



Red Hat Satellite 6.12

Administering Red Hat Satellite

A guide to administering Red Hat Satellite.

Red Hat Satellite 6.12 Administering Red Hat Satellite

A guide to administering Red Hat Satellite.

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

Legal Notice

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This guide provides instructions on how to configure and administer a Red Hat Satellite 6 Server. Before continuing with this workflow you must have successfully installed a Red Hat Satellite 6 Server and any required Capsule Servers.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	6
CHAPTER 1. ACCESSING RED HAT SATELLITE	7
1.1. IMPORTING THE KATELLO ROOT CA CERTIFICATE	7
1.2. LOGGING IN TO SATELLITE	7
1.3. NAVIGATION TABS IN THE SATELLITE WEB UI	8
1.4. CHANGING THE PASSWORD	8
1.5. RESETTING THE ADMINISTRATIVE USER PASSWORD	9
1.6. SETTING A CUSTOM MESSAGE ON THE LOGIN PAGE	10
CHAPTER 2. STARTING AND STOPPING RED HAT SATELLITE	11
CHAPTER 3. MIGRATING FROM INTERNAL SATELLITE DATABASES TO EXTERNAL DATABASES	12
3.1. POSTGRESQL AS AN EXTERNAL DATABASE CONSIDERATIONS	12
3.2. PREPARING A HOST FOR EXTERNAL DATABASES	13
3.3. INSTALLING POSTGRESQL	13
3.4. MIGRATING TO EXTERNAL DATABASES	15
CHAPTER 4. MANAGING SATELLITE WITH ANSIBLE COLLECTIONS	17
4.1. INSTALLING THE SATELLITE ANSIBLE MODULES	17
4.2. VIEWING THE SATELLITE ANSIBLE MODULES	17
CHAPTER 5. MANAGING ORGANIZATIONS	18
5.1. CREATING AN ORGANIZATION	18
5.2. SETTING THE ORGANIZATION CONTEXT	19
5.3. CREATING AN ORGANIZATION DEBUG CERTIFICATE	19
5.4. BROWSING REPOSITORY CONTENT USING AN ORGANIZATION DEBUG CERTIFICATE	20
5.5. DELETING AN ORGANIZATION	21
CHAPTER 6. MANAGING LOCATIONS	22
6.1. CREATING A LOCATION	22
6.2. CREATING MULTIPLE LOCATIONS	22
6.3. SETTING THE LOCATION CONTEXT	23
6.4. DELETING A LOCATION	23
CHAPTER 7. MANAGING USERS AND ROLES	24
7.1. MANAGING USERS	24
7.1.1. Creating a User	24
7.1.2. Assigning Roles to a User	25
7.1.3. Impersonating a Different User Account	26
7.1.4. Creating an API-Only User	26
7.2. MANAGING SSH KEYS	27
7.2.1. Managing SSH Keys for a User	27
7.3. MANAGING PERSONAL ACCESS TOKENS	28
7.3.1. Creating a Personal Access Token	28
7.3.2. Revoking a Personal Access Token	29
7.4. CREATING AND MANAGING USER GROUPS	29
7.4.1. User Groups	29
7.4.2. Creating a User Group	30
7.4.3. Removing a User Group	30
7.5. CREATING AND MANAGING ROLES	30
7.5.1. Creating a Role	30
7.5.2. Cloning a Role	31

7.5.3. Adding Permissions to a Role	31
7.5.4. Viewing Permissions of a Role	32
7.5.5. Creating a Complete Permission Table	32
7.5.6. Removing a Role	33
7.6. PREDEFINED ROLES AVAILABLE IN SATELLITE	33
7.7. GRANULAR PERMISSION FILTERING	35
7.7.1. Creating a Granular Permission Filter	35
7.7.2. Examples of Using Granular Permission Filters	36
7.7.2.1. Applying Permissions for the Host Resource Type	36
7.7.2.2. Creating an Organization Specific Manager Role	37
7.7.3. Supported Operators for Granular Search	37
CHAPTER 8. CONFIGURING EMAIL NOTIFICATIONS	39
8.1. NOTIFICATION TYPES	39
8.2. CONFIGURING EMAIL NOTIFICATION PREFERENCES	39
8.3. TESTING EMAIL DELIVERY	40
8.4. TESTING EMAIL NOTIFICATIONS	40
8.5. CHANGING EMAIL NOTIFICATION SETTINGS FOR A HOST	41
CHAPTER 9. MANAGING SECURITY COMPLIANCE	42
9.1. SECURITY CONTENT AUTOMATION PROTOCOL	42
9.1.1. SCAP Content	42
9.1.2. XCCDF Profile	42
9.1.3. Listing Available XCCDF Profiles	42
9.2. INSTALLING THE OPENSAP PLUG-IN	42
9.3. CONFIGURING SCAP CONTENT	43
9.3.1. Importing OpenSCAP Puppet Modules	43
9.3.2. Loading the Default OpenSCAP Content	43
9.3.3. Extra SCAP Content	44
9.3.4. Uploading Extra SCAP Content	44
9.4. MANAGING COMPLIANCE POLICIES	44
9.4.1. Creating a Compliance Policy	45
9.4.2. Viewing a Compliance Policy	46
9.4.3. Editing a Compliance Policy	46
9.4.4. Deleting a Compliance Policy	46
9.5. CUSTOMIZING OPENSAP POLICIES	46
9.5.1. Uploading a Tailoring File	46
9.5.2. Assigning a Tailoring File to a Policy	47
9.6. CONFIGURING A HOST GROUP FOR OPENSAP	47
CHAPTER 10. RUNNING OPENSAP SCANS	49
10.1. CONFIGURING A HOST FOR OPENSAP	49
10.2. MONITORING COMPLIANCE	50
10.2.1. Compliance Policy Dashboard	50
10.2.2. Viewing the Compliance Policy Dashboard	50
10.2.3. Compliance Email Notifications	51
10.2.4. Compliance Reports	51
10.2.5. Examining Compliance Failures of Hosts	52
10.2.6. Searching Compliance Reports	53
10.2.7. Deleting a Compliance Report	54
10.2.8. Deleting Multiple Compliance Reports	54
10.3. SPECIFICATIONS SUPPORTED BY OPENSAP	54
CHAPTER 11. BACKING UP SATELLITE SERVER AND CAPSULE SERVER	56

11.1. ESTIMATING THE SIZE OF A BACKUP	56
11.2. PERFORMING A FULL BACKUP OF SATELLITE SERVER OR CAPSULE SERVER	58
11.3. PERFORMING A BACKUP WITHOUT PULP CONTENT	59
11.4. PERFORMING AN INCREMENTAL BACKUP	60
11.5. EXAMPLE OF A WEEKLY FULL BACKUP FOLLOWED BY DAILY INCREMENTAL BACKUPS	61
11.6. PERFORMING AN ONLINE BACKUP	61
11.7. PERFORMING A SNAPSHOT BACKUP	62
11.8. WHITE-LISTING AND SKIPPING STEPS WHEN PERFORMING BACKUPS	62
CHAPTER 12. RESTORING SATELLITE SERVER OR CAPSULE SERVER FROM A BACKUP	64
12.1. RESTORING FROM A FULL BACKUP	64
12.2. RESTORING FROM INCREMENTAL BACKUPS	65
12.3. BACKUP AND RESTORE CAPSULE SERVER USING A VIRTUAL MACHINE SNAPSHOT	65
12.3.1. Synchronizing an External Capsule	66
CHAPTER 13. RENAMING SATELLITE SERVER OR CAPSULE SERVER	67
13.1. RENAMING SATELLITE SERVER	67
13.2. RENAMING CAPSULE SERVER	68
CHAPTER 14. MAINTAINING SATELLITE SERVER	71
14.1. DELETING AUDIT RECORDS	71
14.2. ANONYMIZING AUDIT RECORDS	71
14.3. DELETING REPORT RECORDS	71
14.4. CONFIGURING THE CLEANING UNUSED TASKS FEATURE	71
14.5. DELETING TASK RECORDS	72
14.6. DELETING A TASK BY ID	72
14.7. RECOVERING FROM A FULL DISK	73
14.8. MANAGING PACKAGES ON THE BASE OPERATING SYSTEM OF SATELLITE SERVER OR CAPSULE SERVER	74
14.9. RECLAIMING POSTGRESQL SPACE	75
CHAPTER 15. RENEWING THE CUSTOM SSL CERTIFICATE	76
15.1. RENEWING A CUSTOM SSL CERTIFICATE ON SATELLITE SERVER	76
15.2. RENEWING A CUSTOM SSL CERTIFICATE ON CAPSULE SERVER	77
CHAPTER 16. LOGGING AND REPORTING PROBLEMS	79
16.1. ENABLING DEBUG LOGGING	79
16.2. INCREASING THE LOGGING LEVELS TO HELP WITH DEBUGGING	79
16.2.1. Increasing the Logging Level For Hammer	80
16.2.2. Increasing the Logging Level On Capsule	80
16.2.3. Increasing the Logging Level For Candlepin	80
16.2.4. Increasing the Logging Level On Satellite	81
16.2.5. Increasing the Logging Level For Qpid Dispatch Router	81
16.2.6. Increasing the Logging Level For Qpid Broker	82
16.2.7. Increasing the Logging Level For Redis	82
16.2.8. Increasing the Logging Level For Postgres	82
16.2.9. Increasing the Logging Level For Satellite Installer	83
16.2.10. Increasing the Logging Level For Pulp	83
16.2.11. Increasing the Logging Level For Puppet Agent	83
16.2.12. Increasing the Logging Level For Puppet Server	83
16.3. RETRIEVING THE STATUS OF SERVICES	84
16.4. RESTARTING SERVICES	84
16.5. ENABLING INDIVIDUAL LOGGERS	85
16.6. CONFIGURING LOGGING TO JOURNAL OR FILE-BASED LOGGING	86

16.7. LOG FILE DIRECTORIES PROVIDED BY SATELLITE	86
16.8. UTILITIES FOR COLLECTING LOG INFORMATION	87
16.9. SYSTEM JOURNAL METADATA	87
CHAPTER 17. MONITORING RESOURCES	90
17.1. USING THE RED HAT SATELLITE CONTENT DASHBOARD	90
17.1.1. Managing Tasks	93
17.2. CONFIGURING RSS NOTIFICATIONS	94
17.3. MONITORING SATELLITE SERVER	94
17.4. MONITORING CAPSULE SERVER	94
17.4.1. Viewing General Capsule Information	94
17.4.2. Monitoring Services	95
17.4.3. Monitoring Puppet	95
CHAPTER 18. USING WEBHOOKS	96
18.1. MIGRATING TO WEBHOOKS	96
18.2. INSTALLING WEBHOOKS	97
18.3. CREATING A WEBHOOK TEMPLATE	97
18.4. CREATING A WEBHOOK	97
18.5. AVAILABLE WEBHOOK EVENTS	98
18.6. SHELLHOOKS	101
18.7. INSTALLING THE SHELLHOOKS PLUGIN	102
18.8. PASSING ARGUMENTS TO SHELLHOOK SCRIPT USING WEBHOOKS	102
18.9. PASSING ARGUMENTS TO SHELLHOOK SCRIPT USING CURL	102
18.10. CREATING A SHELLHOOK TO PRINT ARGUMENTS	103
CHAPTER 19. SEARCHING AND BOOKMARKING	105
19.1. BUILDING SEARCH QUERIES	105
19.1.1. Query Syntax	105
19.1.2. Query Operators	105
19.1.3. Query Values	106
19.2. USING FREE TEXT SEARCH	107
19.3. MANAGING BOOKMARKS	107
19.3.1. Creating Bookmarks	108
19.3.2. Deleting Bookmarks	108
APPENDIX A. ADMINISTRATION SETTINGS	109
A.1. GENERAL SETTINGS	109
A.2. SATELLITE TASK SETTINGS	110
A.3. TEMPLATE SYNC SETTINGS	111
A.4. DISCOVERED SETTINGS	112
A.5. BOOT DISK SETTINGS	114
A.6. RED HAT CLOUD SETTINGS	115
A.7. CONTENT SETTINGS	115
A.8. AUTHENTICATION SETTINGS	119
A.9. EMAIL SETTINGS	122
A.10. NOTIFICATIONS SETTINGS	123
A.11. PROVISIONING SETTINGS	123
A.12. FACTS SETTINGS	127
A.13. CONFIGURATION MANAGEMENT SETTINGS	128
A.14. REMOTE EXECUTION SETTINGS	129
A.15. ANSIBLE SETTINGS	131

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better.

- For simple comments on specific passages:
 1. Ensure you are viewing the documentation in the *Multi-page HTML* format.
In addition, ensure you see the **Feedback** button in the upper right corner of the document.
 2. Use your mouse cursor to highlight the part of text that you want to comment on.
 3. Click the **Add Feedback** pop-up that appears below the highlighted text.
 4. Follow the displayed instructions.
- For submitting feedback via Bugzilla, create a new ticket:
 1. Go to the [Bugzilla](#) website.
 2. As the Component, use **Documentation**.
 3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
 4. Click **Submit Bug**.

CHAPTER 1. ACCESSING RED HAT SATELLITE

After Red Hat Satellite has been installed and configured, use the Satellite web UI interface to log in to Satellite for further configuration.

1.1. IMPORTING THE KATELLO ROOT CA CERTIFICATE

The first time you log in to Satellite, you might see a warning informing you that you are using the default self-signed certificate and you might not be able to connect this browser to Satellite until the root CA certificate is imported in the browser. Use the following procedure to locate the root CA certificate on Satellite and to import it into your browser.

To use the CLI instead of the Satellite web UI, see [CLI Procedure](#).

Prerequisite

- Your Red Hat Satellite is installed and configured.

Procedure

1. Identify the fully qualified domain name of your Satellite Server:

```
# hostname -f
```

2. Access the **pub** directory on your Satellite Server using a web browser pointed to the fully qualified domain name:

```
https://satellite.example.com/pub
```

3. When you access Satellite for the first time, an untrusted connection warning displays in your web browser. Accept the self-signed certificate and add the Satellite URL as a security exception to override the settings. This procedure might differ depending on the browser being used. Ensure that the Satellite URL is valid before you accept the security exception.
4. Select **katello-server-ca.crt**.
5. Import the certificate into your browser as a certificate authority and trust it to identify websites.

CLI Procedure

1. From the Satellite CLI, copy the **katello-server-ca.crt** file to the machine you use to access the Satellite web UI:

```
# scp /var/www/html/pub/katello-server-ca.crt username@hostname:remotefile
```

2. In the browser, import the **katello-server-ca.crt** certificate as a certificate authority and trust it to identify websites.

1.2. LOGGING IN TO SATELLITE

Use the web user interface to log in to Satellite for further configuration.

Prerequisite

- Ensure that the Katello root CA certificate is installed in your browser. For more information, see [Section 1.1, “Importing the Katello Root CA Certificate”](#).

Procedure


1. Access Satellite Server using a web browser pointed to the fully qualified domain name:

`https://satellite.example.com/`

2. Enter the user name and password created during the configuration process. If a user was not created during the configuration process, the default user name is *admin*. If you have problems logging in, you can reset the password. For more information, see [Section 1.5, “Resetting the Administrative User Password”](#).

1.3. NAVIGATION TABS IN THE SATELLITE WEB UI

Use the navigation tabs to browse the Satellite web UI.

Navigation Tabs	Description
Any Context	Clicking this tab changes the organization and location. If no organization or location is selected, the default organization is <i>Any Organization</i> and the default location is <i>Any Location</i> . Use this tab to change to different values.
Monitor	Provides summary dashboards and reports.
Content	Provides content management tools. This includes Content Views, Activation Keys, and Life Cycle Environments.
Hosts	Provides host inventory and provisioning configuration tools.
Configure	Provides general configuration tools and data including Host Groups and Puppet data.
Infrastructure	Provides tools on configuring how Satellite interacts with the environment.
User Name	Provides user administration where users can edit their personal information.
	Provides event notifications to keep administrators informed of important environment changes.
Administer	Provides advanced configuration for settings such as Users and RBAC, as well as general settings.

1.4. CHANGING THE PASSWORD

These steps show how to change your password.

Procedure

1. Click your user name at the top right corner.
2. Select **My Account** from the menu.
3. In the **Current Password** field, enter the current password.
4. In the **Password** field, enter a new password.
5. In the **Verify** field, enter the new password again.
6. Click the **Submit** button to save your new password.

1.5. RESETTING THE ADMINISTRATIVE USER PASSWORD

Use the following procedures to reset the administrative password to randomly generated characters or to set a new administrative password.

To Reset the Administrative User Password

1. Log in to the base operating system where Satellite Server is installed.
2. Enter the following command to reset the password:

```
# foreman-rake permissions:reset
Reset to user: admin, password: qwJxBptxb7Gfcjj5
```

3. Use this password to reset the password in the Satellite web UI.
4. Edit the `~/.hammer/cli.modules.d/foreman.yml` file on Satellite Server to add the new password:

```
# vi ~/.hammer/cli.modules.d/foreman.yml
```

Unless you update the `~/.hammer/cli.modules.d/foreman.yml` file, you cannot use the new password with Hammer CLI.

To Set a New Administrative User Password

1. Log in to the base operating system where Satellite Server is installed.
2. To set the password, enter the following command:

```
# foreman-rake permissions:reset password=new_password
```

3. Edit the `~/.hammer/cli.modules.d/foreman.yml` file on Satellite Server to add the new password:

```
# vi ~/.hammer/cli.modules.d/foreman.yml
```

Unless you update the `~/.hammer/cli.modules.d/foreman.yml` file, you cannot use the new password with Hammer CLI.

