the Certificate from the LDAP Server.

   a. If you use Active Directory Certificate Services, export the Enterprise PKI CA Certificate using the Base-64 encoded X.509 format. See How to configure Active Directory authentication with **TLS** on Satellite for information on creating and exporting a CA certificate from an Active Directory server.

   b. Download the LDAP server certificate to a temporary location onto Satellite Server and remove it when finished.
   For example, **/tmp/example.crt**. The filename extensions **.cer** and **.crt** are only conventions and can refer to DER binary or PEM ASCII format certificates.

2. Trust the Certificate from the LDAP Server.
   Satellite Server requires the CA certificates for LDAP authentication to be individual files in **/etc/pki/tls/certs/** directory.

   a. Use the **install** command to install the imported certificate into the **/etc/pki/tls/certs/** directory with the correct permissions:

```
# install /tmp/example.crt /etc/pki/tls/certs/
```

b. Enter the following command as **root** to trust the *example.crt* certificate obtained from the LDAP server:

```
# ln -s example.crt /etc/pki/tls/certs/$(openssl \
x509 -noout -hash -in \
/etc/pki/tls/certs/example.crt).0
```

c. Restart the **httpd** service:

```
# systemctl restart httpd
```

5.1.2. Configuring Red Hat Satellite to use LDAP

In the Satellite web UI, configure Satellite to use LDAP.

Note that if you need single sign-on functionality with Kerberos on Satellite web UI, you should use Red Hat Identity Management and AD external authentication instead. For more information, see Using Red Hat Identity Management or Using Active Directory.

Procedure

1. Set the Network Information System (NIS) service boolean to true to prevent SELinux from stopping outgoing LDAP connections:

```
# setsebool -P nis_enabled on
```

2. In the Satellite web UI, navigate to **Administer** > **LDAP Authentication**.
3. Click **Create Authentication Source**.
4. On the **LDAP server** tab, enter the LDAP server's name, host name, port, and server type. The default port is 389, the default server type is POSIX (alternatively you can select FreeIPA or Active Directory depending on the type of authentication server). For **TLS** encrypted connections, select the **LDAPS** checkbox to enable encryption. The port should change to 636, which is the default for LDAPS.
5. On the **Account** tab, enter the account information and domain name details. See Section 5.1.3, "Description of LDAP Settings" for descriptions and examples.
6. On the **Attribute mappings** tab, map LDAP attributes to Satellite attributes. You can map login name, first name, last name, email address, and photo attributes. See Section 5.1.4, "Example Settings for LDAP Connections" for examples.
7. On the **Locations** tab, select locations from the left table. Selected locations are assigned to users created from the LDAP authentication source, and available after their first login.
8. On the **Organizations** tab, select organizations from the left table. Selected organizations are assigned to users created from the LDAP authentication source, and available after their first login.
9. Click **Submit**.

10. Configure new accounts for LDAP users:

    - If you did not select **Automatically Create Accounts In Satellite** checkbox, see Creating a User in *Administering Red Hat Satellite* to create user accounts manually.
    - If you selected the **Automatically Create Accounts In Satellite** checkbox, LDAP users can now log in to Satellite using their LDAP accounts and passwords. After they log in for the first time, the Satellite administrator has to assign roles to them manually. For more information on assigning user accounts appropriate roles in Satellite, see Assigning Roles to a User in *Administering Red Hat Satellite*.

5.1.3. Description of LDAP Settings

The following table provides a description for each setting in the **Account** tab.

Table 5.2. Account Tab Settings

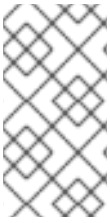
Setting	Description
Account	The user name of the LDAP account that has read access to the LDAP server. User name is not required if the server allows anonymous reading, otherwise use the full path to the user's object. For example: `uid=$login,cn=users,cn=accounts,dc=example,dc=com` The **$login** variable stores the username entered on the login page as a literal string. The value is accessed when the variable is expanded. The variable cannot be used with external user groups from an LDAP source because Satellite needs to retrieve the group list without the user logging in. Use either an anonymous, or dedicated service user.
Account password	The LDAP password for the user defined in the **Account username** field. This field can remain blank if the **Account username** is using the **$login** variable.
Base DN	The top level domain name of the LDAP directory.
Groups base DN	The top level domain name of the LDAP directory tree that contains groups.
LDAP filter	A filter to restrict LDAP queries.
Automatically Create Accounts In Satellite	If this checkbox is selected, Satellite creates user accounts for LDAP users when they log in to Satellite for the first time. After they log in for the first time, the Satellite administrator has to assign roles to them manually. See Assigning Roles to a User in *{AdministeringDocTitle)* to assign user accounts appropriate roles in Satellite.
Usergroup Sync	If this option is selected, the user group membership of a user is automatically synchronized when the user logs in, which ensures the membership is always up to date. If this option is cleared, Satellite relies on a cron job to regularly synchronize group membership (every 30 minutes by default). For more information, see Section 5.4, "Configuring External User Groups".

5.1.4. Example Settings for LDAP Connections

The following table shows example settings for different types of LDAP connections. The example below uses a dedicated service account called *redhat* that has bind, read, and search permissions on the user and group entries. Note that LDAP attribute names are case sensitive.

Table 5.3. Example Settings for Active Directory, Free IPA or Red Hat Identity Management and POSIX LDAP Connections

Setting	Active Directory	FreeIPA or Red Hat Identity Management	POSIX (OpenLDAP)
Account	DOMAIN\redhat	uid=redhat,cn=users, cn=accounts,dc=example, dc=com	uid=redhat,ou=users, dc=example,dc=com
Account password	P@ssword	-	-
Base DN	DC=example,DC=COM	dc=example,dc=com	dc=example,dc=com
Groups Base DN	CN=Users,DC=example,DC=com	cn=groups,cn=accounts, dc=example,dc=com	cn=employee,ou=userclass, dc=example,dc=com
Login name attribute	userPrincipalName	uid	uid
First name attribute	givenName	givenName	givenName
Last name attribute	sn	sn	sn
Email address attribute	mail	mail	mail
Photo attribute	thumbnailPhoto	-	-



NOTE

userPrincipalName allows the use of whitespace in usernames. The login name attribute **sAMAccountName** (which is not listed in the table above) provides backwards compatibility with legacy Microsoft systems. **sAMAccountName** does not allow the use of whitespace in usernames.

5.1.5. Example LDAP Filters

As an administrator, you can create LDAP filters to restrict the access of specific users to Satellite.

Table 5.4. Example filters for allowing specific users to login

User	Filter
User1, User3	(memberOf=cn=Group1,cn=Users,dc=domain,dc=example)
User2, User3	(memberOf=cn=Group2,cn=Users,dc=domain,dc=example)
User1, User2, User3	(|(memberOf=cn=Group1,cn=Users,dc=domain,dc=example) (memberOf=cn=Group2,cn=Users,dc=domain,dc=example))

LDAP directory structure

The LDAP directory structure that the filters in the example use:

```
DC=Domain,DC=Example
  |
  |----- CN=Users
        |
        |----- CN=Group1
        |----- CN=Group2
        |----- CN=User1
        |----- CN=User2
        |----- CN=User3
```

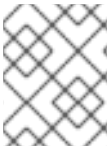
LDAP group membership

The group membership that the filters in the example use:

Group	Members
Group1	User1, User3
Group2	User2, User3

5.2. USING RED HAT IDENTITY MANAGEMENT

This section shows how to integrate Satellite Server with a Red Hat Identity Management server and how to enable host-based access control.



NOTE

You can attach Red Hat Identity Management as an external authentication source with no single sign-on support. For more information, see Section 5.1, “Using LDAP”.

Prerequisites

- The base operating system of Satellite Server must be enrolled in the Red Hat Identity Management domain by the Red Hat Identity Management administrator of your organization.

The examples in this chapter assume separation between Red Hat Identity Management and Satellite configuration. However, if you have administrator privileges for both servers, you can configure Red Hat Identity Management as described in Red Hat Enterprise Linux 8 Installing Identity Management Guide .

5.2.1. Configuring Red Hat Identity Management Authentication on Satellite Server

In the Satellite CLI, configure Red Hat Identity Management authentication by first creating a host entry on the Red Hat Identity Management server.

Procedure

1. On the Red Hat Identity Management server, to authenticate, enter the following command and enter your password when prompted:

```
# kinit admin
```

2. To verify that you have authenticated, enter the following command:

```
# klist
```

3. On the Red Hat Identity Management server, create a host entry for Satellite Server and generate a one-time password, for example:

```
# ipa host-add --random hostname
```

NOTE

The generated one-time password must be used on the client to complete Red Hat Identity Management-enrollment.

For more information on host configuration properties, see Host entry in IdM LDAP in *Configuring and managing Identity Management* .

4. Create an HTTP service for Satellite Server, for example:

```
# ipa service-add HTTP/hostname
```

For more information on managing services, see Red Hat Enterprise Linux 8 Accessing Identity Management Services guide.

5. On Satellite Server, install the IPA client:

WARNING

This command might restart Satellite services during the installation of the package. For more information about installing and updating packages on Satellite, see Managing Packages on the Base Operating System of Satellite Server or Capsule Server in *Administering Red Hat Satellite* .

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

6. On Satellite Server, enter the following command as root to configure Red Hat Identity Management-enrollment:

```
# ipa-client-install --password OTP
```

Replace *OTP* with the one-time password provided by the Red Hat Identity Management administrator.

7. If Satellite Server is running on Red Hat Enterprise Linux 7, execute the following command:

```
# subscription-manager repos --enable rhel-7-server-optional-rpms
```

The installer is dependent on packages which, on Red Hat Enterprise Linux 7, are in the optional repository **rhel-7-server-optional-rpms**.

8. Set **foreman-ipa-authentication** to true, using the following command:

```
# satellite-installer --foreman-ipa-authentication=true
```

9. Restart Satellite services:

```
# satellite-maintain service restart
```

External users can now log in to Satellite using their Red Hat Identity Management credentials. They can now choose to either log in to Satellite Server directly using their username and password or take advantage of the configured Kerberos single sign-on and obtain a ticket on their client machine and be logged in automatically. The two-factor authentication with one-time password (2FA OTP) is also supported. If the user in Red Hat Identity Management is configured for 2FA, and Satellite Server is running on Red Hat Enterprise Linux 7, this user can also authenticate to Satellite with an OTP.

5.2.2. Configuring Host-Based Authentication Control

HBAC rules define which machine within the domain a Red Hat Identity Management user is allowed to access. You can configure HBAC on the Red Hat Identity Management server to prevent selected users from accessing Satellite Server. With this approach, you can prevent Satellite from creating database entries for users that are not allowed to log in. For more information on HBAC, see Managing IdM Users, Groups, Hosts, and Access Control Rules Guide.

On the Red Hat Identity Management server, configure Host-Based Authentication Control (HBAC).

Procedure

1. On the Red Hat Identity Management server, to authenticate, enter the following command and enter your password when prompted:

```
# kinit admin
```

2. To verify that you have authenticated, enter the following command:

```
# klist
```

3. Create HBAC service and rule on the Red Hat Identity Management server and link them together. The following examples use the PAM service name *satellite-prod*. Execute the following commands on the Red Hat Identity Management server:

```
# ipa hbacsvc-add satellite-prod
# ipa hbacrule-add allow_satellite_prod
# ipa hbacrule-add-service allow_satellite_prod --hbacsvcs=satellite-prod
```

4. Add the user who is to have access to the service satellite-prod, and the hostname of Satellite Server:

```
# ipa hbacrule-add-user allow_satellite_prod --user=username
# ipa hbacrule-add-host allow_satellite_prod --hosts=satellite.example.com
```

   Alternatively, host groups and user groups can be added to the *allow_satellite_prod* rule.

5. To check the status of the rule, execute:

```
# ipa hbacrule-find satellite-prod
# ipa hbactest --user=username --host=satellite.example.com --service=satellite-prod
```

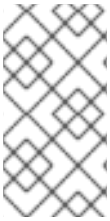
6. Ensure the allow_all rule is disabled on the Red Hat Identity Management server. For instructions on how to do so without disrupting other services see the How to configure HBAC rules in IdM article on the Red Hat Customer Portal.

7. Configure the Red Hat Identity Management integration with Satellite Server as described in Section 5.2.1, “Configuring Red Hat Identity Management Authentication on Satellite Server” . On Satellite Server, define the PAM service as root:

```
# satellite-installer --foreman-pam-service=satellite-prod
```

5.3. USING ACTIVE DIRECTORY

This section shows how to use direct Active Directory (AD) as an external authentication source for Satellite Server.



NOTE

You can attach Active Directory as an external authentication source with no single sign-on support. For more information, see Section 5.1, “Using LDAP”. For an example configuration, see How to configure Active Directory authentication with TLS on Satellite.

Direct AD integration means that Satellite Server is joined directly to the AD domain where the identity is stored. The recommended setup consists of two steps:

- Enrolling Satellite Server with the Active Directory server as described in Section 5.3.2, “Enrolling Satellite Server with the AD Server”.
- Configuring direct Active Directory integration with GSS-proxy as described in Section 5.3.3, “Configuring Direct AD Integration with GSS-Proxy”.

5.3.1. GSS-Proxy

The traditional process of Kerberos authentication in Apache requires the Apache process to have read access to the keytab file. GSS-Proxy allows you to implement stricter privilege separation for the Apache server by removing access to the keytab file while preserving Kerberos authentication functionality. When using AD as an external authentication source for Satellite, it is recommended to implement GSS-proxy, because the keys in the keytab file are the same as the host keys.