1.6. SETTING A CUSTOM MESSAGE ON THE LOGIN PAGE

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **General** tab.
2. Click the edit button next to **Login page footer text**, and enter the desired text to be displayed on the login page. For example, this text may be a warning message required by your company.
3. Click **Save**.
4. Log out of the Satellite web UI and verify that the custom text is now displayed on the login page below the Satellite version number.

CHAPTER 2. STARTING AND STOPPING RED HAT SATELLITE

Satellite provides the **satellite-maintain service** command to manage Satellite services from the command line. This is useful when creating a backup of Satellite. For more information on creating backups, see [Chapter 11, *Backing Up Satellite Server and Capsule Server*](#).

After installing Satellite with the **satellite-installer** command, all Satellite services are started and enabled automatically. View the list of these services by executing:

```
# satellite-maintain service list
```

To see the status of running services, execute:

```
# satellite-maintain service status
```

To stop Satellite services, execute:

```
# satellite-maintain service stop
```

To start Satellite services, execute:

```
# satellite-maintain service start
```

To restart Satellite services, execute:

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

CHAPTER 3. MIGRATING FROM INTERNAL SATELLITE DATABASES TO EXTERNAL DATABASES

When you install Red Hat Satellite, the **satellite-installer** command installs PostgreSQL databases on the same server as Satellite. If you are using the default internal databases but want to start using external databases to help with the server load, you can migrate your internal databases to external databases.

To confirm whether your Satellite Server has internal or external databases, you can query the status of your databases:

For PostgreSQL, enter the following command:

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

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 migrate from the default internal databases to external databases, you must complete the following procedures:

1. [Section 3.2, “Preparing a Host for External Databases”](#). Prepare a Red Hat Enterprise Linux 8 server to host the external databases.
2. [Section 3.3, “Installing PostgreSQL”](#). Prepare PostgreSQL with databases for Satellite, Pulp and Candlepin with dedicated users owning them.
3. [Section 3.4, “Migrating to External Databases”](#). Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**.

3.1. 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.

3.2. 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](#).

3.3. 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"
```

3.4. MIGRATING TO EXTERNAL DATABASES

Back up and transfer existing data, then use the **satellite-installer** command to configure Satellite to connect to an external PostgreSQL database server.

Prerequisite

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

Procedure

1. On Satellite Server, stop Satellite services:

```
# satellite-maintain service stop
```

2. Start the **PostgreSQL** services:

```
# systemctl start postgresql
```

3. Back up the internal databases:

```
# satellite-maintain backup online --skip-pulp-content --preserve-directory -y
/var/migration_backup
```

4. Transfer the data to the new external databases:

```
PGPASSWORD='Foreman_Password' pg_restore -h postgres.example.com -U foreman -d
foreman < /var/migration_backup/foreman.dump
```

```
PGPASSWORD='Candlepin_Password' pg_restore -h postgres.example.com -U candlepin -d candlepin < /var/migration_backup/candlepin.dump
PGPASSWORD='Pulpcore_Password' pg_restore -h postgres.example.com -U pulp -d pulpcore < /var/migration_backup/pulpcore.dump
```

5. Use the **satellite-installer** command to update Satellite to point to the new databases:

```
satellite-installer --scenario satellite \
  --foreman-db-host postgres.example.com \
  --foreman-db-password Foreman_Password \
  --foreman-db-database foreman \
  --foreman-db-manage false \
  --foreman-db-username foreman \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password Candlepin_Password \
  --katello-candlepin-manage-db false \
  --katello-candlepin-db-user candlepin \
  --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
```

CHAPTER 4. MANAGING SATELLITE WITH ANSIBLE COLLECTIONS

Satellite Ansible Collections is a set of Ansible modules that interact with the Satellite API. You can use Satellite Ansible Collections to manage and automate many aspects of Satellite.

4.1. INSTALLING THE SATELLITE ANSIBLE MODULES

Use this procedure to install the Satellite Ansible modules.

Procedure

- Install the package using the following command:

```
# satellite-maintain packages install ansible-collection-redhat-satellite
```

4.2. VIEWING THE SATELLITE ANSIBLE MODULES

You can view the installed Satellite Ansible modules by running:

```
# ansible-doc -l redhat.satellite
```

Alternatively, you can also see the complete list of Satellite Ansible modules and other related information at <https://console.redhat.com/ansible/automation-hub/redhat/satellite/docs>.

All modules are in the **redhat.satellite** namespace and can be referred to in the format **redhat.satellite._module_name_**. For example, to display information about the **activation_key** module, enter the following command:

```
$ ansible-doc redhat.satellite.activation_key
```

CHAPTER 5. MANAGING ORGANIZATIONS

Organizations divide Red Hat Satellite resources into logical groups based on ownership, purpose, content, security level, or other divisions. You can create and manage multiple organizations through Red Hat Satellite, then divide and assign your Red Hat subscriptions to each individual organization. This provides a method of managing the content of several individual organizations under one management system. Here are some examples of organization management:

Single Organization

A small business with a simple system administration chain. In this case, you can create a single organization for the business and assign content to it.

Multiple Organizations

A large company that owns several smaller business units. For example, a company with separate system administration and software development groups. In this case, you can create organizations for the company and each of the business units it owns. This keeps the system infrastructure for each separate. You can then assign content to each organization based on its needs.

External Organizations

A company that manages external systems for other organizations. For example, a company offering cloud computing and web hosting resources to customers. In this case, you can create an organization for the company's own system infrastructure and then an organization for each external business. You can then assign content to each organization where necessary.

A default installation of Red Hat Satellite has a default organization called **Default Organization**.

New Users

If a new user is not assigned a default organization, their access is limited. To grant systems rights to users, assign them to a default organization. The next time the user logs on to Satellite, the user's account has the correct system rights.

5.1. CREATING AN ORGANIZATION

Use this procedure to create an organization. To use the CLI instead of the Satellite web UI, see [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Administer > Organizations**.
2. Click **New Organization**.
3. In the **Name** field, enter a name for the organization.
4. In the **Label** field, enter a unique identifier for the organization. This is used for creating and mapping certain assets, such as directories for content storage. Use letters, numbers, underscores, and dashes, but no spaces.
5. Optional: in the **Description** field, enter a description for the organization.
6. Click **Submit**.
7. If you have hosts with no organization assigned, select the hosts that you want to add to the organization, then click **Proceed to Edit**

8. In the **Edit** page, assign the infrastructure resources that you want to add to the organization. This includes networking resources, installation media, kickstart templates, and other parameters. You can return to this page at any time by navigating to **Administer > Organizations** and then selecting an organization to edit.
9. Click **Submit**.

CLI procedure

1. To create an organization, enter the following command:

```
# hammer organization create \
--name "My_Organization" \
--label "My_Organization_Label" \
--description "My_Organization_Description"
```

2. Optional: To edit an organization, enter the **hammer organization update** command. For example, the following command assigns a compute resource to the organization:

```
# hammer organization update \
--name "My_Organization" \
--compute-resource-ids 1
```

5.2. SETTING THE ORGANIZATION CONTEXT

An organization context defines the organization to use for a host and its associated resources.

Procedure

The organization menu is the first menu item in the menu bar, on the upper left of the Satellite web UI. If you have not selected a current organization, the menu says **Any Organization**. Click the **Any Organization** button and select the organization to use.

CLI procedure

While using the CLI, include either **--organization "My_Organization"** or **--organization-label "My_Organization_Label"** as an option. For example:

```
# hammer subscription list \
--organization "My_Organization"
```

This command outputs subscriptions allocated for the *My_Organization*.

5.3. CREATING AN ORGANIZATION DEBUG CERTIFICATE

If you require a debug certificate for your organization, use the following procedure.

Procedure

1. In the Satellite web UI, navigate to **Administer > Organizations**.
2. Select an organization that you want to generate a debug certificate for.
3. Click **Generate and Download**.

4. Save the certificate file in a secure location.

Debug Certificates for Provisioning Templates

Debug Certificates are automatically generated for provisioning template downloads if they do not already exist in the organization for which they are being downloaded.

5.4. BROWSING REPOSITORY CONTENT USING AN ORGANIZATION DEBUG CERTIFICATE

You can view an organization's repository content using a web browser or using the API if you have a debug certificate for that organization.

Prerequisite

- You created and downloaded an organization certificate. For more information, see [Section 5.3, "Creating an Organization Debug Certificate"](#).

Procedure

1. Split the private and public keys from the certificate into two files.
 - a. Open the X.509 certificate, for example, for the default organization:

```
$ vi 'Default Organization-key-cert.pem'
```
 - b. Copy the contents of the file from **-----BEGIN RSA PRIVATE KEY-----** to **-----END RSA PRIVATE KEY-----**, into a **key.pem** file.
 - c. Copy the contents of the file from **-----BEGIN CERTIFICATE-----** to **-----END CERTIFICATE-----**, into a **cert.pem** file.
2. To use a browser, you must first convert the X.509 certificate to a format your browser supports and then import the certificate.

For Firefox Users

1. Convert the certificate into the PKCS12 format using the following command:

```
$ openssl pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-3DES -export -in cert.pem  
-inkey key.pem -out My_Organization_Label.pfx -name My_Organization
```

2. In the Firefox browser, navigate to **Edit > Preferences > Advanced Tab**.
3. Select **View Certificates** and click the **Your Certificates** tab.
4. Click **Import** and select the **.pfx** file to load.
5. Enter the following URL in the address bar to browse the accessible paths for all the repositories and check their contents:

```
https://satellite.example.com/pulp/content/
```

For CURL Users

- To use the organization debug certificate with CURL, enter the following command:

```
$ curl -k --cert cert.pem --key key.pem \
https://satellite.example.com/pulp/content/My_Organization_Label/Library/content/dist/rhel/server/7/7Server/x86_64/os/
```

Ensure that the paths to **cert.pem** and **key.pem** are the correct absolute paths otherwise the command fails silently. Pulp uses the organization label, therefore, you must enter the organization label into the URL.

5.5. DELETING AN ORGANIZATION

You can delete an organization if the organization is not associated with any life cycle environments or host groups. If there are any life cycle environments or host groups associated with the organization you are about to delete, remove them by navigating to **Administer > Organizations** and clicking the relevant organization. Do not delete the default organization created during installation because the default organization is a placeholder for any unassociated hosts in the Satellite environment. There must be at least one organization in the environment at any given time.

Procedure

1. In the Satellite web UI, navigate to **Administer > Organizations**.
2. From the list to the right of the name of the organization you want to delete, select **Delete**.
3. Click **OK** to delete the organization.

CLI procedure

1. Enter the following command to retrieve the ID of the organization that you want to delete:

```
# hammer organization list
```

From the output, note the ID of the organization that you want to delete.

2. Enter the following command to delete an organization:

```
# hammer organization delete --id Organization_ID
```

CHAPTER 6. MANAGING LOCATIONS

Locations function similar to organizations: they provide a method to group resources and assign hosts. Organizations and locations have the following conceptual differences:

- Locations are based on physical or geographical settings.
- Locations have a hierarchical structure.

6.1. CREATING A LOCATION

Use this procedure to create a location so that you can manage your hosts and resources by location. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Administer > Locations**.
2. Click **New Location**.
3. Optional: from the **Parent** list, select a parent location. This creates a location hierarchy.
4. In the **Name** field, enter a name for the location.
5. Optional: in the **Description** field, enter a description for the location.
6. Click **Submit**.
7. If you have hosts with no location assigned, add any hosts that you want to assign to the new location, then click **Proceed to Edit**.
8. Assign any infrastructure resources that you want to add to the location. This includes networking resources, installation media, kickstart templates, and other parameters. You can return to this page at any time by navigating to **Administer > Locations** and then selecting a location to edit.
9. Click **Submit** to save your changes.

CLI procedure

- Enter the following command to create a location:

```
# hammer location create \
--description "My_Location_Description" \
--name "My_Location" \
--parent-id "My_Location_Parent_ID"
```

6.2. CREATING MULTIPLE LOCATIONS

The following example Bash script creates three locations – London, Munich, Boston – and assigns them to the Example Organization.

```
ORG="Example Organization"
LOCATIONS="London Munich Boston"
```

```

for LOC in ${LOCATIONS}
do
  hammer location create --name "${LOC}"
  hammer location add-organization --name "${LOC}" --organization "${ORG}"
done

```

6.3. SETTING THE LOCATION CONTEXT

A location context defines the location to use for a host and its associated resources.

Procedure

The location menu is the second menu item in the menu bar, on the upper left of the Satellite web UI. If you have not selected a current location, the menu displays **Any Location**. Click **Any location** and select the location to use.

CLI procedure

While using the CLI, include either **--location "My_Location"** or **--location-id "My_Location_ID"** as an option. For example:

```
# hammer host list --location "My_Location"
```

This command lists hosts associated with the *My_Location* location.

6.4. DELETING A LOCATION

You can delete a location if the location is not associated with any life cycle environments or host groups. If there are any life cycle environments or host groups associated with the location you are about to delete, remove them by navigating to **Administer > Locations** and clicking the relevant location. Do not delete the default location created during installation because the default location is a placeholder for any unassociated hosts in the Satellite environment. There must be at least one location in the environment at any given time.

Procedure

1. In the Satellite web UI, navigate to **Administer > Locations**.
2. Select **Delete** from the list to the right of the name of the location you want to delete.
3. Click **OK** to delete the location.

CLI procedure

1. Enter the following command to retrieve the ID of the location that you want to delete:

```
# hammer location list
```

From the output, note the ID of the location that you want to delete.

2. Enter the following command to delete the location:

```
# hammer location delete --id Location ID
```

CHAPTER 7. MANAGING USERS AND ROLES

A User defines a set of details for individuals using the system. Users can be associated with organizations and environments, so that when they create new entities, the default settings are automatically used. Users can also have one or more *roles* attached, which grants them rights to view and manage organizations and environments. See [Section 7.1, “Managing Users”](#) for more information on working with users.

You can manage permissions of several users at once by organizing them into user groups. User groups themselves can be further grouped to create a hierarchy of permissions. For more information on creating user groups, see [Section 7.4, “Creating and Managing User Groups”](#).

Roles define a set of permissions and access levels. Each role contains one or more *permission filters* that specify the actions allowed for the role. Actions are grouped according to the *Resource type*. Once a role has been created, users and user groups can be associated with that role. This way, you can assign the same set of permissions to large groups of users. Satellite provides a set of predefined roles and also enables creating custom roles and permission filters as described in [Section 7.5, “Creating and Managing Roles”](#).

7.1. MANAGING USERS

As an administrator, you can create, modify and remove Satellite users. You can also configure access permissions for a user or a group of users by assigning them different *roles*.

7.1.1. Creating a User

Use this procedure to create a user. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Users**.
2. Click **Create User**.
3. In the **Login** field, enter a username for the user.
4. In the **Firstname** and **Lastname** fields, enter the real first name and last name of the user.
5. In the **Mail** field, enter the user’s email address.
6. In the **Description** field, add a description of the new user.
7. Select a specific language for the user from the **Language** list.
8. Select a timezone for the user from the **Timezone** list.
By default, Satellite Server uses the language and timezone settings of the user’s browser.
9. Set a password for the user:
 - a. From the **Authorized by** list, select the source by which the user is authenticated.
 - **INTERNAL**: to enable the user to be managed inside Satellite Server.
 - **EXTERNAL**: to configure external authentication as described in [Configuring External Authentication](#) in *Installing Satellite Server in a Connected Network Environment*.

- b. Enter an initial password for the user in the **Password** field and the **Verify** field.
10. Click **Submit** to create the user.

CLI procedure

- To create a user, enter the following command:

```
# hammer user create \
--auth-source-id My_Authentication_Source \
--login My_User_Name \
--mail My_User_Mail \
--organization-ids My_Organization_ID_1,My_Organization_ID_2 \
--password My_User_Password
```

The **--auth-source-id 1** setting means that the user is authenticated internally, you can specify an external authentication source as an alternative. Add the **--admin** option to grant administrator privileges to the user. Specifying organization IDs is not required, you can modify the user details later using the **update** subcommand.

For more information about user related subcommands, enter **hammer user --help**.

7.1.2. Assigning Roles to a User

Use this procedure to assign roles to a user. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. Click the **username** of the user to be assigned one or more roles.



NOTE

If a user account is not listed, check that you are currently viewing the correct organization. To list all the users in Satellite, click **Default Organization** and then **Any Organization**.

3. Click the **Locations** tab, and select a location if none is assigned.
4. Click the **Organizations** tab, and check that an organization is assigned.
5. Click the **Roles** tab to display the list of available roles.
6. Select the roles to assign from the **Roles** list.
To grant all the available permissions, select the **Admin** checkbox.
7. Click **Submit**.

To view the roles assigned to a user, click the **Roles** tab; the assigned roles are listed under **Selected items**. To remove an assigned role, click the role name in **Selected items**.

CLI procedure

- To assign roles to a user, enter the following command:

```
# hammer user add-role --id user_id --role role_name
```

7.1.3. Impersonating a Different User Account

Administrators can impersonate other authenticated users for testing and troubleshooting purposes by temporarily logging on to the Satellite web UI as a different user. When impersonating another user, the administrator has permissions to access exactly what the impersonated user can access in the system, including the same menus.

Audits are created to record the actions that the administrator performs while impersonating another user. However, all actions that an administrator performs while impersonating another user are recorded as having been performed by the impersonated user.

Prerequisites

- Ensure that you are logged on to the Satellite web UI as a user with administrator privileges for Satellite.

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. To the right of the user that you want to impersonate, from the list in the **Actions** column, select **Impersonate**.

When you want to stop the impersonation session, in the upper right of the main menu, click the impersonation icon.

7.1.4. Creating an API-Only User

You can create users that can interact only with the Satellite API.

Prerequisite

- You have created a user and assigned roles to them. Note that this user must be authorized internally. For more information, see [Creating a User](#) and [Assigning Roles to a User](#).

Procedure

1. Log in to your Satellite as admin.
2. Navigate to **Administer > Users** and select a user.
3. On the **User** tab, set a password. Do not save or communicate this password with others. You can create pseudo-random strings on your console:

```
# openssl rand -hex 32
```

4. Create a Personal Access Token for the user. For more information, see [Section 7.3.1, "Creating a Personal Access Token"](#).

7.2. MANAGING SSH KEYS

Adding SSH keys to a user allows deployment of SSH keys during provisioning. For information on deploying SSH keys during provisioning, see [Deploying SSH Keys during Provisioning](#) in *Provisioning Hosts*.

For information on SSH keys and SSH key creation, see [Using SSH-based Authentication](#) in *Red Hat Enterprise Linux 8 Configuring basic system settings*.

7.2.1. Managing SSH Keys for a User

Use this procedure to add or remove SSH keys for a user. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Prerequisites

- Ensure that you are logged in to the Satellite web UI as an Admin user of Red Hat Satellite or a user with the `create_ssh_key` permission enabled for adding SSH key and `destroy_ssh_key` permission for removing a key.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Users**.
2. From the **Username** column, click on the username of the required user.
3. Click on the **SSH Keys** tab.
 - To Add SSH key
 - i. Prepare the content of the public SSH key in a clipboard.
 - ii. Click **Add SSH Key**.
 - iii. In the **Key** field, paste the public SSH key content from the clipboard.
 - iv. In the **Name** field, enter a name for the SSH key.
 - v. Click **Submit**.
 - To Remove SSH key
 - i. Click **Delete** on the row of the SSH key to be deleted.
 - ii. Click **OK** in the confirmation prompt.

CLI procedure

To add an SSH key to a user, you must specify either the path to the public SSH key file, or the content of the public SSH key copied to the clipboard.

- If you have the public SSH key file, enter the following command:

```
# hammer user ssh-keys add \
--user-id user_id \
--name key_name \
--key-file ~/.ssh/id_rsa.pub
```

- If you have the content of the public SSH key, enter the following command:

```
# hammer user ssh-keys add \
--user-id user_id \
--name key_name \
--key ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTltbmlzdHAyNTYAAAAIbmlzdHAyNtYAAABBBHHS2KmNyIYa27Qaa7
EHp+2l99ucGStx4P77e03ZvE3yVRJEFikpoP3MJtYYfle8k 1/46MTIZo9CPTX4CYUHeN8=
host@user
```

To delete an SSH key from a user, enter the following command:

```
# hammer user ssh-keys delete --id key_id --user-id user_id
```

To view an SSH key attached to a user, enter the following command:

```
# hammer user ssh-keys info --id key_id --user-id user_id
```

To list SSH keys attached to a user, enter the following command:

```
# hammer user ssh-keys list --user-id user_id
```

7.3. MANAGING PERSONAL ACCESS TOKENS

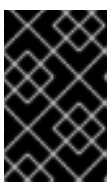
Personal Access Tokens allow you to authenticate API requests without using your password. You can set an expiration date for your Personal Access Token and you can revoke it if you decide it should expire before the expiration date.

7.3.1. Creating a Personal Access Token

Use this procedure to create a Personal Access Token.

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. Select a user for which you want to create a Personal Access Token.
3. On the **Personal Access Tokens** tab, click **Add Personal Access Token**.
4. Enter a **Name** for your Personal Access Token.
5. Optional: Select the **Expires** date to set an expiration date. If you do not set an expiration date, your Personal Access Token will never expire unless revoked.
6. Click Submit. You now have the Personal Access Token available to you on the **Personal Access Tokens** tab.



IMPORTANT

Ensure to store your Personal Access Token as you will not be able to access it again after you leave the page or create a new Personal Access Token. You can click **Copy to clipboard** to copy your Personal Access Token.

Verification

1. Make an API request to your Satellite Server and authenticate with your Personal Access Token:

```
# curl https://satellite.example.com/api/status --user
My_Username:My_Personal_Access_Token
```

2. You should receive a response with status **200**, for example:

```
{"satellite_version":"6.12.0","result":"ok","status":200,"version":"3.5.1.10","api_version":2}
```

If you go back to **Personal Access Tokens** tab, you can see the updated **Last Used** time next to your Personal Access Token.

7.3.2. Revoking a Personal Access Token

Use this procedure to revoke a Personal Access Token before its expiration date.

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. Select a user for which you want to revoke the Personal Access Token.
3. On the **Personal Access Tokens** tab, locate the Personal Access Token you want to revoke.
4. Click **Revoke** in the **Actions** column next to the Personal Access Token you want to revoke.

Verification

1. Make an API request to your Satellite Server and try to authenticate with the revoked Personal Access Token:

```
# curl https://satellite.example.com/api/status --user
My_Username:My_Personal_Access_Token
```

2. You receive the following error message:

```
{
  "error": {"message":"Unable to authenticate user My_Username"}
}
```

7.4. CREATING AND MANAGING USER GROUPS

7.4.1. User Groups

With Satellite, you can assign permissions to groups of users. You can also create user groups as collections of other user groups. If using an external authentication source, you can map Satellite user groups to external user groups as described in [Configuring External User Groups](#) in *Installing Satellite Server in a Connected Network Environment*.

User groups are defined in an organizational context, meaning that you must select an organization before you can access user groups.

7.4.2. Creating a User Group

Use this procedure to create a user group.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **User Groups**.
2. Click **Create User group**.
3. On the **User Group** tab, specify the name of the new user group and select group members:
 - Select the previously created user groups from the **User Groups** list.
 - Select users from the **Users** list.
4. On the **Roles** tab, select the roles you want to assign to the user group. Alternatively, select the **Admin** checkbox to assign all available permissions.
5. Click **Submit**.

CLI procedure

- To create a user group, enter the following command:

```
# hammer user-group create \  
--name My_User_Group_Name \  
--role-ids My_Role_ID_1,My_Role_ID_2 \  
--user-ids My_User_ID_1,My_User_ID_2
```

7.4.3. Removing a User Group

Use the Satellite web UI to remove a user group.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **User Groups**.
2. Click **Delete** to the right of the user group you want to delete.
3. In the alert box that appears, click **OK** to delete a user group.

7.5. CREATING AND MANAGING ROLES

Satellite provides a set of predefined roles with permissions sufficient for standard tasks, as listed in [Section 7.6, "Predefined Roles Available in Satellite"](#). It is also possible to configure custom roles, and assign one or more permission filters to them. Permission filters define the actions allowed for a certain resource type. Certain Satellite plug-ins create roles automatically.

7.5.1. Creating a Role

Use this procedure to create a role.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Roles**.
2. Click **Create Role**.
3. Provide a **Name** for the role.
4. Click **Submit** to save your new role.

CLI procedure

- To create a role, enter the following command:

```
# hammer role create --name My_Role_Name
```

To serve its purpose, a role must contain permissions. After creating a role, proceed to [Section 7.5.3, “Adding Permissions to a Role”](#).

7.5.2. Cloning a Role

Use the Satellite web UI to clone a role.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Roles** and select **Clone** from the drop-down menu to the right of the required role.
2. Provide a **Name** for the role.
3. Click **Submit** to clone the role.
4. Click the name of the cloned role and navigate to **Filters**.
5. Edit the permissions as required.
6. Click **Submit** to save your new role.

7.5.3. Adding Permissions to a Role

Use this procedure to add permissions to a role. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Roles**.
2. Select **Add Filter** from the drop-down list to the right of the required role.
3. Select the **Resource type** from the drop-down list. The *(Miscellaneous)* group gathers permissions that are not associated with any resource group.
4. Click the permissions you want to select from the **Permission** list.
5. Depending on the **Resource type** selected, you can select or deselect the **Unlimited** and **Override** checkbox. The **Unlimited** checkbox is selected by default, which means that the permission is applied on all resources of the selected type. When you disable the **Unlimited**

checkbox, the **Search** field activates. In this field you can specify further filtering with use of the Satellite search syntax. For more information, see [Section 7.7, “Granular Permission Filtering”](#). When you enable the **Override** checkbox, you can add additional locations and organizations to allow the role to access the resource type in the additional locations and organizations; you can also remove an already associated location and organization from the resource type to restrict access.

6. Click **Next**.
7. Click **Submit** to save changes.

CLI procedure

1. List all available permissions:

```
# hammer filter available-permissions
```

2. Add permissions to a role:

```
# hammer filter create \
--permission-ids My_Permission_ID_1,My_Permission_ID_2 \
--role My_Role_Name
```

For more information about roles and permissions parameters, enter the **hammer role --help** and **hammer filter --help** commands.

7.5.4. Viewing Permissions of a Role

Use the Satellite web UI to view the permissions of a role.

Procedure

1. In the Satellite web UI, navigate to **Administer > Roles**.
2. Click **Filters** to the right of the required role to get to the **Filters** page.

The **Filters** page contains a table of permissions assigned to a role grouped by the resource type. It is also possible to generate a complete table of permissions and actions that you can use on your Satellite system. For more information, see [Section 7.5.5, “Creating a Complete Permission Table”](#).

7.5.5. Creating a Complete Permission Table

Use the Satellite CLI to create a permission table.

Procedure

1. Ensure that the required packages are installed. Execute the following command on Satellite Server:

```
# satellite-maintain packages install foreman-console
```

2. Start the Satellite console with the following command:

```
# foreman-rake console
```

Insert the following code into the console:

```
f = File.open('/tmp/table.html', 'w')

result = Foreman::AccessControl.permissions {|a,b| a.security_block <=>
  b.security_block}.collect do |p|
  actions = p.actions.collect { |a| "<li>#{a}</li>" }
  "<tr><td>#{p.name}</td><td><ul>#{actions.join("</td><td>#{p.resource_type}</td>
</tr>"
end.join("\n")

f.write(result)
```

The above syntax creates a table of permissions and saves it to the **/tmp/table.html** file.

3. Press **Ctrl + D** to exit the Satellite console. Insert the following text at the first line of **/tmp/table.html**:

```
<table border="1"><tr><td>Permission name</td><td>Actions</td><td>Resource type</td>
</tr>
```

Append the following text at the end of **/tmp/table.html**:

```
</table>
```

4. Open **/tmp/table.html** in a web browser to view the table.

7.5.6. Removing a Role

Use the Satellite web UI to remove a role.

Procedure

1. In the Satellite web UI, navigate to **Administer > Roles**.
2. Select **Delete** from the drop-down list to the right of the role to be deleted.
3. In an alert box that appears, click **OK** to delete the role.

7.6. PREDEFINED ROLES AVAILABLE IN SATELLITE

The following table provides an overview of permissions that predefined roles in Satellite grant to a user.

To view the exact set of permissions a predefined role grants, display the role in Satellite web UI as the privileged user. For more information, see [Section 7.5.4, "Viewing Permissions of a Role"](#).

Table 7.1. Permissions provided by role

Role	Permissions Provided by Role
Access Insights Admin	Add and edit Insights rules.

Role	Permissions Provided by Role
Access Insights Viewer	View Insight reports.
Ansible Roles Manager	Play roles on hosts and host groups. View, destroy, and import Ansible roles. View, edit, create, destroy, and import Ansible variables.
Ansible Tower Inventory Reader	View facts, hosts, and host groups.
Bookmarks manager	Create, edit, and delete bookmarks.
Boot disk access	Download the boot disk.
Compliance manager	View, create, edit, and destroy SCAP content files, compliance policies, and tailoring files. View compliance reports.
Compliance viewer	View compliance reports.
Create ARF report	Create compliance reports.
Default role	The set of permissions that every user is granted, irrespective of any other roles.
Discovery Manager	View, provision, edit, and destroy discovered hosts and manage discovery rules.
Discovery Reader	View hosts and discovery rules.
Edit hosts	View, create, edit, destroy, and build hosts.
Edit partition tables	View, create, edit and destroy partition tables.
Manager	View and edit global settings.
Organization admin	An administrator role defined per organization. The role has no visibility into resources in other organizations.
Red Hat Access Logs	View the log viewer and the logs.
Remote Execution Manager	Control which roles have permission to run infrastructure jobs.
Remote Execution User	Run remote execution jobs against hosts.
Site manager	A restrained version of the Manager role.

Role	Permissions Provided by Role
System admin	<ul style="list-style-type: none"> • Edit global settings in Administer > Settings. • View, create, edit and destroy users, user groups, and roles. • View, create, edit, destroy, and assign organizations and locations but not view resources within them. <p>Users with this role can create users and assign all roles to them. Therefore, ensure to give this role only to trusted users.</p>
Tasks manager	View and edit Satellite tasks.
Tasks reader	A role that can only view Satellite tasks.
Viewer	A passive role that provides the ability to view the configuration of every element of the Satellite structure, logs, reports, and statistics.
View hosts	A role that can only view hosts.
Virt-who Manager	A role with full virt-who permissions.
Virt-who Reporter	Upload reports generated by virt-who to Satellite. It can be used if you configure virt-who manually and require a user role that has limited virt-who permissions.
Virt-who Viewer	View virt-who configurations. Users with this role can deploy virt-who instances using existing virt-who configurations.

7.7. GRANULAR PERMISSION FILTERING

As mentioned in [Section 7.5.3, "Adding Permissions to a Role"](#), Red Hat Satellite provides the ability to limit the configured user permissions to selected instances of a resource type. These granular filters are queries to the Satellite database and are supported by the majority of resource types.

7.7.1. Creating a Granular Permission Filter

Use this procedure to create a granular filter. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Satellite does not apply search conditions to create actions. For example, limiting the `create_locations` action with `name = "Default Location"` expression in the search field does not prevent the user from assigning a custom name to the newly created location.

Procedure

Specify a query in the **Search** field on the **Edit Filter** page. Deselect the **Unlimited** checkbox for the field to be active. Queries have the following form:

```
field_name operator value
```

- *field_name* marks the field to be queried. The range of available field names depends on the resource type. For example, the *Partition Table* resource type offers *family*, *layout*, and *name* as query parameters.
- *operator* specifies the type of comparison between *field_name* and *value*. See [Section 7.7.3, “Supported Operators for Granular Search”](#) for an overview of applicable operators.
- *value* is the value used for filtering. This can be for example a name of an organization. Two types of wildcard characters are supported: underscore (_) provides single character replacement, while percent sign (%) replaces zero or more characters.

For most resource types, the **Search** field provides a drop-down list suggesting the available parameters. This list appears after placing the cursor in the search field. For many resource types, you can combine queries using logical operators such as *and*, *not* and *has* operators.

CLI procedure

- To create a granular filter, enter the **hammer filter create** command with the **--search** option to limit permission filters, for example:

```
# hammer filter create \
--permission-ids 91 \
--search "name ~ ccv*" \
--role qa-user
```

This command adds to the **qa-user** role a permission to view, create, edit, and destroy Content Views that only applies to Content Views with name starting with **ccv**.

7.7.2. Examples of Using Granular Permission Filters

As an administrator, you can allow selected users to make changes in a certain part of the environment path. The following filter allows you to work with content while it is in the development stage of the application life cycle, but the content becomes inaccessible once is pushed to production.

7.7.2.1. Applying Permissions for the Host Resource Type

The following query applies any permissions specified for the Host resource type only to hosts in the group named host-editors.