Perform the following procedures on Red Hat Enterprise Linux that acts as a base operating system for your Satellite Server. For the examples in this section *EXAMPLE.ORG* is the Kerberos realm for the AD domain. By completing the procedures, users that belong to the EXAMPLE.ORG realm can log in to Satellite Server.

5.3.2. Enrolling Satellite Server with the AD Server

In the Satellite CLI, enroll Satellite Server with the Active Directory server.

Prerequisite

- GSS-proxy and nfs-utils are installed.
  Installing GSS-proxy and nfs-utils:

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

Procedure

1. Install the required packages:

```
# satellite-maintain packages install sssd adcli realmd ipa-python-compat krb5-workstation
samba-common-tools
```

2. Enroll Satellite Server with the AD server. You may need to have administrator permissions to perform the following command:

```
# realm join -v EXAMPLE.ORG --membership-software=samba -U Administrator
```

5.3.3. Configuring Direct AD Integration with GSS-Proxy

In the Satellite CLI, configure the direct Active Directory integration with GSS-proxy.

Prerequisite

- Satellite is enrolled with the Active Directory server. For more information, see Section 5.3.2, “Enrolling Satellite Server with the AD Server”.

Procedure

1. Create the **/etc/ipa/** directory and the **default.conf** file:

```
# mkdir /etc/ipa
# touch /etc/ipa/default.conf
```

2. To the **default.conf** file, add the following content:

```
[global]
server = unused
realm = EXAMPLE.ORG
```

3. Create the **/etc/net-keytab.conf** file with the following content:

```
[global]
workgroup = EXAMPLE
realm = EXAMPLE.ORG
kerberos method = system keytab
security = ads
```

4. Determine the effective user ID of the Apache user:

```
# id apache
```

Apache user must not have access to the keytab file.

5. Create the **/etc/gssproxy/00-http.conf** file with the following content:

```
[service/HTTP]
mechs = krb5
cred_store = keytab:/etc/krb5.keytab
cred_store = ccache:/var/lib/gssproxy/clients/krb5cc_%U
euid = ID_of_Apache_User
```

6. Create a keytab entry:

```
# KRB5_KTNAME=FILE:/etc/httpd/conf/http.keytab net ads keytab add HTTP -U
administrator -d3 -s /etc/net-keytab.conf
# chown root.apache /etc/httpd/conf/http.keytab
# chmod 640 /etc/httpd/conf/http.keytab
```

7. Enable IPA authenication in Satellite:

```
# satellite-installer --foreman-ipa-authentication=true
```

8. Start and enable the **gssproxy** service:

```
# systemctl restart gssproxy
# systemctl enable --now gssproxy
```

9. To configure the Apache server to use the **gssproxy** service, create a **systemd** drop-in file and add the following content to it:

```
# mkdir -p /etc/systemd/system/httpd.service.d/
# vi /etc/systemd/system/httpd.service.d/gssproxy.conf
[Service]
Environment=GSS_USE_PROXY=1
```

10. Apply changes to the service:

```
# systemctl daemon-reload
```

11. Start and enable the **httpd** service:

```
# systemctl restart httpd
```

12. Verify that SSO is working as expected.
With a running Apache server, users making HTTP requests against the server are authenticated if the client has a valid Kerberos ticket.

    a. Retrieve the Kerberos ticket of the LDAP user, using the following command:

```
# kinit ldapuser
```

    b. View the Kerberos ticket, using the following command:

```
# klist
```

    c. View output from successful SSO-based authentication, using the following command:

```
# curl -k -u : --negotiate https://satellite.example.com/users/extlogin
```

    This returns the following response:

```
<html><body>You are being <a href="https://satellite.example.com/users/4-ldapuserexample-com/edit">redirected</a>.</body></html>
```

5.3.4. Kerberos Configuration in Web Browsers

For information on configuring Firefox, see Configuring Firefox to Use Kerberos for Single Sign-On in the *Red Hat Enterprise Linux Configuring authentication and authorization in RHEL* guide.

If you use the Internet Explorer browser, add Satellite Server to the list of Local Intranet or Trusted sites, and turn on the *Enable Integrated Windows Authentication* setting. See the Internet Explorer documentation for details.

NOTE

With direct AD integration, HBAC through Red Hat Identity Management is not available. As an alternative, you can use Group Policy Objects (GPO) that enable administrators to centrally manage policies in AD environments. To ensure correct GPO to PAM service mapping, use the following sssd configuration:

```
access_provider = ad
ad_gpo_access_control = enforcing
ad_gpo_map_service = +foreman
```

Here, *foreman* is the PAM service name. For more information on GPOs, see How SSSD interprets GPO access control rules in *Integrating RHEL systems directly with Windows Active Directory*.

5.3.5. Active Directory with Cross-Forest Trust

Kerberos can create **cross-forest trust** that defines a relationship between two otherwise separate domain forests. A domain forest is a hierarchical structure of domains; both AD and Red Hat Identity Management constitute a forest. With a trust relationship enabled between AD and Red Hat Identity Management, users of AD can access Linux hosts and services using a single set of credentials. For more information on cross-forest trusts, see Planning a cross-forest trust between IdM and AD in *Red Hat Enterprise Linux Planning Identity Management* .

From the Satellite point of view, the configuration process is the same as integration with Red Hat Identity Management server without cross-forest trust configured. Satellite Server has to be enrolled in the IPM domain and integrated as described in Section 5.2, “Using Red Hat Identity Management”.

5.3.6. Configuring the Red Hat Identity Management Server to Use Cross-Forest Trust

On the Red Hat Identity Management server, configure the server to use **cross-forest trust**.

Procedure

1. Enable HBAC:
   a. Create an external group and add the AD group to it.
   b. Add the new external group to a POSIX group.
   c. Use the POSIX group in a HBAC rule.
2. Configure sssd to transfer additional attributes of AD users.
   - Add the AD user attributes to the *nss* and *domain* sections in **/etc/sssd/sssd.conf**. For example:

```
[nss]
user_attributes=+mail, +sn, +givenname

[domain/EXAMPLE]
ldap_user_extra_attrs=mail, sn, givenname
```

5.4. CONFIGURING EXTERNAL USER GROUPS

Satellite does not associate external users with their user group automatically. You must create a user group with the same name as in the external source on Satellite. Members of the external user group then automatically become members of the Satellite user group and receive the associated permissions.

The configuration of external user groups depends on the type of external authentication.

To assign additional permissions to an external user, add this user to an internal user group that has no external mapping specified. Then assign the required roles to this group.

Prerequisites

- If you use an LDAP server, configure Satellite to use LDAP authentication. For more information see Section 5.1, “Using LDAP”.
  When using external user groups from an LDAP source, you cannot use the **$login** variable as a substitute for the account user name. You must use either an anonymous or dedicated service user.

- If you use a Red Hat Identity Management or AD server, configure Satellite to use Red Hat Identity Management or AD authentication. For more information, see Configuring External Authentication in *Installing Satellite Server in a Connected Network Environment* .
- Ensure that at least one external user authenticates for the first time.
- Retain a copy of the external group names you want to use. To find the group membership of external users, enter the following command:

```
# id username
```

Procedure

1. In the Satellite web UI, navigate to **Administer** > **User Groups**, and click **Create User Group**.
2. Specify the name of the new user group. Do not select any users to avoid adding users automatically when you refresh the external user group.
3. Click the **Roles** tab and select the roles you want to assign to the user group. Alternatively, select the **Administrator** checkbox to assign all available permissions.
4. Click the **External groups** tab, then click **Add external user group**, and select an authentication source from the **Auth source** drop-down menu.
   Specify the exact name of the external group in the **Name** field.
5. Click **Submit**.

5.5. REFRESHING EXTERNAL USER GROUPS FOR LDAP

To set the LDAP source to synchronize user group membership automatically on user login, in the **Auth Source** page, select the **Usergroup Sync** option. If this option is not selected, LDAP user groups are refreshed automatically through a scheduled cron job synchronizing the LDAP Authentication source every 30 minutes by default.

If the user groups in the LDAP Authentication source change in the lapse of time between scheduled tasks, the user can be assigned to incorrect external user groups. This is corrected automatically when the scheduled task runs.

Use this procedure to refresh the LDAP source manually.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Usergroups** and select a user group.
2. On the **External Groups** tab, click **Refresh** to the right of the required user group.

CLI procedure

- Enter the following command:

```
# foreman-rake ldap:refresh_usergroups
```

5.6. REFRESHING EXTERNAL USER GROUPS FOR RED HAT IDENTITY MANAGEMENT OR AD

External user groups based on Red Hat Identity Management or AD are refreshed only when a group member logs in to Satellite. It is not possible to alter user membership of external user groups in the Satellite web UI, such changes are overwritten on the next group refresh.

5.7. EXTERNAL AUTHENTICATION FOR PROVISIONED HOSTS

Use this section to configure Satellite Server or Capsule Server for Red Hat Identity Management realm support, then add hosts to the Red Hat Identity Management realm group.

Prerequisites

- Satellite Server that is registered to the Content Delivery Network or an external Capsule Server that is registered to Satellite Server.
- A deployed realm or domain provider such as Red Hat Identity Management.

To install and configure Red Hat Identity Management packages on Satellite Server or Capsule Server:

To use Red Hat Identity Management for provisioned hosts, complete the following steps to install and configure Red Hat Identity Management packages on Satellite Server or Capsule Server:

1. Install the **ipa-client** package on Satellite Server or Capsule Server:

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

2. Configure the server as a Red Hat Identity Management client:

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

3. Create a realm proxy user, **realm-capsule**, and the relevant roles in Red Hat Identity Management:

```
# foreman-prepare-realm admin realm-capsule
```

Note the principal name that returns and your Red Hat Identity Management server configuration details because you require them for the following procedure.

To configure Satellite Server or Capsule Server for Red Hat Identity Management Realm Support:

Complete the following procedure on Satellite and every Capsule that you want to use:

1. Copy the **/root/freeipa.keytab** file to any Capsule Server that you want to include in the same principal and realm:

```
# scp /root/freeipa.keytab root@capsule.example.com:/etc/foreman-proxy/freeipa.keytab
```

2. Move the **/root/freeipa.keytab** file to the **/etc/foreman-proxy** directory and set the ownership settings to the **foreman-proxy** user:

```
# mv /root/freeipa.keytab /etc/foreman-proxy
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/freeipa.keytab
```

3. Enter the following command on all Capsules that you want to include in the realm. If you use the integrated Capsule on Satellite, enter this command on Satellite Server:

```
# satellite-installer --foreman-proxy-realm true \
--foreman-proxy-realm-keytab /etc/foreman-proxy/freeipa.keytab \
--foreman-proxy-realm-principal realm-capsule@EXAMPLE.COM \
--foreman-proxy-realm-provider freeipa
```

You can also use these options when you first configure the Satellite Server.

4. Ensure that the most updated versions of the ca-certificates package is installed and trust the Red Hat Identity Management Certificate Authority:

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

5. Optional: If you configure Red Hat Identity Management on an existing Satellite Server or Capsule Server, complete the following steps to ensure that the configuration changes take effect:

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

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

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

   c. Locate the Capsule you have configured for Red Hat Identity Management and from the list in the **Actions** column, select **Refresh**.

To create a realm for the Red Hat Identity Management-enabled Capsule

After you configure your integrated or external Capsule with Red Hat Identity Management, you must create a realm and add the Red Hat Identity Management-configured Capsule to the realm.

Procedure

1. In the Satellite web UI, navigate to **Infrastructure** > **Realms** and click **Create Realm**.
2. In the **Name** field, enter a name for the realm.
3. From the **Realm Type** list, select the type of realm.
4. From the **Realm Capsule** list, select Capsule Server where you have configured Red Hat Identity Management.
5. Click the **Locations** tab and from the **Locations** list, select the location where you want to add the new realm.
6. Click the **Organizations** tab and from the **Organizations** list, select the organization where you want to add the new realm.

7. Click **Submit**.

Updating Host Groups with Realm Information

You must update any host groups that you want to use with the new realm information.

1. In the Satellite web UI, navigate to **Configure** > **Host Groups**, select the host group that you want to update, and click the **Network** tab.
2. From the **Realm** list, select the realm you create as part of this procedure, and then click **Submit**.

Adding Hosts to a Red Hat Identity Management Host Group

Red Hat Identity Management supports the ability to set up automatic membership rules based on a system's attributes. Red Hat Satellite's realm feature provides administrators with the ability to map the Red Hat Satellite host groups to the Red Hat Identity Management parameter **userclass** which allow administrators to configure automembership.

When nested host groups are used, they are sent to the Red Hat Identity Management server as they are displayed in the Red Hat Satellite User Interface. For example, "Parent/Child/Child".

Satellite Server or Capsule Server sends updates to the Red Hat Identity Management server, however automembership rules are only applied at initial registration.

To Add Hosts to a Red Hat Identity Management Host Group:

1. On the Red Hat Identity Management server, create a host group:

```
# ipa hostgroup-add hostgroup_name --desc=hostgroup_description
```

2. Create an **automembership** rule:

```
# ipa automember-add --type=hostgroup hostgroup_name automember_rule
```

Where you can use the following options:

- **automember-add** flags the group as an automember group.
- **--type=hostgroup** identifies that the target group is a host group, not a user group.
- ***automember_rule*** adds the name you want to identify the automember rule by.