```
hostgroup = host-editors
```

The following query returns records where the name matches *XXXX*, *Yyyy*, or *zzzz* example strings:

```
name ^ (XXXX, Yyyy, zzzz)
```

You can also limit permissions to a selected environment. To do so, specify the environment name in the **Search** field, for example:

```
Dev
```

You can limit user permissions to a certain organization or location with the use of the granular permission filter in the **Search** field. However, some resource types provide a GUI alternative, an **Override** checkbox that provides the **Locations** and **Organizations** tabs. On these tabs, you can select

from the list of available organizations and locations. For more information, see [Section 7.7.2.2, “Creating an Organization Specific Manager Role”](#).

7.7.2.2. Creating an Organization Specific Manager Role

Use the Satellite web UI to create an administrative role restricted to a single organization named *org-1*.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Roles**.
2. Clone the existing **Organization admin** role. Select **Clone** from the drop-down list next to the **Filters** button. You are then prompted to insert a name for the cloned role, for example *org-1 admin*.
3. Click the desired locations and organizations to associate them with the role.
4. Click **Submit** to create the role.
5. Click *org-1 admin*, and click **Filters** to view all associated filters. The default filters work for most use cases. However, you can optionally click **Edit** to change the properties for each filter. For some filters, you can enable the **Override** option if you want the role to be able to access resources in additional locations and organizations. For example, by selecting the **Domain** resource type, the **Override** option, and then additional locations and organizations using the **Locations** and **Organizations** tabs, you allow this role to access domains in the additional locations and organizations that is not associated with this role. You can also click **New filter** to associate new filters with this role.

7.7.3. Supported Operators for Granular Search

Table 7.2. Logical Operators

Operator	Description
and	Combines search criteria.
not	Negates an expression.
has	Object must have a specified property.

Table 7.3. Symbolic Operators

Operator	Description
=	<i>Is equal to</i> . An equality comparison that is case-sensitive for text fields.
!=	<i>Is not equal to</i> . An inversion of the = operator.
~	<i>Like</i> . A case-insensitive occurrence search for text fields.
!~	<i>Not like</i> . An inversion of the ~ operator.

^	<i>In</i> . An equality comparison that is case-sensitive search for text fields. This generates a different SQL query to the <i>Is equal to</i> comparison, and is more efficient for multiple value comparison.
!^	<i>Not in</i> . An inversion of the ^ operator.
>, >=	<i>Greater than, greater than or equal to</i> . Supported for numerical fields only.
<, <=	<i>Less than, less than or equal to</i> . Supported for numerical fields only.

CHAPTER 8. CONFIGURING EMAIL NOTIFICATIONS

Email notifications are created by Satellite Server periodically or after completion of certain events. The periodic notifications can be sent daily, weekly or monthly.

For the events that trigger a notification and notification types, see [Section 8.1, “Notification Types”](#).

Users do not receive any email notifications by default. An administrator can configure users to receive notifications based on criteria such as the type of notification, and frequency.



IMPORTANT

Satellite Server does not enable outgoing emails by default, therefore you must review your email configuration. For more information, see [Configuring Satellite Server for Outgoing Emails](#) in *Installing Satellite Server in a Connected Network Environment*.

8.1. NOTIFICATION TYPES

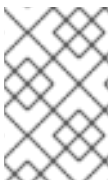
The following are the notifications created by Satellite:

- **Audit summary:** A summary of all activity audited by Satellite Server.
- **Host built:** A notification sent when a host is built.
- **Host errata advisory:** A summary of applicable and installable errata for hosts managed by the user.
- **OpenSCAP policy summary:** A summary of OpenSCAP policy reports and their results.
- **Promote errata:** A notification sent only after a Content View promotion. It contains a summary of errata applicable and installable to hosts registered to the promoted Content View. This allows a user to monitor what updates have been applied to which hosts.
- **Puppet error state:** A notification sent after a host reports an error related to Puppet.
- **Puppet summary:** A summary of Puppet reports.
- **Sync errata:** A notification sent only after synchronizing a repository. It contains a summary of new errata introduced by the synchronization.

8.2. CONFIGURING EMAIL NOTIFICATION PREFERENCES

You can configure Satellite to send email messages to individual users registered to Satellite. Satellite sends the email to the email address that has been added to the account, if present. Users can edit the email address by clicking on their name in the top-right of the Satellite web UI and selecting **My account**.

Configure email notifications for a user from the Satellite web UI.



NOTE

If you want to send email notifications to a group email address instead of an individual email address, create a user account with the group email address and minimal Satellite permissions, then subscribe the user account to the desired notification types.

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. Click the **Username** of the user you want to edit.
3. On the **User** tab, verify the value of the **Mail** field. Email notifications will be sent to the address in this field.
4. On the **Email Preferences** tab, select **Mail Enabled**.
5. Select the notifications you want the user to receive using the drop-down menus next to the notification types.



NOTE

The **Audit Summary** notification can be filtered by entering the required query in the **Mail Query** text box.

6. Click **Submit**.
The user will start receiving the notification emails.

8.3. TESTING EMAIL DELIVERY

To verify the delivery of emails, send a test email to a user. If the email gets delivered, the settings are correct.

Procedure

1. In the Satellite web UI, navigate to **Administer > Users**.
2. Click on the username.
3. On the **Email Preferences** tab, click **Test email**.
A test email message is sent immediately to the user's email address.

If the email is delivered, the verification is complete. Otherwise, you must perform the following diagnostic steps:

- a. Verify the user's email address.
- b. Verify Satellite Server's email configuration.
- c. Examine firewall and mail server logs.

8.4. TESTING EMAIL NOTIFICATIONS

To verify that users are correctly subscribed to notifications, trigger the notifications manually.

Procedure

- To trigger the notifications, execute the following command:

```
# foreman-rake reports:_My_Frequency_
```

Replace *My_Frequency* with one of the following:

- daily
- weekly
- monthly

This triggers all notifications scheduled for the specified frequency for all the subscribed users. If every subscribed user receives the notifications, the verification succeeds.



NOTE

Sending manually triggered notifications to individual users is currently not supported.

8.5. CHANGING EMAIL NOTIFICATION SETTINGS FOR A HOST

Satellite can send event notifications for a host to the host's registered owner. You can configure Satellite to send email notifications either to an individual user or a user group. When set to a user group, all group members who are subscribed to the email type receive a message.

To view the notification status for a host, navigate to **Hosts > All Hosts** and click the host you want to view. In the host details page, click the **Additional Information** tab, you can view the email notification status.

Receiving email notifications for a host can be useful, but also overwhelming if you are expecting to receive frequent errors, for example, because of a known issue or error you are working around. To change the email notification settings for a host, complete the following steps.

Procedure

1. In the Satellite web UI, navigate to **Hosts > All Hosts**, and select the host with the notification setting you want to change.
2. Select the host's checkbox, and from the **Select Action** list, select **Enable Notifications** or **Disable Notifications**, depending on what you want.

CHAPTER 9. MANAGING SECURITY COMPLIANCE

Security compliance management is the ongoing process of defining security policies, auditing for compliance with those policies and resolving instances of non-compliance. Any non-compliance is managed according to the organization's configuration management policies. Security policies range in scope from host-specific to industry-wide, therefore, flexibility in their definition is required.

9.1. SECURITY CONTENT AUTOMATION PROTOCOL

Satellite uses the Security Content Automation Protocol (SCAP) to define security configuration policies. For example, a security policy might specify that for hosts running Red Hat Enterprise Linux, login via SSH is not permitted for the **root** account. With Satellite, you can schedule compliance auditing and reporting on all managed hosts. For more information about SCAP, see [Red Hat Enterprise Linux 8 Security hardening](#).

9.1.1. SCAP Content

SCAP content is a datastream format containing the configuration and security baseline against which hosts are checked. Checklists are described in the extensible checklist configuration description format (XCCDF) and vulnerabilities in the open vulnerability and assessment language (OVAL). Checklist items, also known as rules express the desired configuration of a system item. For example, you may specify that no one can log in to a host over SSH using the **root** user account. Rules can be grouped into one or more profiles, allowing multiple profiles to share a rule. SCAP content consists of both rules and profiles.

You can either create SCAP content or obtain it from a vendor. Supported profiles are provided for Red Hat Enterprise Linux in the **scap-security-guide** package. The creation of SCAP content is outside the scope of this guide, but see [Red Hat Enterprise Linux 8 Security hardening](#) for information on how to download, deploy, modify, and create your own content.

The default SCAP content provided with the OpenSCAP components of Satellite depends on the version of Red Hat Enterprise Linux. On Red Hat Enterprise Linux 7, content for both Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7 is installed.

9.1.2. XCCDF Profile

An XCCDF profile is a checklist against which a host or host group is evaluated. Profiles are created to verify compliance with an industry standard or custom standard.

The profiles provided with Satellite are obtained from the [OpenSCAP project](#).

9.1.3. Listing Available XCCDF Profiles

In the Satellite web UI, list the available XCCDF profiles.

Procedure

- In the Satellite web UI, navigate to **Hosts > SCAP contents**.

9.2. INSTALLING THE OPENSAP PLUG-IN

You can install and enable the OpenSCAP plug-in to generate OpenSCAP compliance reports. The OpenSCAP plug-in consists of the main OpenSCAP plug-in itself, the OpenSCAP smart proxy plug-in, and the OpenSCAP Hammer CLI plug-in.

Procedure

1. Install the OpenSCAP plug-in on your Satellite Server:

```
# satellite-installer --enable-foreman-plugin-openscap --enable-foreman-proxy-plugin-openscap
```

2. Install the OpenSCAP plug-in on any Capsule Servers:

```
# satellite-installer --enable-foreman-proxy-plugin-openscap
```

3. Install the OpenSCAP plug-in Puppet module:

```
# dnf install puppet-foreman_scap_client
```

4. In the Satellite web UI, navigate to **Configure > Puppet Classes**
5. Click **Import environments from satellite.example.com**
You can use Puppet to install and configure the OpenSCAP plug-in on your Satellite Server and Capsules.

9.3. CONFIGURING SCAP CONTENT

9.3.1. Importing OpenSCAP Puppet Modules



NOTE

If you do not use Puppet to configure OpenSCAP auditing on hosts, you can skip this procedure.

To audit hosts with OpenSCAP, you must first import a Puppet environment. The Puppet environment contains the Puppet classes you must assign to each host to deploy the OpenSCAP configuration.

You must associate each host that you want to audit with the Puppet environment in the Satellite web UI.

Procedure

1. In the Satellite web UI, navigate to **Configure > Environments**.
2. Click **Import environments from satellite.example.com**.
3. Select the Puppet environment checkbox associated with the host you want to audit.
If no Puppet environment exists, select the **production** environment checkbox. The Puppet classes that you require for OpenSCAP are in the **production** environment by default.
4. Click **Update**.

9.3.2. Loading the Default OpenSCAP Content

In the CLI, load the default OpenSCAP content using one of the following methods.

Procedure

- Use the Hammer command:

```
# hammer scap-content bulk-upload --type default
```

- (Deprecated) Use the **foreman-rake** command:

```
# foreman-rake foreman_openscap:bulk_upload:default
```

9.3.3. Extra SCAP Content

You can upload extra SCAP content into Satellite Server, either content created by yourself or obtained elsewhere. SCAP content must be imported into Satellite Server before being applied in a policy.

For example, the **scap-security-guide** RPM package available in the Red Hat Enterprise Linux repositories includes a profile for the Payment Card Industry Data Security Standard (PCI-DSS) version 3. You can upload this content into a Satellite Server even if it is not running Red Hat Enterprise Linux as the content is not specific to an operating system version.

9.3.4. Uploading Extra SCAP Content

In the Satellite web UI, upload the extra SCAP content. To use the CLI instead of the Satellite web UI, see the [CLI procedure](#).

Procedure

1. In the Satellite web UI, navigate to **Hosts > SCAP contents** and click **New SCAP Content**.
2. Enter a title in the **Title** text box.
Example: **RHEL 7.2 SCAP Content**.
3. Click **Choose file**, navigate to the location containing the SCAP content file and select **Open**.
4. Click **Submit**.

If the SCAP content file is loaded successfully, a message similar to **Successfully created RHEL 7.2 SCAP Content** is shown and the list of **SCAP Contents** includes the new title.

CLI procedure

1. To upload SCAP content to your Satellite Server, enter the following command:

```
# hammer scap-content bulk-upload \  
--directory /usr/share/xml/scap/ssg/content/ \  
--location "_My_Location_" \  
--organization "_My_Organization_" \  
--type directory
```

SCAP content in **/usr/share/xml/scap/ssg/content/** is part of the **scap-security-guide** package.

9.4. MANAGING COMPLIANCE POLICIES

A scheduled audit, also known as a *compliance policy*, is a scheduled task that checks the specified hosts for compliance against an XCCDF profile. The schedule for scans is specified by Satellite Server and the scans are performed on the host. When a scan completes, an *Asset Reporting File* (ARF) is generated in XML format and uploaded to Satellite Server. You can see the results of the scan in the compliance policy dashboard. No changes are made to the scanned host by the compliance policy. The SCAP content includes several profiles with associated rules but policies are not included by default.

9.4.1. Creating a Compliance Policy

With Satellite, you can create a compliance policy to scan your content hosts to ensure that the hosts remain compliant to your security requirements.

You can use either Puppet or Ansible to deploy the compliance policy to your hosts. Note that Puppet runs by default every 30 minutes. If you assign a new policy, the next Puppet run synchronizes the policy to the host. However Ansible does not perform scheduled runs. To add a new policy, you must run Ansible role manually or using remote execution. For more information about remote execution, see [Configuring and Setting up Remote Jobs](#) in *Managing Hosts*.

Prerequisites

Before you begin, you must decide whether you want to use a Puppet or Ansible deployment.

- For Puppet deployment, ensure that each host that you want to audit is associated with a Puppet environment. For more information, see [Section 9.3.1, "Importing OpenSCAP Puppet Modules"](#).
- For Ansible deployment, ensure that you import the **theforeman.foreman_scap_client** Ansible role. For more information about importing Ansible roles, see [Getting Started with Ansible in Satellite](#) in *Managing Configurations Using Ansible Integration in Red Hat Satellite*.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Policies**, and select whether you want a manual, Ansible, or Puppet deployment.
2. Enter a name for this policy, a description (optional), then click **Next**.
3. Select the SCAP Content and XCCDF Profile to be applied, then click **Next**.
Note that the openSCAP plugin does not detect if a SCAP content role has no content, which means that the **Default XCCDF Profile** might return an empty report.
4. Specify the scheduled time when the policy is to be applied, then click **Next**.
Select **Weekly**, **Monthly**, or **Custom** from the **Period** list.
 - If you select **Weekly**, also select the desired day of the week from the **Weekday** list.
 - If you select **Monthly**, also specify the desired day of the month in the **Day of month** field.
 - If you select **Custom**, enter a valid Cron expression in the **Cron line** field.
The **Custom** option allows for greater flexibility in the policy's schedule than either the **Weekly** or **Monthly** options.
5. Select the locations to which the policy is to be applied, then click **Next**.
6. Select the organizations to which the policy is to be applied, then click **Next**.
7. Select the host groups to which the policy is to be applied, then click **Submit**.

When the Puppet agent runs on the hosts which belong to the selected host group, or hosts to which the policy has been applied, the OpenSCAP client will be installed and a Cron job added with the policy's specified schedule. The **SCAP Content** tab provides the name of the SCAP content file which will be distributed to the directory `/var/lib/openscap/content/` on all target hosts.

9.4.2. Viewing a Compliance Policy

You can preview the rules which will be applied by specific OpenSCAP content and profile combination. This is useful when planning policies.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Policies**.
2. In the **Actions** column of the required policy, click **Show Guide** or select it from the list.

9.4.3. Editing a Compliance Policy

In the Satellite web UI, you can edit compliance policies.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Policies**.
2. From the drop-down list to the right of the policy's name, select **Edit**.
3. Edit the necessary attributes.
4. Click **Submit**.

An edited policy is applied to the host when its Puppet agent next checks with Satellite Server for updates. By default, this occurs every 30 minutes.

9.4.4. Deleting a Compliance Policy

In the Satellite web UI, you can delete existing compliance policies.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Policies**.
2. From the drop-down list to the right of the policy's name, select **Delete**.
3. Click **OK** in the confirmation message.

9.5. CUSTOMIZING OPENSCAP POLICIES

You can customize existing OpenSCAP policies using Tailoring Files without forking or rewriting the policy. You can assign a Tailoring File to a policy when creating or updating a policy.

You can create a Tailoring File using the [SCAP Workbench](#). For more information on using the SCAP Workbench tool, see [Customizing SCAP Security Guide for your use-case](#).

9.5.1. Uploading a Tailoring File

In the Satellite web UI, you can upload a Tailoring file.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Compliance – Tailoring Files** and click **New Tailoring File**.
2. Enter a name in the **Name** text box.
3. Click **Choose File**, navigate to the location containing the SCAP DataStream Tailoring File and select **Open**.
4. Click **Submit** to upload the chosen Tailoring File.

9.5.2. Assigning a Tailoring File to a Policy

In the Satellite web UI, assign a Tailoring file to a policy.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Compliance – Policies**.
2. Click **New Policy**, or **New Compliance Policy** if there are existing Compliance Policies.
3. Enter a name in the **Name** text box, and click **Next**.
4. Select a **Scap content** from the dropdown menu.
5. Select a **XCCDF Profile** from the dropdown menu.
6. Select a **Tailoring File** from the dropdown menu.
7. Select a **XCCDF Profile in Tailoring File** from the dropdown menu.
It is important to select the XCCDF Profile because Tailoring Files are able to contain multiple XCCDF Profiles.
8. Click **Next**.
9. Select a **Period** from the dropdown menu.
10. Select a **Weekday** from the dropdown menu, and click **Next**.
11. Select a **Location** to move it to the **Selected Items** window, and click **Next**.
12. Select an **Organization** to move it to the **Selected Items** window, and click **Next**.
13. Select a **Hostgroup** to move it to the **Selected Items** window, and click **Submit**.

9.6. CONFIGURING A HOST GROUP FOR OPENSCAP

Use this procedure to configure all the OpenSCAP requirements for a host group.

Prerequisites

- Enable OpenSCAP on Capsule. For more information, see [Enabling OpenSCAP on External Capsules](#) in *Installing Capsule Server*.

- Assign an OpenSCAP Capsule.
- Assign a Puppet environment that contains the Puppet classes to deploy the OpenSCAP policies.
- Assign the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
- Assign any compliance policies that you want to add.

For information about creating and administering hosts, see the [Managing Hosts](#) guide.

Procedure

1. In the Satellite web UI, navigate to **Configure > Host Groups**, and either create a host group or click the host group that you want to configure for OpenSCAP reporting.
2. From the **Puppet Environment** list, select the Puppet environment that contains the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
3. From the **OpenSCAP Capsule** list, select the Capsule with OpenSCAP enabled that you want to use.
4. Click the **Puppet Classes** tab, and add the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
5. Click **Submit** to save your changes.
6. In the Satellite web UI, navigate to **Hosts > Policies**.
7. Select the policy that you want to assign to the host group.
8. Click the **Host Groups** tab.
9. From the **Host Groups** list, select as many host groups as you want to assign to this policy.
10. Click **Submit** to save your changes.

CHAPTER 10. RUNNING OPENS CAP SCANS

Procedure

1. In the Satellite web UI, navigate to **Hosts > All Hosts**
2. Select one or multiple hosts.
3. Click on **Run OpenSCAP scan**
Alternatively, [schedule a remote job](#) to scan one or multiple hosts.

10.1. CONFIGURING A HOST FOR OPENS CAP

Use this procedure to configure all the OpenSCAP requirements for a host.

Prerequisites

- Enable OpenSCAP on Capsule. For more information, see [Enabling OpenSCAP on External Capsules](#) in *Installing Capsule Server*.
- Assign an OpenSCAP Capsule.
- Assign a Puppet environment that contains the Puppet classes to deploy the OpenSCAP policies.
- Assign the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
- Assign any compliance policies that you want to add.

For information about creating and administering hosts, see the [Managing Hosts](#) guide.

Procedure

1. In the Satellite web UI, navigate to **Hosts > All Hosts**, and select **Edit** on the host you want to configure for OpenSCAP reporting.
2. From the **Puppet Environment** list, select the Puppet environment that contains the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
3. From the **OpenSCAP Capsule** list, select the Capsule with OpenSCAP enabled that you want to use.
4. Click the **Puppet Classes** tab, and add the **foreman_scap_client** and **foreman_scap_client::params** Puppet classes.
5. To add a compliance policy, navigate to one of the following locations:
6. In the Satellite web UI, navigate to **Hosts > All Hosts**.
7. Select the host or hosts to which you want to add the policy.
8. Click **Select Action**.
9. Select **Assign Compliance Policy** from the list.

10. In the Policy window, select the policy that you want from the list of available policies and click **Submit**.

10.2. MONITORING COMPLIANCE

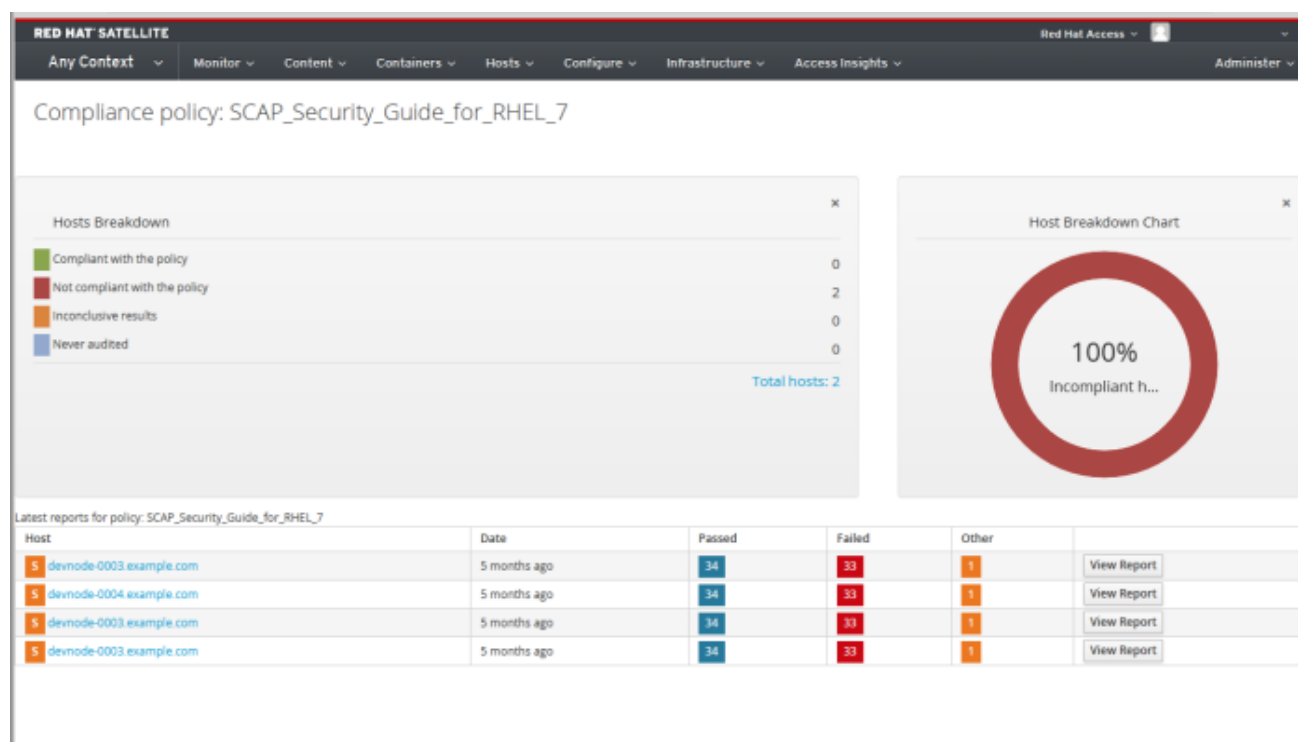
Satellite enables centralized compliance monitoring and management. A compliance dashboard provides an overview of compliance of hosts and the ability to view details for each host within the scope of that policy. Compliance reports provide a detailed analysis of compliance of each host with the applicable policy. With this information, you can evaluate the risks presented by each host and manage the resources required to bring hosts into compliance.

Common objectives when monitoring compliance using SCAP include the following:

- Verifying policy compliance.
- Detecting changes in compliance.

10.2.1. Compliance Policy Dashboard

The compliance policy dashboard provides a statistical summary of compliance of hosts and the ability to view details for each host within the scope of that policy. For all hosts which were evaluated as non-compliant, the **Failed** statistic provides a useful metric for prioritizing compliance effort. The hosts detected as **Never audited** should also be a priority, since their status is unknown.



10.2.2. Viewing the Compliance Policy Dashboard

Use the Satellite web UI to verify policy compliance with the compliance policy dashboard.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Policies**.
2. Click the required policy name. The dashboard provides the following information:

- A ring chart illustrating a high-level view of compliance of hosts with the policy.
- A statistical breakdown of compliance of hosts with the policy, in a tabular format.
- Links to the latest policy report for each host.

10.2.3. Compliance Email Notifications

Satellite Server sends an OpenSCAP Summary email to all users who subscribe to the **Openscap policy summary** email notifications. For more information on subscribing to email notifications, see [Section 8.2, “Configuring Email Notification Preferences”](#). Each time a policy is run, Satellite checks the results against the previous run, noting any changes between them. The email is sent according to the frequency requested by each subscriber, providing a summary of each policy and its most recent result.

An **OpenSCAP Summary** email message contains the following information:

- Details of the time period it covers.
- Totals for all hosts by status: changed, compliant, and noncompliant.
- A tabular breakdown of each host and the result of its latest policy, including totals of the rules that passed, failed, changed, or where results were unknown.

10.2.4. Compliance Reports

A compliance report is the output of a policy run against a host. Each report includes the total number of rules passed or failed per policy. By default, reports are listed in descending date order.

In the Satellite web UI, navigate to **Hosts > Reports** to list all compliance reports.

A compliance report consists of the following areas:

- Introduction
- Evaluation Characteristics
- Compliance and Scoring
- Rule Overview

Evaluation Characteristics

The Evaluation Characteristics area provides details about an evaluation against a specific profile, including the host that was evaluated, the profile used in the evaluation, and when the evaluation started and finished. For reference, the IPv4, IPv6, and MAC addresses of the host are also listed.

Name	Description	Example
Target machine	The fully-qualified domain name (FQDN) of the evaluated host.	test-system.example.com
Benchmark URL	The URL of the SCAP content against which the host was evaluated.	/var/lib/openscap/content/1fbdc87d24db51ca184419a2b6f

Name	Description	Example
Benchmark ID	The identifier of the benchmark against which the host was evaluated. A benchmark is a set of profiles	xccdf_org.ssgproject.content_benchmark_RHEL_7
Profile ID	The identifier of the profile against which the host was evaluated.	xccdf_org.ssgproject_content_profile_rht-ccp
Started at	The date and time at which the evaluation started, in ISO 8601 format.	2015-09-12T14:40:02
Finished at	The date and time at which the evaluation finished, in ISO 8601 format.	2015-09-12T14:40:05
Performed by	The local account name under which the evaluation was performed on the host.	root

Compliance and Scoring

The Compliance and Scoring area provides an overview of whether or not the host is in compliance with the profile rules, a breakdown of compliance failures by severity, and an overall compliance score as a percentage. If compliance with a rule was not checked, this is categorized in the **Rule results** field as **Other**.

Rule Overview

The Rule Overview area provides details about every rule and the compliance result, with the rules presented in a hierarchical layout.

Select or clear the checkboxes to narrow the list of rules included in the compliance report. For example, if the focus of your review is any non-compliance, clear the **pass** and **informational** checkboxes.

To search all rules, enter a criterion in the **Search** field. The search is dynamically applied as you type. The **Search** field only accepts a single plain-text search term and it is applied as a case-insensitive search. When you perform a search, only those rules whose descriptions match the search criterion will be listed. To remove the search filter, delete the search criterion.

For an explanation of each result, hover the cursor over the status shown in the **Result** column.

10.2.5. Examining Compliance Failures of Hosts

Use the Satellite web UI to determine why a host failed compliance on a rule.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Reports** to list all compliance reports.
2. Click **View Report** in the row of the specific host to view the details of an individual report.
3. Click on the rule's title to see further details:

- A description of the rule with instructions for bringing the host into compliance if available.
- The rationale for the rule.
- In some cases, a remediation script.



WARNING

Do not implement any of the recommended remedial actions or scripts without first testing them in a non-production environment.

10.2.6. Searching Compliance Reports

Use the Compliance Reports search field to filter the list of available reports on any given subset of hosts.

Procedure

- To apply a filter, enter the search query in the **Search** field and click **Search**. The search query is case insensitive.

Search Use Cases

- The following search query finds all compliance reports for which more than five rules failed:

```
failed > 5
```

- The following search query finds all compliance reports created after January 1, YYYY, for hosts with host names that contain the **prod-** group of characters:

```
host ~ prod- AND date > "Jan 1, YYYY"
```

- The following search query finds all reports generated by the **rhel7_audit** compliance policy from an hour ago:

```
"1 hour ago" AND compliance_policy = date = "1 hour ago" AND compliance_policy = rhel7_audit
```

- The following search query finds reports that pass an XCCDF rule:

```
xccdf_rule_passed = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

- The following search query finds reports that fail an XCCDF rule:

```
xccdf_rule_failed = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

- The following search query finds reports that have a result different than fail or pass for an XCCDF rule:

```
xccdf_rule_othered = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

Additional Information

- To see a list of available search parameters, click the empty **Search** field.
- You can create complex queries with the following logical operators: **and**, **not** and **has**. For more information about logical operators, see [Section 7.7.3, "Supported Operators for Granular Search"](#).
- You cannot use regular expressions in a search query. However, you can use multiple fields in a single search expression. For more information about all available search operators, see [Section 7.7.3, "Supported Operators for Granular Search"](#).
- You can bookmark a search to reuse the same search query. For more information, see [Section 19.3.1, "Creating Bookmarks"](#).

10.2.7. Deleting a Compliance Report

You can delete compliance reports on your Satellite.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Reports**.
2. In the Compliance Reports window, identify the policy that you want to delete and, on the right of the policy's name, select **Delete**.
3. Click **OK**.

10.2.8. Deleting Multiple Compliance Reports

You can delete multiple compliance policies simultaneously. However, in the Satellite web UI, compliance policies are paginated, so you must delete one page of reports at a time. If you want to delete all OpenSCAP reports, use the script in the [Deleting OpenSCAP Reports](#) section of *API Guide*.

Procedure

1. In the Satellite web UI, navigate to **Hosts > Reports**.
2. In the Compliance Reports window, select the compliance reports that you want to delete.
3. In the upper right of the list, select **Delete reports**.
4. Repeat these steps for as many pages as you want to delete.

10.3. SPECIFICATIONS SUPPORTED BY OPENS CAP

The following specifications are supported by OpenSCAP:

Title	Description	Version
XCCDF	The Extensible Configuration Checklist Description Format	1.2
OVAL	Open Vulnerability and Assessment Language	5.11
-	Asset Identification	1.1
ARF	Asset Reporting Format	1.1
CCE	Common Configuration Enumeration	5.0
CPE	Common Platform Enumeration	2.3
CVE	Common Vulnerabilities and Exposures	-
CVSS	Common Vulnerability Scoring System	2.0

CHAPTER 11. BACKING UP SATELLITE SERVER AND CAPSULE SERVER

You can back up your Satellite deployment to ensure the continuity of your Red Hat Satellite deployment and associated data in the event of a disaster. If your deployment uses custom configurations, you must consider how to handle these custom configurations when you plan your backup and disaster recovery policy.

+



NOTE

If you create a new instance of the Satellite Server, decommission the old instances after restoring the backup. Cloned instances are not supposed to run in parallel in a production environment.

To create a backup of your Satellite Server or Capsule Server and all associated data, use the **satellite-maintain backup** command. Backing up to a separate storage device on a separate system is highly recommended.

Satellite services are unavailable during the backup. Therefore, you must ensure that no other tasks are scheduled by other administrators. You can schedule a backup using **cron**. For more information, see the [Section 11.5, “Example of a Weekly Full Backup Followed by Daily Incremental Backups”](#).

During offline or snapshot backups, the services are inactive and Satellite is in a maintenance mode. All the traffic from outside on port 443 is rejected by a firewall to ensure there are no modifications triggered.

A backup contains sensitive information from the **/root/ssl-build** directory. For example, it can contain hostnames, ssh keys, request files and SSL certificates. You must encrypt or move the backup to a secure location to minimize the risk of damage or unauthorized access to the hosts.

Conventional Backup Methods

You can also use conventional backup methods. For more information, see [Recovering and restoring a system](#) in *Red Hat Enterprise Linux 8 Configuring basic system settings*.



NOTE

If you plan to use the **satellite-maintain backup** command to create a backup, do not stop Satellite services.

- When creating a snapshot or conventional backup, you must stop all services as follows:

```
# satellite-maintain service stop
```

- Start the services after creating a snapshot or conventional backup:

```
# satellite-maintain service start
```

11.1. ESTIMATING THE SIZE OF A BACKUP

The full backup creates uncompressed archives of PostgreSQL and Pulp database files, and Satellite configuration files. Compression occurs after the archives are created to decrease the time when Satellite services are unavailable.

A full backup requires space to store the following data:

- Uncompressed Satellite database and configuration files
- Compressed Satellite database and configuration files
- An extra 20% of the total estimated space to ensure a reliable backup

Procedure

1. Enter the **du** command to estimate the size of uncompressed directories containing Satellite database and configuration files:

```
# du -sh /var/lib/pgsql/data /var/lib/pulp
100G  /var/lib/pgsql/data
100G /var/lib/pulp

# du -csh /var/lib/qpidd /var/lib/tftpboot /etc /root/ssl-build \
/var/www/html/pub /opt/puppetlabs
886M  /var/lib/qpidd
16M   /var/lib/tftpboot
37M   /etc
900K  /root/ssl-build
100K  /var/www/html/pub
2M    /opt/puppetlabs
942M  total
```

2. Calculate how much space is required to store the compressed data.
The following table describes the compression ratio of all data items included in the backup:

Table 11.1. Backup Data Compression Ratio

Data type	Directory	Ratio	Example results
PostgreSQL database files	/var/lib/pgsql/data	80 – 85%	100 GB → 20 GB
Pulp RPM files	/var/lib/pulp	(not compressed)	100 GB
Configuration files	/var/lib/qpidd /var/lib/tftpboot /etc /root/ssl-build /var/www/html/pub /opt/puppetlabs	85%	942 MB → 141 MB

In this example, the compressed backup data occupies 120 GB in total.

3. To calculate the amount of available space you require to store a backup, calculate the sum of the estimated values of compressed and uncompressed backup data, and add an extra 20% to ensure a reliable backup.
This example requires 201 GB plus 120 GB for the uncompressed and compressed backup data, 321 GB in total. With 64 GB of extra space, 385 GB must be allocated for the backup location.

11.2. PERFORMING A FULL BACKUP OF SATELLITE SERVER OR CAPSULE SERVER

Red Hat Satellite uses the **satellite-maintain backup** command to make backups.

There are three main methods of backing up Satellite Server:

- Offline backup
 - Online backup
 - Snapshot backups
- For more information about each of these methods, you can view the usage statements for each backup method.

Offline backups

```
# satellite-maintain backup offline --help
```

Online backups

```
# satellite-maintain backup online --help
```

Snapshots backups

```
# satellite-maintain backup snapshot --help
```

Directory creation

The **satellite-maintain backup** command creates a time-stamped subdirectory in the backup directory that you specify. The **satellite-maintain backup** command does not overwrite backups, therefore you must select the correct directory or subdirectory when restoring from a backup or an incremental backup. The **satellite-maintain backup** command stops and restarts services as required.