3. Define an automembership condition based on the **userclass** attribute:

```
# ipa automember-add-condition --key=userclass --type=hostgroup --inclusive-
regex=^webserver hostgroup_name
----------------------------------
Added condition(s) to "hostgroup_name"
----------------------------------
Automember Rule: automember_rule
Inclusive Regex: userclass=^webserver
----------------------------
Number of conditions added 1
----------------------------
```

Where you can use the following options:

- **automember-add-condition** adds regular expression conditions to identify group members.
- **--key=userclass** specifies the key attribute as **userclass**.
- **--type=hostgroup** identifies that the target group is a host group, not a user group.
- **--inclusive-regex=** *^webserver* identifies matching values with a regular expression pattern.
- *hostgroup_name* – identifies the target host group's name.

When a system is added to Satellite Server's *hostgroup_name* host group, it is added automatically to the Red Hat Identity Management server's "*hostgroup_name*" host group. Red Hat Identity Management host groups allow for Host-Based Access Controls (HBAC), sudo policies and other Red Hat Identity Management functions.

5.8. CONFIGURING SATELLITE WITH RED HAT SINGLE SIGN-ON AUTHENTICATION

Use this section to configure Satellite to use Red Hat Single Sign-On as an OpenID provider for external authentication.

5.8.1. Prerequisites for Configuring Satellite with Red Hat Single Sign-On Authentication

Before configuring Satellite with Red Hat Single Sign-On external authentication, ensure that you meet the following requirements:

- A working installation of Red Hat Single Sign-On server that uses HTTPS instead of HTTP.
- A Red Hat Single Sign-On account with admin privileges.
- A realm for Satellite user accounts created in Red Hat Single Sign-On.
- If the certificates or the CA are self-signed, ensure that they are added to the end-user certificate trust store.
- Users imported or added to Red Hat Single Sign-On.
  If you have an existing user database configured such as LDAP or Kerberos, you can import users from it by configuring user federation. For more information, see User Storage Federation in the *Red Hat Single Sign-On Server Administration Guide* .

  If you do not have an existing user database configured, you can manually create users in Red Hat Single Sign-On. For more information, see Creating New Users in the *Red Hat Single Sign-On Server Administration Guide*.

5.8.2. Registering Satellite as a Red Hat Single Sign-On Client

Use this procedure to register Satellite to Red Hat Single Sign-On as a client and configure Satellite to use Red Hat Single Sign-On as an authentication source.

You can configure Satellite and Red Hat Single Sign-On with two different authentication methods:

1. Users authenticate to Satellite using the Satellite web UI.

2. Users authenticate to Satellite using the Satellite CLI.

You must decide on how you want your users to authenticate in advance because both methods require different Satellite clients to be registered to Red Hat Single Sign-On and configured. The steps to register and configure Satellite client in Red Hat Single Sign-On are distinguished within the procedure.

You can also register two different Satellite clients to Red Hat Single Sign-On if you want to use both authentication methods and configure both clients accordingly.

Procedure

1. On the Satellite server, install the following packages:

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install
```

2. Register Satellite to Red Hat Single Sign-On as a client. Note that you the registration process for logging in using the web UI and the CLI are different. You can register two clients Satellite clients to Red Hat Single Sign-On to be able to log in to Satellite from the web UI and the CLI.

   - If you want you users to authenticate to Satellite using the web UI, create a client as follows:

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

     Enter the password for the administer account when prompted. This command creates a client for Satellite in Red Hat Single Sign-On.

     Then, configure Satellite to use Red Hat Single Sign-On as an authentication source:

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

   - If you want your users to authenticate to Satellite using the CLI, create a client as follows:

```
# keycloak-httpd-client-install --app-name hammer-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

     Enter the password for the administer account when prompted. This command creates a client for Satellite in Red Hat Single Sign-On.

3. Restart the **httpd** service:

```
# systemctl restart httpd
```

5.8.3. Configuring the Satellite Client in Red Hat Single Sign-On

Use this procedure to configure the Satellite client in the Red Hat Single Sign-On web UI and create group and audience mappers for the Satellite client.

Procedure

1. In the Red Hat Single Sign-On web UI, navigate to **Clients** and click the Satellite client.
2. Configure access type:
   - If you want your users to authenticate to Satellite using the Satellite web UI, from the **Access Type** list, select **confidential**.
   - If you want your users to authenticate to Satellite using the CLI, from the **Access Type** list, select **public**.
3. In the **Valid redirect URI** fields, add a valid redirect URI.
   - If you want your users to authenticate to Satellite using the Satellite web UI, in the blank field below the existing URI, enter a URI in the form **https://satellite.example.com/users/extlogin**. Note that you must add the string **/users/extlogin** after the Satellite FQDN.
     After completing this step, the Satellite client for logging in using the Satellite web UI must have the following **Valid Redirect URIs**:

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     https://satellite.example.com/users/extlogin
     ```

   - If you want your users to authenticate to Satellite using the CLI, in the blank field below the existing URI, enter **urn:ietf:wg:oauth:2.0:oob**.
     After completing this step, the Satellite client for logging in using the CLI must have the following **Valid Redirect URIs**:

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     urn:ietf:wg:oauth:2.0:oob
     ```

4. Click **Save**.
5. Click the **Mappers** tab and click **Create** to add an audience mapper.
6. In the **Name** field, enter a name for the audience mapper.
7. From the **Mapper Type** list, select **Audience**.
8. From the **Included Client Audience** list, select the Satellite client.
9. Click **Save**.
10. Click **Create** to add a group mapper so that you can specify authorization in Satellite based on group membership.
11. In the **Name** field, enter a name for the group mapper.
12. From the **Mapper Type** list, select **Group Membership**.

13. In the **Token Claim Name** field, enter **groups**.
14. Set the **Full group path** setting to OFF.
15. Click **Save**.

5.8.4. Configuring Satellite Settings for Red Hat Single Sign-On Authentication

Use this section to configure Satellite for Red Hat Single Sign-On authentication using the Satellite web UI or the CLI.

5.8.4.1. Configuring Satellite Settings for Red Hat Single Sign-On Authentication Using the Web UI

Use this procedure to configure Satellite settings for Red Hat Single Sign-On authentication using the Satellite web UI.

Note that you can navigate to the following URL within your realm to obtain values to configure Satellite settings: **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration**

Prerequisite

- Ensure that the **Access Type** setting in the Satellite client in the Red Hat Single Sign-On web UI is set to **confidential**

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **Authentication** tab.
2. Locate the **Authorize login delegation** row, and in the **Value** column, set the value to **Yes**.
3. Locate the **Authorize login delegation auth source user autocreate** row, and in the **Value** column, set the value to **External**.
4. Locate the **Login delegation logout URL** row, and in the **Value** column, set the value to **https://satellite.example.com/users/extlogout**.
5. Locate the **OIDC Algorithm** row, and in the **Value** column, set the algorithm for encoding on Red Hat Single Sign-On to **RS256**.
6. Locate the **OIDC Audience** row, and in the **Value** column, set the value to the client ID for Red Hat Single Sign-On.
7. Locate the **OIDC Issuer** row, and in the **Value** column, set the value to *https://RHSSO.example.com/auth/realms/Satellite_Realm*.
8. Locate the **OIDC JWKs URL** row, and in the **Value** column, set the value to *https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs*.
9. In the Satellite web UI, navigate to **Administer** > **Authentication Sources** and click **External**.
10. Click **Create LDAP Authentication Source** and select the Red Hat Single Sign-On server.
11. Click the **Locations** tab and add locations that can use the Red Hat Single Sign-On authentication source.

12. Click the **Organizations** tab and add organizations that can use the Red Hat Single Sign-On authentication source.

13. Click **Submit**.

5.8.4.2. Configuring Satellite Settings for Red Hat Single Sign-On Authentication Using the CLI

Use this procedure to configure Satellite settings for Red Hat Single Sign-On authentication using the Satellite CLI.

Note that you can navigate to the following URL within your realm to obtain values to configure Satellite settings: **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration**

Prerequisite

- Ensure that the **Access Type** setting in the Satellite client in the Red Hat Single Sign-On web UI is set to **public**

Procedure

1. On Satellite, set the login delegation to **true** so that users can authenticate using the Open IDC protocol:

```
# hammer settings set --name authorize_login_delegation --value true
```

2. Set the login delegation logout URL:

```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

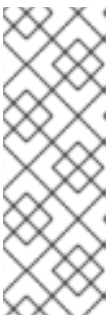
3. Set the algorithm for encoding on Red Hat Single Sign-On, for example, **RS256**:

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

4. Open the ***RHSSO.example.com*/auth/realms/*RHSSO_REALM*/.well-known/openid-configuration** URL and note the values to populate the options in the following steps.

5. Add the value for the Hammer client in the Open IDC audience:

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-hammer-openidc']"
```



NOTE

If you register several Red Hat Single Sign-On clients to Satellite, ensure that you append all audiences in the array. For example:

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-foreman-openidc', 'satellite.example.com-hammer-openidc']"
```

6. Set the value for the Open IDC issuer:

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

7. Set the value for Open IDC Java Web Token (JWT):

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

8. Retrieve the ID of the Red Hat Single Sign-On authentication source:

```
# hammer auth-source external list
```

9. Set the location and organization:

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

5.8.5. Logging in to the Satellite web UI Using Red Hat Single Sign-On

Use this procedure to log in to the Satellite web UI using Red Hat Single Sign-On.

Procedure

- In your browser, log in to Satellite and enter your credentials.

5.8.6. Logging in to the Satellite CLI Using Red Hat Single Sign-On

Use this procedure to authenticate to the Satellite CLI using the code grant type.

Procedure

1. To authenticate to the Satellite CLI using the code grant type, enter the following command:

```
# hammer auth login oauth \
--two-factor \
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-connect/token' \
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \
--oidc-client-id 'satellite.example.com-foreman-openidc' \
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

   The command prompts you to enter a success code.

2. To retrieve the success code, navigate to the URL that the command returns and provide the required information.

3. Copy the success code that the web UI returns.

4. In the command prompt of **hammer auth login oauth**, enter the success code to authenticate to the Satellite CLI.

5.8.7. Configuring Group Mapping for Red Hat Single Sign-On Authentication

Optionally, to implement the Role Based Access Control (RBAC), create a group in Satellite, assign a role to this group, and then map an Active Directory group to the Satellite group. As a result, anyone in the given group in Red Hat Single Sign-On are logged in under the corresponding Satellite group. This example configures users of the Satellite-admin user group in the Active Directory to authenticate as users with administrator privileges on Satellite.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **User Groups**, and click the **Create User Group** button.
2. In the **Name** field, enter a name for the user group. The name should not be the same as in the Active Directory.
3. Do not add users and user groups to the right-hand columns. Click the **Roles** tab.
4. Select the **Administer** checkbox.
5. Click the **External Groups** tab.
6. Click **Add external user group**.
7. In the **Name** field, enter the name of the Active Directory group.
8. From the list, select **EXTERNAL**.

5.9. CONFIGURING RED HAT SINGLE SIGN-ON AUTHENTICATION WITH TOTP

Use this section to configure Satellite to use Red Hat Single Sign-On as an OpenID provider for external authentication with TOTP cards.

5.9.1. Prerequisites for Configuring Satellite with Red Hat Single Sign-On Authentication

Before configuring Satellite with Red Hat Single Sign-On external authentication, ensure that you meet the following requirements:

- A working installation of Red Hat Single Sign-On server that uses HTTPS instead of HTTP.
- A Red Hat Single Sign-On account with admin privileges.
- A realm for Satellite user accounts created in Red Hat Single Sign-On.
- If the certificates or the CA are self-signed, ensure that they are added to the end-user certificate trust store.
- Users imported or added to Red Hat Single Sign-On.
  If you have an existing user database configured such as LDAP or Kerberos, you can import users from it by configuring user federation. For more information, see User Storage Federation in the *Red Hat Single Sign-On Server Administration Guide* .

If you do not have an existing user database configured, you can manually create users in Red Hat Single Sign-On. For more information, see Creating New Users in the *Red Hat Single Sign-On Server Administration Guide*.

5.9.2. Registering Satellite as a Red Hat Single Sign-On Client

Use this procedure to register Satellite to Red Hat Single Sign-On as a client and configure Satellite to use Red Hat Single Sign-On as an authentication source.

You can configure Satellite and Red Hat Single Sign-On with two different authentication methods:

1. Users authenticate to Satellite using the Satellite web UI.
2. Users authenticate to Satellite using the Satellite CLI.

You must decide on how you want your users to authenticate in advance because both methods require different Satellite clients to be registered to Red Hat Single Sign-On and configured. The steps to register and configure Satellite client in Red Hat Single Sign-On are distinguished within the procedure.

You can also register two different Satellite clients to Red Hat Single Sign-On if you want to use both authentication methods and configure both clients accordingly.

Procedure

1. On the Satellite server, install the following packages:

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install
```

2. Register Satellite to Red Hat Single Sign-On as a client. Note that you the registration process for logging in using the web UI and the CLI are different. You can register two clients Satellite clients to Red Hat Single Sign-On to be able to log in to Satellite from the web UI and the CLI.

   - If you want you users to authenticate to Satellite using the web UI, create a client as follows:

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

     Enter the password for the administer account when prompted. This command creates a client for Satellite in Red Hat Single Sign-On.

     Then, configure Satellite to use Red Hat Single Sign-On as an authentication source:

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

   - If you want your users to authenticate to Satellite using the CLI, create a client as follows:

```
# keycloak-httpd-client-install --app-name hammer-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
```

```
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

Enter the password for the administer account when prompted. This command creates a client for Satellite in Red Hat Single Sign-On.