When you run the **satellite-maintain backup offline** command, the following default backup directories are created:

- **satellite-backup** on Satellite
- **foreman-proxy-backup** on Capsule

If you want to set a custom directory name, add the **--preserve-directory** option and add a directory name. The backup is then stored in the directory you provide in the command line. If you use the **--preserve-directory** option, no data is removed if the backup fails.

Note that if you use a local PostgreSQL database, the **postgres** user requires write access to the backup directory.

Remote databases

You can use the **satellite-maintain backup** command to back up remote databases.

You can use both online and offline methods to back up remote databases, but if you use offline methods, such as snapshot, the **satellite-maintain backup** command performs a database dump.

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see [Section 11.1, “Estimating the Size of a Backup”](#).



WARNING

Request other users of Satellite Server or Capsule Server to save any changes and warn them that Satellite services are unavailable for the duration of the backup. Ensure no other tasks are scheduled for the same time as the backup.

Procedure

- On Satellite Server, enter the following command:

```
# satellite-maintain backup offline /var/satellite-backup
```

- On Capsule Server, enter the following command:

```
# satellite-maintain backup offline /var/foreman-proxy-backup
```

11.3. PERFORMING A BACKUP WITHOUT PULP CONTENT

You can perform an offline backup that excludes the contents of the Pulp directory. The backup without Pulp content is useful for debugging purposes and is only intended to provide access to configuration files without backing up the Pulp database. You cannot restore from a directory that does not contain Pulp content.



WARNING

Request other users of Satellite Server or Capsule Server to save any changes and warn them that Satellite services are unavailable for the duration of the backup. Ensure no other tasks are scheduled for the same time as the backup.

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see [Section 11.1, “Estimating the Size of a Backup”](#).

Procedure

- To perform an offline backup without Pulp content, enter the following command:

```
# satellite-maintain backup offline --skip-pulp-content /var/backup_directory
```

11.4. PERFORMING AN INCREMENTAL BACKUP

Use this procedure to perform an offline backup of any changes since a previous backup.

To perform incremental backups, you must perform a full backup as a reference to create the first incremental backup of a sequence. Keep the most recent full backup and a complete sequence of incremental backups to restore from.



WARNING

Request other users of Satellite Server or Capsule Server to save any changes and warn them that Satellite services are unavailable for the duration of the backup. Ensure no other tasks are scheduled for the same time as the backup.

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see [Section 11.1, “Estimating the Size of a Backup”](#).

Procedure

- To perform a full offline backup, enter the following command:

```
# satellite-maintain backup offline /var/backup_directory
```

- To create a directory within your backup directory to store the first incremental back up, enter the **satellite-maintain backup** command with the **--incremental** option:

```
# satellite-maintain backup offline --incremental /var/backup_directory/full_backup
/var/backup_directory
```

- To create the second incremental backup, enter the **satellite-maintain backup** command with the **--incremental** option and include the path to the first incremental backup to indicate the starting point for the next increment. This creates a directory for the second incremental backup in your backup directory:

```
# satellite-maintain backup offline --incremental
/var/backup_directory/first_incremental_backup /var/backup_directory
```

- Optional: If you want to point to a different version of the backup, and make a series of increments with that version of the backup as the starting point, you can do this at any time. For example, if you want to make a new incremental backup from the full backup rather than the first or second incremental backup, point to the full backup directory:

```
# satellite-maintain backup offline --incremental /var/backup_directory/full_backup
/var/backup_directory
```

11.5. EXAMPLE OF A WEEKLY FULL BACKUP FOLLOWED BY DAILY INCREMENTAL BACKUPS

The following script performs a full backup on a Sunday followed by incremental backups for each of the following days. A new subdirectory is created for each day that an incremental backup is performed. The script requires a daily cron job.

```
#!/bin/bash -e
PATH=/sbin:/bin:/usr/sbin:/usr/bin
DESTINATION=/var/backup_directory
if [[ $(date +%w) == 0 ]]; then
    satellite-maintain backup offline --assumeyes $DESTINATION
else
    LAST=$(ls -td -- $DESTINATION/* | head -n 1)
    satellite-maintain backup offline --assumeyes --incremental "$LAST" $DESTINATION
fi
exit 0
```

Note that the **satellite-maintain backup** command requires **/sbin** and **/usr/sbin** directories to be in **PATH** and the **--assumeyes** option is used to skip the confirmation prompt.

11.6. PERFORMING AN ONLINE BACKUP

Perform an online backup only for debugging purposes.

Risks Associated with Online Backups

When performing an online backup, if there are procedures affecting the Pulp database, the Pulp part of the backup procedure repeats until it is no longer being altered. Because the backup of the Pulp database is the most time consuming part of backing up Satellite, if you make a change that alters the Pulp database during this time, the backup procedure keeps restarting.

For production environments, use the snapshot method. For more information, see [Section 11.7, “Performing a Snapshot Backup”](#). If you want to use the online backup method in production, proceed with caution and ensure that no modifications occur during the backup.



WARNING

Request other users of Satellite Server or Capsule Server to save any changes and warn them that Satellite services are unavailable for the duration of the backup. Ensure no other tasks are scheduled for the same time as the backup.

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see [Section 11.1, “Estimating the Size of a Backup”](#).

Procedure

- To perform an online backup, enter the following command:

```
# satellite-maintain backup online /var/backup_directory
```

11.7. PERFORMING A SNAPSHOT BACKUP

You can perform a snapshot backup that uses Logical Volume Manager (LVM) snapshots of the Pulp, and PostgreSQL directories. Creating a backup from LVM snapshots mitigates the risk of an inconsistent backup.

The snapshot backup method is faster than a full offline backup and therefore reduces Satellite downtime.

To view the usage statement, enter the following command:

```
satellite-maintain backup snapshot -h
```



WARNING

Request other Satellite Server or Capsule Server users to save any changes and warn them that Satellite services are unavailable for the duration of the backup. Ensure no other tasks are scheduled for the same time as the backup.

Prerequisites

- The system uses LVM for the directories that you snapshot: **/var/lib/pulp/**, and **/var/lib/pgsql**.
- The free disk space in the relevant volume group (VG) is three times the size of the snapshot. More precisely, the VG must have enough space unreserved by the member logical volumes (LVs) to accommodate new snapshots. In addition, one of the LVs must have enough free space for the backup directory.
- The target backup directory is on a different LV than the directories that you snapshot.

Procedure

- To perform a snapshot backup, enter the **satellite-maintain backup snapshot** command:

```
# satellite-maintain backup snapshot /var/backup_directory
```

The **satellite-maintain backup snapshot** command creates snapshots when the services are active, and stops all services which can impact the backup. This makes the maintenance window shorter. After the successful snapshot, all services are restarted and LVM snapshots are removed.

11.8. WHITE-LISTING AND SKIPPING STEPS WHEN PERFORMING BACKUPS

A backup using the **satellite-maintain backup** command proceeds in a sequence of steps. To skip part of the backup add the **--whitelist** option to the command and add the step label that you want to omit.

Procedure

- To display a list of available step labels, enter the following command:

```
# satellite-maintain advanced procedure run -h
```

- To skip a step of the backup, enter the **satellite-maintain backup** command with the **--whitelist** option. For example:

```
# satellite-maintain backup online --whitelist backup-metadata -y /var/backup_directory
```

CHAPTER 12. RESTORING SATELLITE SERVER OR CAPSULE SERVER FROM A BACKUP

You can restore Satellite Server or Capsule Server from the backup data that you create as part of [Chapter 11, Backing Up Satellite Server and Capsule Server](#). This process outlines how to restore the backup on the same server that generated the backup, and all data covered by the backup is deleted on the target system. If the original system is unavailable, provision a system with the same configuration settings and host name.

12.1. RESTORING FROM A FULL BACKUP

Use this procedure to restore Red Hat Satellite or Capsule Server from a full backup. When the restore process completes, all processes are online, and all databases and system configuration revert to the state at the time of the backup.

Prerequisites

- Ensure that you are restoring to the correct instance. The Red Hat Satellite instance must have the same host name, configuration, and be the same minor version (X.Y) as the original system.
- Ensure that you have an existing target directory. The target directory is read from the configuration files contained within the archive.
- Ensure that you have enough space to store this data on the base system of Satellite Server or Capsule Server as well as enough space after the restoration to contain all the data in the **/etc/** and **/var/** directories contained within the backup.

To check the space used by a directory, enter the following command:

```
# du -sh /var/backup_directory
```

To check for free space, enter the following command:

```
# df -h /var/backup_directory
```

Add the **--total** option to get a total of the results from more than one directory.

- Ensure that all SELinux contexts are correct. Enter the following command to restore the correct SELinux contexts:

```
# restorecon -Rv /
```

Procedure

1. Choose the appropriate method to install Satellite or Capsule:
 - To install Satellite Server from a connected network, follow the procedures in [Installing Satellite Server in a Connected Network Environment](#).
 - To install Satellite Server from a disconnected network, follow the procedures in [Installing Satellite Server in a Disconnected Network Environment](#).
 - To install a Capsule Server, follow the procedures in [Installing Capsule Server](#).

2. Copy the backup data to Satellite Server's local file system. Use **/var/** or **/var/tmp/**.
3. Run the restoration script.

```
# satellite-maintain restore /var/backup_directory
```

Where *backup_directory* is the time-stamped directory or subdirectory containing the backed-up data.

The restore process can take a long time to complete, because of the amount of data to copy.

Additional Resources

- For troubleshooting, you can check **/var/log/foreman/production.log** and **/var/log/messages**.

12.2. RESTORING FROM INCREMENTAL BACKUPS

Use this procedure to restore Satellite or Capsule Server from incremental backups. If you have multiple branches of incremental backups, select your full backup and each incremental backup for the *branch* you want to restore, in chronological order.

When the restore process completes, all processes are online, and all databases and system configuration revert to the state at the time of the backup.

Procedure

1. Restore the last full backup using the instructions in [Section 12.1, "Restoring from a Full Backup"](#).
2. Remove the full backup data from Satellite Server's local file system, for example, **/var/** or **/var/tmp/**.
3. Copy the incremental backup data to Satellite Server's local file system, for example, **/var/** or **/var/tmp/**.
4. Restore the incremental backups in the same sequence that they are made:

```
# satellite-maintain restore /var/backup_directory/FIRST_INCREMENTAL
# satellite-maintain restore /var/backup_directory/SECOND_INCREMENTAL
```

Additional Resources

- For troubleshooting, you can check **/var/log/foreman/production.log** and **/var/log/messages**.

12.3. BACKUP AND RESTORE CAPSULE SERVER USING A VIRTUAL MACHINE SNAPSHOT

If your Capsule Server is a virtual machine, you can restore it from a snapshot. Creating weekly snapshots to restore from is recommended. In the event of failure, you can install, or configure a new Capsule Server, and then synchronize the database content from Satellite Server.

If required, deploy a new Capsule Server, ensuring the host name is the same as before, and then install the Capsule certificates. You may still have them on Satellite Server, the package name ends in -certs.tar, alternately create new ones. Follow the procedures in [Installing Capsule Server](#) until you can

confirm, in the Satellite web UI, that Capsule Server is connected to Satellite Server. Then use the procedure [Section 12.3.1, “Synchronizing an External Capsule”](#) to synchronize from Satellite.

12.3.1. Synchronizing an External Capsule

Synchronize an external Capsule with Satellite.

Procedure

1. To synchronize an external Capsule, select the relevant organization and location in the Satellite web UI, or choose **Any Organization** and **Any Location**.
2. In the Satellite web UI, navigate to **Infrastructure > Capsules** and click the name of the Capsule to synchronize.
3. On the **Overview** tab, select **Synchronize**.

CHAPTER 13. RENAMING SATELLITE SERVER OR CAPSULE SERVER

To rename Satellite Server or Capsule Server, use the **satellite-change-hostname** script.



IMPORTANT

When changing the domain name of your Satellite Server or Capsule Server, update the hostname using **satellite-change-hostname** to avoid networking issues.

13.1. RENAMING SATELLITE SERVER

The host name of Satellite Server is used by Satellite Server components, all Capsule Servers, and hosts registered to it for communication. This procedure ensures that in addition to renaming Satellite Server, you also update all references to point to the new host name.



WARNING

Renaming your Satellite Server host shuts down all Satellite services on that host. The services restart after the renaming is complete.

Prerequisites

- Back up your Satellite Server before changing its host name. If you fail to successfully rename it, restore it from the backup. For more information, see [Chapter 11, Backing Up Satellite Server and Capsule Server](#).
- Run the **hostname** and **hostname -f** commands on Satellite Server. If both commands do not return the FQDN of Satellite Server, the **satellite-change-hostname** script will fail to complete. If the **hostname** command returns the shortname of Satellite Server instead of the FQDN, use **hostnamectl set-hostname *My_Old_FQDN*** to set the old FQDN correctly before using the **satellite-change-hostname** script.
- If Satellite Server has a custom SSL certificate installed, obtain a new certificate for the new FQDN of the host. For more information, see [Configuring Satellite Server with a Custom SSL Certificate](#) in *Installing Satellite Server in a Connected Network Environment*.

Procedure

1. On Satellite Server, run the **satellite-change-hostname** script, and provide the new host name. Choose one of the following methods:
 - If your Satellite Server is installed with the default self-signed SSL certificates, enter the following command:

```
# satellite-change-hostname new-satellite \
--username admin \
--password password
```

- If your Satellite Server is installed with custom SSL certificates:

```
# satellite-change-hostname new-satellite \
--username admin \
--password password \
--custom-cert "/root/ownca/test.com/test.com.crt" \
--custom-key "/root/ownca/test.com/test.com.key"
```

2. If you have created a custom SSL certificate for the new Satellite Server host name, run the Satellite installation script to install the certificate. For more information about installing a custom SSL certificate, see [Deploying a Custom SSL Certificate to Satellite Server](#) in *Installing Satellite Server in a Connected Network Environment*.
3. Reregister all Satellite hosts. For more information, see [Registering Hosts](#) in *Managing Hosts*.
4. On all Capsule Servers, run the Satellite installation script to update references to the new host name:

```
# satellite-installer \
--foreman-proxy-foreman-base-url https://new-satellite.example.com \
--foreman-proxy-trusted-hosts new-satellite.example.com \
--puppet-server-foreman-url https://new-satellite.example.com
```

5. On Satellite Server, list all Capsule Servers:

```
# hammer capsule list
```

6. On Satellite Server, synchronize content for each Capsule Server:

```
# hammer capsule content synchronize \
--id My_capsule_ID
```

7. If you use the *virt-who* agent, update the *virt-who* configuration files with the new host name. For more information, see [Modifying a virt-who Configuration](#) in *Configuring Virtual Machine Subscriptions in Red Hat Satellite*.
8. If you use external authentication, reconfigure Satellite Server for external authentication after you run the **satellite-change-hostname** script. For more information, see [Configuring External Authentication](#) in *Installing Satellite Server in a Connected Network Environment*.

13.2. RENAMING CAPSULE SERVER

The host name of Capsule Server is referenced by Satellite Server components and all hosts registered to it. This procedure ensures that in addition to renaming Capsule Server, you also update all references to the new host name.

**WARNING**

Renaming your Capsule Server host shuts down all Satellite services on that host. The services restart after the renaming is complete.

Prerequisites

- Back up your Capsule Server before renaming. If you fail to successfully rename it, restore it from the backup. For more information, see [Chapter 11, Backing Up Satellite Server and Capsule Server](#).
- Run the **hostname** and **hostname -f** commands on Capsule Server. If both commands do not return the FQDN of Capsule Server, the **satellite-change-hostname** script will fail to complete. If the **hostname** command returns the shortname of Capsule Server instead of the FQDN, use **hostnamectl set-hostname *My_Old_FQDN*** to set the old FQDN correctly before attempting to use the **satellite-change-hostname** script.

Procedure

1. On Satellite Server, generate a new certificates archive file for Capsule Server.

- If you are using the default SSL certificate, enter the following command:

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

Ensure that you enter the full path to the **.tar** file.

- If you are using a custom SSL certificate, create a new SSL certificate for Capsule Server. For more information, see [Configuring Capsule Server with a Custom SSL Certificate](#) in *Installing Capsule Server*.
2. On Satellite Server, copy the certificates archive file to Capsule Server. For example, to copy the archive file to the **root** user's home directory:

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

3. On Capsule Server, run the **satellite-change-hostname** script and provide the host's new name, Satellite credentials, and certificates archive file name.

```
# satellite-change-hostname new-capsule --username admin \
--password password \
--certs-tar /root/new-capsule.example.com-certs.tar
```

Ensure that you enter the full path to the **.tar** file.

4. If you have created a custom certificate for Capsule Server, deploy the certificate on Capsule Server by entering the **satellite-installer** command that the **capsule-certs-generate** command returned in a previous step. For more information, see [Deploying a Custom SSL Certificate to Capsule Server](#) in *Installing Capsule Server*.

5. On all Capsule clients, enter the following commands to reinstall the bootstrap RPM, reregister clients, and refresh their subscriptions.

You can use the remote execution feature to perform this step. For more information, see [Configuring and Setting up Remote Jobs](#) in *Managing Hosts*.

```
# dnf remove katello-ca-consumer*

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

# subscription-manager register --org="My_Organization" \
--environment="Library" \
--force

# subscription-manager refresh
```

6. Update the Capsule host name in the Satellite web UI.
 - a. In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.
 - b. Locate Capsule Server in the list, and click **Edit**.
 - c. Edit the **Name** and **URL** fields to match Capsule Server's new host name, then click **Submit**.
 - d. On your DNS server, add a record for the new hostname of your Capsule Server, and delete the record of the previous host name.