3. Restart the **httpd** service:

```
# systemctl restart httpd
```

5.9.3. Configuring the Satellite Client in Red Hat Single Sign-On

Use this procedure to configure the Satellite client in the Red Hat Single Sign-On web UI and create group and audience mappers for the Satellite client.

Procedure

1. In the Red Hat Single Sign-On web UI, navigate to **Clients** and click the Satellite client.
2. Configure access type:
   - If you want your users to authenticate to Satellite using the Satellite web UI, from the **Access Type** list, select **confidential**.
   - If you want your users to authenticate to Satellite using the CLI, from the **Access Type** list, select **public**.
3. In the **Valid redirect URI** fields, add a valid redirect URI.
   - If you want your users to authenticate to Satellite using the Satellite web UI, in the blank field below the existing URI, enter a URI in the form **https://satellite.example.com/users/extlogin**. Note that you must add the string **/users/extlogin** after the Satellite FQDN.
     After completing this step, the Satellite client for logging in using the Satellite web UI must have the following **Valid Redirect URIs**:

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     https://satellite.example.com/users/extlogin
     ```

   - If you want your users to authenticate to Satellite using the CLI, in the blank field below the existing URI, enter **urn:ietf:wg:oauth:2.0:oob**.
     After completing this step, the Satellite client for logging in using the CLI must have the following **Valid Redirect URIs**:

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     urn:ietf:wg:oauth:2.0:oob
     ```

4. Click **Save**.
5. Click the **Mappers** tab and click **Create** to add an audience mapper.
6. In the **Name** field, enter a name for the audience mapper.

7. From the **Mapper Type** list, select **Audience**.
8. From the **Included Client Audience** list, select the Satellite client.
9. Click **Save**.
10. Click **Create** to add a group mapper so that you can specify authorization in Satellite based on group membership.
11. In the **Name** field, enter a name for the group mapper.
12. From the **Mapper Type** list, select **Group Membership**.
13. In the **Token Claim Name** field, enter **groups**.
14. Set the **Full group path** setting to OFF.
15. Click **Save**.

5.9.4. Configuring Satellite Settings for Red Hat Single Sign-On Authentication

Use this section to configure Satellite for Red Hat Single Sign-On authentication using the Satellite web UI or the CLI.

5.9.4.1. Configuring Satellite Settings for Red Hat Single Sign-On Authentication Using the Web UI

Use this procedure to configure Satellite settings for Red Hat Single Sign-On authentication using the Satellite web UI.

Note that you can navigate to the following URL within your realm to obtain values to configure Satellite settings: **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration**

Prerequisite

- Ensure that the **Access Type** setting in the Satellite client in the Red Hat Single Sign-On web UI is set to **confidential**

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **Authentication** tab.
2. Locate the **Authorize login delegation** row, and in the **Value** column, set the value to **Yes**.
3. Locate the **Authorize login delegation auth source user autocreate** row, and in the **Value** column, set the value to **External**.
4. Locate the **Login delegation logout URL** row, and in the **Value** column, set the value to **https://satellite.example.com/users/extlogout**.
5. Locate the **OIDC Algorithm** row, and in the **Value** column, set the algorithm for encoding on Red Hat Single Sign-On to **RS256**.
6. Locate the **OIDC Audience** row, and in the **Value** column, set the value to the client ID for Red Hat Single Sign-On.

7. Locate the **OIDC Issuer** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm***.
8. Locate the **OIDC JWKs URL** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs***.
9. In the Satellite web UI, navigate to **Administer** > **Authentication Sources** and click **External**.
10. Click **Create LDAP Authentication Source**and select the Red Hat Single Sign-On server.
11. Click the **Locations** tab and add locations that can use the Red Hat Single Sign-On authentication source.
12. Click the **Organizations** tab and add organizations that can use the Red Hat Single Sign-On authentication source.
13. Click **Submit**.

5.9.4.2. Configuring Satellite Settings for Red Hat Single Sign-On Authentication Using the CLI

Use this procedure to configure Satellite settings for Red Hat Single Sign-On authentication using the Satellite CLI.

Note that you can navigate to the following URL within your realm to obtain values to configure Satellite settings: **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration**

Prerequisite

- Ensure that the **Access Type** setting in the Satellite client in the Red Hat Single Sign-On web UI is set to **public**

Procedure

1. On Satellite, set the login delegation to **true** so that users can authenticate using the Open IDC protocol:

```
# hammer settings set --name authorize_login_delegation --value true
```

2. Set the login delegation logout URL:

```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

3. Set the algorithm for encoding on Red Hat Single Sign-On, for example, **RS256**:

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

4. Open the ***RHSSO.example.com*/auth/realms/*RHSSO_REALM*/.well-known/openid-configuration** URL and note the values to populate the options in the following steps.
5. Add the value for the Hammer client in the Open IDC audience:

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-hammer-openidc']"
```

NOTE

If you register several Red Hat Single Sign-On clients to Satellite, ensure that you append all audiences in the array. For example:

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-foreman-openidc', 'satellite.example.com-
hammer-openidc']"
```

6. Set the value for the Open IDC issuer:

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

7. Set the value for Open IDC Java Web Token (JWT):

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

8. Retrieve the ID of the Red Hat Single Sign-On authentication source:

```
# hammer auth-source external list
```

9. Set the location and organization:

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

5.9.5. Configuring Satellite with Red Hat Single Sign-On for TOTP Authentication

Use this procedure to configure Satellite to use Red Hat Single Sign-On as an OpenID provider for external authentication with Time-based One-time Password (TOTP).

Procedure

1. In the Red Hat Single Sign-On web UI, navigate to the Satellite realm.
2. Navigate to **Authentication**, and click the **OTP Policy** tab.
3. Ensure that the **Supported Applications** field includes FreeOTP or Google Authenticator.
4. Configure the OTP settings to suit your requirements.
5. Optional: If you want to use TOTP authentication as a default authentication method for all users, click the **Flows** tab, and to the right of the **OTP Form** setting, select **REQUIRED**.
6. Click the **Required Actions** tab.
7. To the right of the **Configure OTP** row, select the **Default Action** checkbox.

5.9.6. Logging in to the Satellite web UI Using Red Hat Single Sign-On TOTP Authentication

Use this procedure to log in to the Satellite web UI using Red Hat Single Sign-On TOTP authentication.

Procedure

1. Log in to Satellite, Satellite redirects you to the Red Hat Single Sign-On login screen.
2. Enter your username and password, and click **Log In**.
3. The first attempt to log in, Red Hat Single Sign-On requests you to configure your client by scanning the barcode and entering the pin displayed.
4. After you configure your client and enter a valid PIN, Red Hat Single Sign-On redirects you to Satellite and logs you in.

5.9.7. Logging in to the Satellite CLI Using Red Hat Single Sign-On

Use this procedure to authenticate to the Satellite CLI using the code grant type.

Procedure

1. To authenticate to the Satellite CLI using the code grant type, enter the following command:

```
# hammer auth login oauth \
--two-factor \
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-connect/token' \
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \
--oidc-client-id 'satellite.example.com-foreman-openidc' \
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

   The command prompts you to enter a success code.

2. To retrieve the success code, navigate to the URL that the command returns and provide the required information.
3. Copy the success code that the web UI returns.
4. In the command prompt of **hammer auth login oauth**, enter the success code to authenticate to the Satellite CLI.

5.9.8. Configuring Group Mapping for Red Hat Single Sign-On Authentication

Optionally, to implement the Role Based Access Control (RBAC), create a group in Satellite, assign a role to this group, and then map an Active Directory group to the Satellite group. As a result, anyone in the given group in Red Hat Single Sign-On are logged in under the corresponding Satellite group. This example configures users of the Satellite-admin user group in the Active Directory to authenticate as users with administrator privileges on Satellite.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **User Groups**, and click the **Create User Group** button.

2. In the **Name** field, enter a name for the user group. The name should not be the same as in the Active Directory.
3. Do not add users and user groups to the right-hand columns. Click the **Roles** tab.
4. Select the **Administer** checkbox.
5. Click the **External Groups** tab.
6. Click **Add external user group**.
7. In the **Name** field, enter the name of the Active Directory group.
8. From the list, select **EXTERNAL**.

5.10. DISABLING RED HAT SINGLE SIGN-ON AUTHENTICATION

If you want to disable Red Hat Single Sign-On authentication in Satellite, complete this procedure.

Procedure

- Enter the following command to disable Red Hat Single Sign-On Authentication:

```
# satellite-installer --reset-foreman-keycloak
```

CHAPTER 6. CONFIGURING SATELLITE SERVER WITH EXTERNAL SERVICES

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

6.1. CONFIGURING SATELLITE SERVER WITH EXTERNAL DNS

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

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

Prerequisites

- You must have a configured external DNS server.
- This guide assumes you have an existing installation.

Procedure

1. Copy the **/etc/rndc.key** file from the external DNS server to Satellite 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 Satellite Server and select **Refresh** from the list in the **Actions** column.
7. Associate the DNS service with the appropriate subnets and domain.

6.2. CONFIGURING SATELLITE SERVER WITH EXTERNAL DHCP

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

1. Section 6.2.1, "Configuring an External DHCP Server to Use with Satellite Server"
2. Section 6.2.2, "Configuring Satellite Server with an External DHCP Server"

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

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

NOTE

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

Procedure

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

```
# dnf install dhcp bind
```

2. Generate a security token:

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

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

3. Copy the secret hash from the key:

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

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

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

subnet 192.168.38.0 netmask 255.255.255.0 {
 range 192.168.38.10 192.168.38.100;
 option routers 192.168.38.1;
```

```
 option subnet-mask 255.255.255.0;
 option domain-search "virtual.lan";
 option domain-name "virtual.lan";
 option domain-name-servers 8.8.8.8;
}

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

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

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

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

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

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

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

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

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

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

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

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

11. Enable and start the DHCP service:

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

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

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

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

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

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

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

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

```
# mount -a
```

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

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

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

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

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

17. Reload the NFS server:

```
# exportfs -rva
```

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

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

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

```
# 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
```

6.2.2. Configuring Satellite Server with an External DHCP Server

You can configure Satellite Server with an external DHCP server.

Prerequisite

- Ensure that you have configured an external DHCP server and that you have shared the DHCP configuration and lease files with Satellite Server. For more information, see Section 6.2.1, "Configuring an External DHCP Server to Use with Satellite Server".

Procedure

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

```
# dnf install nfs-utils
```

2. Create the DHCP directories for NFS:

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

3. Change the file owner:

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

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

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

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

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

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

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

```
# mount -a
```

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

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

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

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

```
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=DHCP_Server_FQDN
```

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

6.3. CONFIGURING SATELLITE SERVER WITH EXTERNAL TFTP

You can configure Satellite 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=true \
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot
```

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

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

6. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.
7. Locate the Satellite Server and select **Refresh** from the list in the **Actions** column.
8. Associate the TFTP service with the appropriate subnets and domain.

6.4. CONFIGURING SATELLITE SERVER WITH EXTERNAL IDM DNS

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

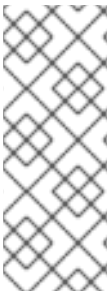
Satellite Server can be configured to use a Red Hat Identity Management (IdM) server to provide DNS service. For more information about Red Hat Identity Management, see the Linux Domain Identity, Authentication, and Policy Guide.

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

- Section 6.4.1, "Configuring Dynamic DNS Update with GSS-TSIG Authentication"
- Section 6.4.2, "Configuring Dynamic DNS Update with TSIG Authentication"

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

- Section 6.4.3, "Reverting to Internal DNS Service"



NOTE

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

6.4.1. Configuring Dynamic DNS Update with GSS-TSIG Authentication

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

Prerequisites

- 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 Satellite Server to use to authenticate on the IdM server.

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

-

Installing and Configuring the IdM Client

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

```
# 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/047satellite.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\047satellite.example.com@EXAMPLE.COM wildcard * ANY;
```

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

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

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

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

- On Satellite, enter the following command:

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--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
```

- On Capsule, enter the following command:

```
satellite-installer --scenario capsule \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--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
```

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**, locate the Satellite Server, and from the list in the **Actions** column, select **Refresh**.
2. Configure the domain:
    a. In the Satellite web UI, navigate to **Infrastructure** > **Domains** and select the domain name.
    b. In the **Domain** tab, ensure **DNS Capsule** is set to the Capsule where the subnet is connected.
3. Configure the subnet:
    a. In the Satellite web UI, navigate to **Infrastructure** > **Subnets** and select the subnet name.
    b. In the **Subnet** tab, set **IPAM** to **None**.
    c. In the **Domains** tab, select the domain that you want to manage using the IdM server.
    d. In the **Capsules** tab, ensure **Reverse DNS Capsule** is set to the Capsule where the subnet is connected.
    e. Click **Submit** to save the changes.

6.4.2. Configuring Dynamic DNS Update with TSIG Authentication

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

Prerequisites

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

Procedure

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

Enabling External Updates to the DNS Zone in the IdM Server

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

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

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

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

```
# systemctl reload named
```

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

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

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

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

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

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

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

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

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

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

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

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

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

Testing External Updates to the DNS Zone in the IdM Server

1. 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.

6.4.3. 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=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1"
```

  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.

APPENDIX A. APPLYING CUSTOM CONFIGURATION TO RED HAT SATELLITE

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

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

Common examples include:

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

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

- To turn off the Apache server signature entirely:

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

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

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

APPENDIX B. RESTORING MANUAL CHANGES OVERWRITTEN BY A PUPPET RUN

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

Procedure

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

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

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

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

3. Restore the overwritten file:

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

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