CHAPTER 14. MAINTAINING SATELLITE SERVER

This chapter provides information on how to maintain a Satellite Server, including information on how to work with audit records, how to clean unused tasks, and how to recover Pulp from a full disk.

14.1. DELETING AUDIT RECORDS

Audit records are created automatically in Satellite. You can use the **foreman-rake audits:expire** command to remove audits at any time. You can also use a cron job to schedule audit record deletions at the set interval that you want.

By default, using the **foreman-rake audits:expire** command removes audit records that are older than 90 days. You can specify the number of days to keep the audit records by adding the **days** option and add the number of days.

For example, if you want to delete audit records that are older than seven days, enter the following command:

```
# foreman-rake audits:expire days=7
```

14.2. ANONYMIZING AUDIT RECORDS

You can use the **foreman-rake audits:anonymize** command to remove any user account or IP information while maintaining the audit records in the database. You can also use a cron job to schedule anonymizing the audit records at the set interval that you want.

By default, using the **foreman-rake audits:anonymize** command anonymizes audit records that are older than 90 days. You can specify the number of days to keep the audit records by adding the **days** option and add the number of days.

For example, if you want to anonymize audit records that are older than seven days, enter the following command:

```
# foreman-rake audits:anonymize days=7
```

14.3. DELETING REPORT RECORDS

Report records are created automatically in Satellite. You can use the **foreman-rake reports:expire** command to remove reports at any time. You can also use a cron job to schedule report record deletions at the set interval that you want.

By default, using the **foreman-rake reports:expire** command removes report records that are older than 90 days. You can specify the number of days to keep the report records by adding the **days** option and add the number of days.

For example, if you want to delete report records that are older than seven days, enter the following command:

```
# foreman-rake reports:expire days=7
```

14.4. CONFIGURING THE CLEANING UNUSED TASKS FEATURE

Satellite performs regular cleaning to reduce disc space in the database and limit the rate of disk growth. As a result, Satellite backup completes faster and overall performance is higher.

By default, Satellite executes a cron job that cleans tasks every day at 19:45. Satellite removes the following tasks during the cleaning:

- Tasks that have run successfully and are older than thirty days
- All tasks that are older than a year

You can configure the cleaning unused tasks feature using these options:

- To configure the time at which Satellite runs the cron job, set the **--foreman-plugin-tasks-cron-line** parameter to the time you want in cron format. For example, to schedule the cron job to run every day at 15:00, enter the following command:

```
# satellite-installer --foreman-plugin-tasks-cron-line "00 15 * * *"
```

- To configure the period after which Satellite deletes the tasks, edit the **:rules:** section in the **/etc/foreman/plugins/foreman-tasks.yaml** file.
- To disable regular task cleanup on Satellite, enter the following command:

```
# satellite-installer --foreman-plugin-tasks-automatic-cleanup false
```

- To reenabling regular task cleanup on Satellite, enter the following command:

```
# satellite-installer --foreman-plugin-tasks-automatic-cleanup true
```

14.5. DELETING TASK RECORDS

Task records are created automatically in Satellite. You can use the **foreman-rake foreman_tasks:cleanup** command to remove tasks at any time. You can also use a cron job to schedule Task record deletions at the set interval that you want.

For example, if you want to delete task records from successful repository synchronizations, enter the following command:

```
# foreman-rake foreman_tasks:cleanup TASK_SEARCH='label = Actions::Katello::Repository::Sync'  
STATES='stopped'
```

14.6. DELETING A TASK BY ID

You can delete tasks by ID, for example if you have submitted confidential data by mistake.

Procedure

1. Connect to your Satellite Server using SSH:

```
# ssh root@satellite.example.com
```

2. Optional: View the task:

```
# hammer task info --id My_Task_ID
```

3. Delete the task:

```
# foreman-rake foreman_tasks:cleanup TASK_SEARCH="id=My_Task_ID"
```

4. Optional: Ensure the task has been removed from Satellite Server:

```
# hammer task info --id My_Task_ID
```

Note that because the task is deleted, this command returns a non-zero exit code.

14.7. RECOVERING FROM A FULL DISK

The following procedure describes how to resolve the situation when a logical volume (LV) with the Pulp database on it has no free space.

Procedure

1. Let running Pulp tasks finish but do not trigger any new ones as they can fail due to the full disk.
2. Ensure that the LV with the **/var/lib/pulp** directory on it has sufficient free space. Here are some ways to achieve that:

- a. Remove orphaned content:

```
# foreman-rake katello:delete_orphaned_content RAILS_ENV=production
```

This is run weekly so it will not free much space.

- b. Change the download policy from **Immediate** to **On Demand** for as many repositories as possible and remove already downloaded packages. See the Red Hat Knowledgebase solution [How to change syncing policy for Repositories on Satellite from "Immediate" to "On-Demand"](#) on the Red Hat Customer Portal for instructions.
- c. Grow the file system on the LV with the **/var/lib/pulp** directory on it. For more information, see [Growing a logical volume and file system](#) in *Red Hat Enterprise Linux 8 Configuring and managing logical volumes*.



NOTE

If you use an untypical file system (other than for example ext3, ext4, or xfs), you might need to unmount the file system so that it is not in use. In that case, complete the following steps:

1. Stop Satellite services:

```
# satellite-maintain service stop
```

2. Grow the file system on the LV.

3. Start Satellite services:

```
# satellite-maintain service start
```

3. If some Pulp tasks failed due to the full disk, run them again.

14.8. MANAGING PACKAGES ON THE BASE OPERATING SYSTEM OF SATELLITE SERVER OR CAPSULE SERVER

To install and update packages on the Satellite Server or Capsule Server base operating system, you must enter the **satellite-maintain packages** command. Satellite prevents users from installing and updating packages with **yum** because **yum** might also update the packages related to Satellite Server or Capsule Server and result in system inconsistency.



IMPORTANT

The **satellite-maintain packages** command restarts some services on the operating system where you run it because it runs the **satellite-installer** command after installing packages.

Procedure

- To install packages on Satellite Server or Capsule Server, enter the following command:

```
# satellite-maintain packages install package_1 package_2
```

- To update specific packages on Satellite Server or Capsule Server, enter the following command:

```
# satellite-maintain packages update package_1 package_2
```

- To update all packages on Satellite Server or Capsule Server, enter the following command:

```
# satellite-maintain packages update
```

Using yum to Check for Package Updates

If you want to check for updates using **yum**, enter the command to install and update packages manually and then you can use **yum** to check for updates:

```
# satellite-maintain packages unlock
# yum check update
# satellite-maintain packages lock
```

Updating packages individually can lead to package inconsistencies in Satellite Server or Capsule Server. For more information about updating packages in Satellite Server, see [Updating Satellite Server](#).

Enabling yum for Satellite Server or Capsule Server Package Management

If you want to install and update packages on your system using **yum** directly and control the stability of the system yourself, enter the following command:

```
# satellite-maintain packages unlock
```

Restoring Package Management to the Default Settings

If you want to restore the default settings and enable Satellite Server or Capsule Server to prevent users from installing and updating packages with **yum** and ensure the stability of the system, enter the following command:

```
# satellite-maintain packages lock
```

14.9. RECLAIMING POSTGRESQL SPACE

The PostgreSQL database can use a large amount of disk space especially in heavily loaded deployments. Use this procedure to reclaim some of this disk space on Satellite.

Procedure

1. Stop all services, except for the **postgresql** service:

```
# satellite-maintain service stop --exclude postgresql
```

2. Switch to the **postgres** user and reclaim space on the database:

```
# su - postgres -c 'vacuumdb --full --all'
```

3. Start the other services when the vacuum completes:

```
# satellite-maintain service start
```

CHAPTER 15. RENEWING THE CUSTOM SSL CERTIFICATE

This chapter provides information on how to renew the custom SSL certificate on Satellite Server as well as on Capsule Server.

15.1. RENEWING A CUSTOM SSL CERTIFICATE ON SATELLITE SERVER

Use this procedure to update your custom SSL certificate for Satellite Server.

Prerequisite

- You must create a new Certificate Signing Request (CSR) and send it to the Certificate Authority to sign the certificate. Refer to the [Configuring Satellite Server with a Custom SSL Certificate](#) guide before creating a new CSR because the Server certificate must have X.509 v3 **Key Usage** and **Extended Key Usage** extensions with required values. In return, you will receive the Satellite Server certificate and CA bundle.

Procedure

- Before deploying a renewed custom certificate on your Satellite Server, validate the custom SSL 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 -t satellite \
-b /root/satellite_cert/ca_cert_bundle.pem \
-c /root/satellite_cert/satellite_cert.pem \
-k /root/satellite_cert/satellite_cert_key.pem
```

If the command is successful, it returns the following **satellite-installer** command. You can use this command to deploy the renewed CA certificates to Satellite Server:

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



IMPORTANT

Do not delete the certificate files after you deploy the certificate. They are required when upgrading Satellite Server.



NOTE

If a new consumer package **katello-ca-consumer-latest.noarch.rpm** is generated due to a different Certificate Signing Authority, all the clients registered to Satellite Server must be updated.

Verification

- Access the Satellite web UI from your local machine. For example, <https://satellite.example.com>.

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

15.2. RENEWING A CUSTOM SSL CERTIFICATE ON CAPSULE SERVER

Use this procedure to update your custom SSL certificate for Capsule Server. The **satellite-installer** command, which the **capsule-certs-generate** command returns, is unique to each Capsule Server. You cannot use the same command on more than one Capsule Server.

Prerequisite

- You must create a new Certificate Signing Request and send it to the Certificate Authority to sign the certificate. Refer to the [Configuring Satellite Server with a Custom SSL Certificate](#) guide before creating a new CSR because the Satellite Server certificate must have X.509 v3 **Key Usage** and **Extended Key Usage** extensions with required values. In return, you will receive the Capsule Server certificate and CA bundle.

Procedure

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

```
# katello-certs-check -t capsule \
-b /root/capsule_cert/ca_cert_bundle.pem \
-c /root/capsule_cert/capsule_cert.pem \
-k /root/capsule_cert/capsule_cert_key.pem
```

2. On your Satellite Server, generate the certificate archive file for your Capsule Server:

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

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

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

You can move the copied file to the applicable path if required.

4. Retain a copy of the **satellite-installer** command that the **capsule-certs-generate** command returns for deploying the certificate to your Capsule Server.
5. Deploy the certificate on your Capsule Server using the **satellite-installer** command returned by the **capsule-certs-generate** command:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn "satellite.example.com" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-content-certs-tar "/root/My_Certificates/capsule.example.com-certs.tar" \
--certs-update-server
```



IMPORTANT

Do not delete the certificate archive file on the Capsule Server after you deploy the certificate. They are required when upgrading Capsule Server.



NOTE

If a new consumer package **katello-ca-consumer-latest.noarch.rpm** is generated due to a different Certificate Signing Authority, all the clients registered to Capsule Server must be updated.

CHAPTER 16. LOGGING AND REPORTING PROBLEMS

This chapter provides information on how to log and report problems in Satellite, including information on relevant log files, how to enable debug logging, how to open a support case and attach the relevant log tar files, and how to access support cases within the Satellite web UI.

You can use the log files and other information described in this chapter to do your own troubleshooting, or you can capture these and many more files, as well as diagnostic and configuration information, to send to Red Hat Support if you need further assistance.

For more information about Satellite logging settings, use **satellite-installer** with the **--full-help** option:

```
# satellite-installer --full-help | grep logging
```

16.1. ENABLING DEBUG LOGGING

Debug logging provides the most detailed log information and can help with troubleshooting issues that can arise with Satellite and its components. In the Satellite CLI, enable debug logging to log detailed debugging information for Satellite.

Procedure

1. To enable debug logging, enter the following command:

```
# satellite-installer --foreman-logging-level debug
```

2. After you complete debugging, reset the logging level to the default value:

```
# satellite-installer --reset-foreman-logging-level
```

16.2. INCREASING THE LOGGING LEVELS TO HELP WITH DEBUGGING

By default, Satellite comes with **:INFO** level logging enabled. You can increase or decrease the log levels on your Satellite.

Enabling debug level logging on all components

```
# hammer admin logging --all --level-debug
# satellite-maintain service restart
```

Enabling debug level logging for a specific component

```
# hammer admin logging --components "Component" --level-debug
```

Reverting debug level logging to INFO

```
# hammer admin logging --all --level-production
# satellite-maintain service restart
```

Listing all components and changed configuration files

```
# hammer admin logging --list
```

```
-----|-----|-----
COMPONENT | AUTO-DETECTED BY EXISTENCE OF | DESTINATIONS
-----|-----|-----
dhcpd      | /etc/dhcp/dhcpd.conf           | syslog /var/log/dhcpd-debug.log
postgresql | /var/lib/pgsql/data/postgresql.conf | syslog /var/lib/pgsql/data/pg_log/
proxy      | /etc/foreman-proxy/settings.yml  | /var/log/foreman-proxy/proxy.log
qpidd      | /etc/qpidd/qpidd.conf           | syslog
rails      | /etc/foreman/settings.yml        | /var/log/foreman/production.log
tomcat     | /etc/candlepin/candlepin.conf    | /var/log/candlepin/ /var/log/tomcat/
virt-who   | /etc/sysconfig/virt-who         | syslog
-----|-----|-----
```

16.2.1. Increasing the Logging Level For Hammer

You can find the log for Hammer in `~/.hammer/log/hammer.log`. Edit `/etc/hammer/cli_config.yml` and set the `:log_level:`:

```
:log_level: 'debug'
```

16.2.2. Increasing the Logging Level On Capsule

You can find the log for Capsule in `/var/log/foreman-proxy/proxy.log`. Uncomment the **DEBUG** line in `/etc/foreman-proxy/settings.yml`:

```
:log_level: DEBUG
```

Ensure to restart the **foreman-proxy** service afterwards:

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

CAUTION

Running the installer will revert this change back.

16.2.3. Increasing the Logging Level For Candlepin

You can find the log for Candlepin in `/var/log/candlepin/candlepin.log`. Errors are also logged to a separate file for easier debugging `/var/log/candlepin/error.log`.

Extend `/etc/candlepin/candlepin.conf`:

```
log4j.logger.org.candlepin=DEBUG
```

Ensure to restart the **tomcat** service afterwards:

```
# systemctl restart tomcat
```

If the candlepin log files are too verbose, you can decrease the default debug level:

```
log4j.logger.org.candlepin.resource.ConsumerResource=WARN
log4j.logger.org.candlepin.resource.HypervisorResource=WARN
```

16.2.4. Increasing the Logging Level On Satellite

You can find the log for Satellite in `/var/log/foreman/production.log`.

Satellite stores logs for Apache in:

- `/var/log/httpd/foreman_error.log`
- `/var/log/httpd/foreman_access.log`
- `/var/log/httpd/foreman_ssl_error.log`
- `/var/log/httpd/foreman_ssl_access.log`

Procedure

1. Set the logging level in `/etc/foreman/settings.yaml`:

```
:logging:
  :production:
    :type: file
    :layout: pattern
    :level: debug
```

2. Enable selected loggers in `/etc/foreman/settings.yaml`:

```
:loggers:
  :ldap:
    :enabled: true
  :permissions:
    :enabled: true
  :sql:
    :enabled: true
```

Note that to see logging from some area, debug logging has to be set.

3. Restart Satellite services:

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

You can find the complete list of loggers with their default values in `/usr/share/foreman/config/application.rb` in the `Foreman::Logging.add_loggers` command.

16.2.5. Increasing the Logging Level For Qpid Dispatch Router

Qpid logs to syslog and can be viewed in `/var/log/messages` or with `journalctl`. Enable debug logging in `/etc/qpid-dispatch/qdrouterd.conf`:

```
enable: debug+
```

Ensure to restart the Qpid Dispatch Router afterwards:

```
# systemctl restart qdrouterd
```

CAUTION

Running the installer will revert this change back.

16.2.6. Increasing the Logging Level For Qpid Broker

Qpid logs to syslog and can be viewed in **/var/log/messages** or with **journalctl**. Set the log level in **/etc/qpid/qpid.conf**:

```
log-enable=debug+
```

Ensure to restart the Qpid Broker afterwards:

```
# systemctl restart qpid
```

CAUTION

Running the installer will revert this change.

16.2.7. Increasing the Logging Level For Redis

You can find the log for Redis in **/var/log/redis/redis.log**. Set the log level in **/etc/opt/rh/rh-redis5/redis.conf**:

```
loglevel debug
```

Ensure to restart the Redis service afterwards:

```
# systemctl restart rh-redis5-redis
```

16.2.8. Increasing the Logging Level For Postgres

You can find the log for Postgres in **/var/lib/pgsql/data/log**. Uncomment the **log_statement** in **/var/lib/pgsql/data/postgresql.conf**:

```
log_statement = 'all'
```

Ensure to restart Satellite services afterwards:

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

CAUTION

Based on the size of your Satellite installation, this can cause disk space to fill up very quickly. Only turn this on if absolutely needed.

For more debug log settings, refer to the [Postgresql documentation](#).

16.2.9. Increasing the Logging Level For Satellite Installer

You can find the log files in `/var/log/foreman-installer/`. To increase the log level of the Satellite Installer during an install:

```
# satellite-installer --verbose-log-level debug
```

16.2.10. Increasing the Logging Level For Pulp

By default, Pulp logs to syslog and can be viewed in `/var/log/messages` or with `journalctl`. Add the following config to the `/etc/pulp/settings.py` file:

```
LOGGING = {"dynaconf_merge": True, "loggers": {"": {'handlers': ['console'], 'level': 'DEBUG'}}
```

Ensure to restart the Pulp services afterwards:

```
# systemctl restart \
pulpcore-api \
pulpcore-content \
pulpcore-resource-manager \
pulpcore-worker@1 \
pulpcore-worker@2 \
rh-redis5-redis
```

16.2.11. Increasing the Logging Level For Puppet Agent

You can increase the logging level for Puppet agent on your Satellite Server.

Procedure

1. Add the following line to the **[agent]** block in the `/etc/puppetlabs/puppet/puppet.conf` file:

```
[agent]
log_level = debug
```

You can find the logs in `/var/log/puppetlabs/puppet/`

16.2.12. Increasing the Logging Level For Puppet Server

You can increase the logging level for Puppet server on your Satellite Server.

Prerequisite

- Puppet must be enabled in your Satellite. For more information, see [Enabling Puppet Integration with Satellite](#) in *Managing Configurations Using Puppet Integration in Red Hat Satellite*.

Procedure

1. Add the following line to the **[master]** block in `/etc/puppetlabs/puppet/puppet.conf` file:

```
[master]  
log_level = debug
```

2. Restart the Puppet server:

```
# satellite-maintain service restart --only puppetserver
```

You can find the logs in `/var/log/puppetlabs/puppetserver/`.

16.3. RETRIEVING THE STATUS OF SERVICES

Procedure

1. In the Satellite web UI, navigate to **Administer > About**
2. On the **Smart Proxies** tab, you can view the status of all Capsules.
3. On the **Compute Resources** tab, you can view the status of attached compute resource providers.
4. In the **Backend System Status** table, you can view the status of all back-end services.

CLI procedure

- Run **hammer ping** to get information from the database and Satellite services:

```
# hammer ping
```

- Use **satellite-maintain** to check the status of the services running in systemd:

```
# satellite-maintain service status
```

- Use **satellite-maintain** to perform a health check:

```
$ satellite-maintain health check
```

16.4. RESTARTING SERVICES

Satellite uses a set of back-end services to perform tasks. If you experience an issue with your Satellite, check the status of Satellite services.

Procedure

- Use **satellite-maintain** to restart Satellite services:

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

TIP

Run **foreman-maintain --help** for more information.

16.5. ENABLING INDIVIDUAL LOGGERS

You can enable individual loggers for selective logging. Satellite uses the following loggers:

app

Logs web requests and all general application messages. Default value: true.

audit

Logs additional fact statistics, numbers of added, updated, and removed facts. Default value: true.

background

Logs information from the background processing component.

blob

Logs contents of rendered templates for auditing purposes.



IMPORTANT

The **blob** logger might contain sensitive data.

dynflow

Logs information from the Dynflow process.

ldap

Logs high level LDAP queries and LDAP operations. Default value: false.

notifications

Logs information from the notifications component.

permissions

Logs queries to user roles, filters, and permissions when loading pages. Default value: false.

sql

Logs SQL queries made through Rails ActiveRecord. Default value: false.

telemetry

Logs debugging information from telemetry.

templates

Logs information from the template renderer component.

Procedure

1. Enable the individual loggers that you want. For example, to enable **sql** and **ldap** loggers, enter the following command:

```
# satellite-installer \
--foreman-loggers ldap:true \
--foreman-loggers sql:true
```

2. Optional: To reset loggers to their default values, enter the following command:

```
# satellite-installer --reset-foreman-loggers
```

16.6. CONFIGURING LOGGING TO JOURNAL OR FILE-BASED LOGGING

Satellite uses file-based logging by default. You can use the **satellite-installer** command to reconfigure logging.

Procedure for configuring logging with Journal

1. Enter the following **satellite-installer** command to configure logging to the **journald** service:

```
# satellite-installer \
--foreman-logging-layout pattern \
--foreman-logging-type journald \
--foreman-proxy-log JOURNAL
```

2. Optional: To inspect the log messages, use the **journalctl** utility. For example:
 - **journalctl --unit foreman** and **journalctl --unit foreman-proxy** show messages for the **foreman** and **foreman-proxy** units
 - **journalctl REQUEST=request_ID** shows messages for a specified request

Procedure for configuring file-based logging

1. Enter the following **satellite-installer** command to configure file-based logging:

```
# satellite-installer \
--reset-foreman-logging-layout \
--reset-foreman-logging-type \
--reset-foreman-proxy-log
```

2. Optional: To inspect the log messages, view these files:
 - **/var/log/foreman/production.log**
 - **/var/log/foreman-proxy.log**

Additional resources

For more information about Journal, see [Viewing logs using the command line](#) in the *Red Hat Enterprise Linux 8 Configuring Basic System Settings Guide*.

16.7. LOG FILE DIRECTORIES PROVIDED BY SATELLITE

Red Hat Satellite provides system information in the form of notifications and log files.

Table 16.1. Log File Directories for Reporting and Troubleshooting

Log File Directories	Description of Log File Content
/var/log/candlepin	Subscription management
/var/log/foreman-installer	Installer

Log File Directories	Description of Log File Content
/var/log/foreman-maintain	Foreman maintain
/var/log/foreman-proxy	Foreman proxy
/var/log/foreman	Foreman
/var/log/httpd	Apache HTTP server
/var/log/messages	Various other log messages
/var/log/puppetlabs/puppet	Configuration management
/var/log/rhsm	Subscription management
/var/log/tomcat	Candlepin webservice logs

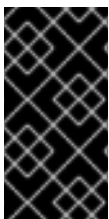
You can also use the **foreman-tail** command to follow many of the log files related to Satellite. You can run **foreman-tail -l** to list the processes and services that it follows.

16.8. UTILITIES FOR COLLECTING LOG INFORMATION

You can collect information from log files to troubleshoot Satellite.

sosreport

The **sosreport** command collects configuration and diagnostic information from a Linux system, such as the running Kernel version, loaded modules, running services, and system and service configuration files. This output is stored in a tar file located at **/var/tmp/sosreport-XXX-20171002230919.tar.xz**. For more information, run **sosreport --help** or see [What is a sosreport and how can I create one?](#).



IMPORTANT

The collection process removes security information such as passwords, tokens, and keys while collecting information. However, the tar files can still contain sensitive information about the Satellite Server. Red Hat recommends that you send this information directly to the intended recipient and not to a public target.

16.9. SYSTEM JOURNAL METADATA

The following table lists metadata that the **journald** service uses in Satellite. You can use this metadata to filter your queries.

Table 16.2. System Journal Metadata

Name	Description
AUDIT_ACTION	Audit action performed Example: Create, update, or delete
AUDIT_TYPE	Audit resource type Example: Host, Subnet, or ContentView
AUDIT_ID	Audit resource database ID as a number
AUDIT_ATTRIBUTE	Audit resource field or an updated database column
AUDIT_FIELD_OLD	Old audit value of an update action
AUDIT_FIELD_NEW	New audit value of an update action
AUDIT_ID	Record database ID of the audit subject
AUDIT_ATTRIBUTE	Attribute name or column on which an action was performed Example: Name or description
EXCEPTION_MESSAGE	Exception message when error is logged
EXCEPTION_CLASS	Exception Ruby class when error is logged
EXCEPTION_BACKTRACE	Exception backtrace as a multiline string when error is logged
LOC_ID	Location database ID
LOC_NAME	Location name
LOC_LABEL	Location label
LOGGER	Logger name To see the current list of loggers enabled by default, enter this command: <pre># awk '/add_loggers/,/^\\$/' /usr/share/foreman/config/application.rb</pre>
ORG_ID	Organization database ID
ORG_NAME	Organization name

Name	Description
ORG_LABEL	Organization label
REMOTE_IP	Remote IP address of a client
REQUEST	Request ID generated by the Action Dispatch module
SESSION	Random ID generated per session or a request for a sessionless request
TEMPLATE_NAME	Template name
TEMPLATE_DIGEST	Digest (SHA256) of rendered template contents
TEMPLATE_HOST_NAME	Host name for a rendered template if present
TEMPLATE_HOST_ID	Host database ID for a rendered template if present
USER_LOGIN	User login name

CHAPTER 17. MONITORING RESOURCES

The following chapter details how to configure monitoring and reporting for managed systems. This includes host configuration, Content Views, compliance, subscriptions, registered hosts, promotions, and synchronization.

17.1. USING THE RED HAT SATELLITE CONTENT DASHBOARD








The Red Hat Satellite content dashboard contains various widgets which provide an overview of the host configuration, Content Views, compliance reports, subscriptions and hosts currently registered, promotions and synchronization, and a list of the latest notifications.

In the Satellite web UI, navigate to **Monitor > Dashboard** to access the content dashboard. The dashboard can be rearranged by clicking on a widget and dragging it to a different position. The following widgets are available:

Host Configuration Status

An overview of the configuration states and the number of hosts associated with it during the last reporting interval. The following table shows the descriptions of the possible configuration states.

Table 17.1. Host Configuration States

Icon	State	Description
	Hosts that had performed modifications without error	Host that successfully performed modifications during the last reporting interval.
	Hosts in error state	Hosts on which an error was detected during the last reporting interval.
	Good host reports in the last 35 minutes	Hosts without error that did not perform any modifications in the last 35 minutes.
	Hosts that had pending changes	Hosts on which some resources would be applied but Puppet was configured to run in the noop mode.
	Out of sync hosts	Hosts that were not synchronized and the report was not received during the last reporting interval.
	Hosts with no reports	Hosts for which no reports were collected during the last reporting interval.
	Hosts with alerts disabled	Hosts which are not being monitored.

Click the particular configuration status to view hosts associated with it.

Host Configuration Chart

A pie chart shows the proportion of the configuration status and the percentage of all hosts associated with it.

Latest Events

A list of messages produced by hosts including administration information, product and subscription changes, and any errors.

Monitor this section for global notifications sent to all users and to detect any unusual activity or errors.

Run Distribution (last 30 minutes)

A graph shows the distribution of the running Puppet agents during the last puppet interval which is 30 minutes by default. In this case, each column represents a number of reports received from clients during 3 minutes.

New Hosts

A list of the recently created hosts. Click the host for more details.

Task Status

A summary of all current tasks, grouped by their state and result. Click the number to see the list of corresponding tasks.

Latest Warning/Error Tasks

A list of the latest tasks that have been stopped due to a warning or error. Click a task to see more details.

Discovered Hosts

A list of all bare-metal hosts detected on the provisioning network by the Discovery plug-in.

Latest Errata

A list of all errata available for hosts registered to Satellite.

Content Views

A list of all Content Views in Satellite and their publish status.




Sync Overview

An overview of all products or repositories enabled in Satellite and their synchronization status. All products that are in the queue for synchronization, are unsynchronized or have been previously synchronized are listed in this section.

Host Subscription Status

An overview of the subscriptions currently consumed by the hosts registered to Satellite. A subscription is a purchased certificate that unlocks access to software, upgrades, and security fixes for hosts. The following table shows the possible states of subscriptions.

Table 17.2. Host Subscription States

Icon	State	Description
	Invalid	Hosts that have products installed, but are not correctly subscribed. These hosts need attention immediately.
	Partial	Hosts that have a subscription and a valid entitlement, but are not using their full entitlements. These hosts should be monitored to ensure they are configured as expected.
	Valid	Hosts that have a valid entitlement and are using their full entitlements.

Click the subscription type to view hosts associated with subscriptions of the selected type.

Subscription Status

An overview of the current subscription totals that shows the number of active subscriptions, the number of subscriptions that expire in the next 120 days, and the number of subscriptions that have recently expired.

Host Collections

A list of all host collections in Satellite and their status, including the number of content hosts in each host collection.

Virt-who Configuration Status

An overview of the status of reports received from the **virt-who** daemon running on hosts in the environment. The following table shows the possible states.

Table 17.3. Virt-who Configuration States

State	Description
No Reports	No report has been received because either an error occurred during the virt-who configuration deployment, or the configuration has not been deployed yet, or virt-who cannot connect to Satellite during the scheduled interval.
No Change	No report has been received because hypervisor did not detect any changes on the virtual machines, or virt-who failed to upload the reports during the scheduled interval. If you added a virtual machine but the configuration is in the No Change state, check that virt-who is running.
OK	The report has been received without any errors during the scheduled interval.
Total Configurations	A total number of virt-who configurations.

Click the configuration status to see all configurations in this state.

The widget also lists the three latest configurations in the **No Change** state under **Latest Configurations Without Change**.

Latest Compliance Reports

A list of the latest compliance reports. Each compliance report shows a number of rules passed (P), failed (F), or othered (O). Click the host for the detailed compliance report. Click the policy for more details on that policy.

Compliance Reports Breakdown

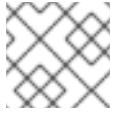
A pie chart shows the distribution of compliance reports according to their status.

Red Hat Insights Actions

Red Hat Insights is a tool embedded in Satellite that checks the environment and suggests actions you can take. The actions are divided into 4 categories: Availability, Stability, Performance, and Security.

Red Hat Insights Risk Summary

A table shows the distribution of the actions according to the risk levels. Risk level represents how critical the action is and how likely it is to cause an actual issue. The possible risk levels are: Low, Medium, High, and Critical.

**NOTE**

It is not possible to change the date format displayed in the Satellite web UI.

17.1.1. Managing Tasks

Red Hat Satellite keeps a complete log of all planned or performed tasks, such as repositories synchronised, errata applied, and Content Views published. To review the log, navigate to **Monitor > Tasks**.

In the Task window, you can search for specific tasks, view their status, details, and elapsed time since they started. You can also cancel and resume one or more tasks.

The tasks are managed using the Dynflow engine. Remote tasks have a timeout which can be adjusted as needed.

To Adjust Timeout Settings:

1. In the Satellite web UI, navigate to **Administer > Settings**.
2. Enter `%_timeout` in the search box and click **Search**. The search should return four settings, including a description.
3. In the **Value** column, click the icon next to a number to edit it.
4. Enter the desired value in seconds, and click **Save**.

**NOTE**

Adjusting the `%_finish_timeout` values might help in case of low bandwidth. Adjusting the `%_accept_timeout` values might help in case of high latency.

When a task is initialized, any back-end service that will be used in the task, such as Candlepin or Pulp, will be checked for correct functioning. If the check fails, you will receive an error similar to the following one:

There was an issue with the backend service candlepin: Connection refused – connect(2).

If the back-end service checking feature turns out to be causing any trouble, it can be disabled as follows.

To Disable Checking for Services:

1. In the Satellite web UI, navigate to **Administer > Settings**.
2. Enter `check_services_before_actions` in the search box and click **Search**.
3. In the **Value** column, click the icon to edit the value.
4. From the drop-down menu, select **false**.
5. Click **Save**.

17.2. CONFIGURING RSS NOTIFICATIONS

To view Satellite event notification alerts, click the **Notifications** icon in the upper right of the screen.

By default, the Notifications area displays RSS feed events published in the [Red Hat Satellite Blog](#).

The feed is refreshed every 12 hours and the Notifications area is updated whenever new events become available.

You can configure the RSS feed notifications by changing the URL feed. The supported feed format is RSS 2.0 and Atom. For an example of the RSS 2.0 feed structure, see the [Red Hat Satellite Blog feed](#). For an example of the Atom feed structure, see the [Foreman blog feed](#).

To Configure RSS Feed Notifications:

1. In the Satellite web UI, navigate to **Administer** > **Settings** and select the **Notifications** tab.
2. In the RSS URL row, click the edit icon in the **Value** column and type the required URL.
3. In the RSS enable row, click the edit icon in the **Value** column to enable or disable this feature.

17.3. MONITORING SATELLITE SERVER

Audit records list the changes made by all users on Satellite. This information can be used for maintenance and troubleshooting.

Procedure

1. In the Satellite web UI, navigate to **Monitor** > **Audits** to view the audit records.
2. To obtain a list of all the audit attributes, use the following command:

```
# foreman-rake audits:list_attributes
```

17.4. MONITORING CAPSULE SERVER

The following section shows how to use the Satellite web UI to find Capsule information valuable for maintenance and troubleshooting.

17.4.1. Viewing General Capsule Information

In the Satellite web UI, navigate to **Infrastructure** > **Capsules** to view a table of Capsule Servers registered to Satellite Server. The information contained in the table answers the following questions:

Is Capsule Server running?

This is indicated by a green icon in the **Status** column. A red icon indicates an inactive Capsule, use the **service foreman-proxy restart** command on Capsule Server to activate it.

What services are enabled on Capsule Server?

In the **Features** column you can verify if Capsule for example provides a DHCP service or acts as a Pulp mirror. Capsule features can be enabled during installation or configured in addition. For more information, see [Installing Capsule Server](#).

What organizations and locations is Capsule Server assigned to?

A Capsule Server can be assigned to multiple organizations and locations, but only Capsules belonging to the currently selected organization are displayed. To list all Capsules, select **Any Organization** from the context menu in the top left corner.

After changing the Capsule configuration, select **Refresh** from the drop-down menu in the **Actions** column to ensure the Capsule table is up to date.

Click the Capsule name to view further details. At the **Overview** tab, you can find the same information as in the Capsule table. In addition, you can answer to the following questions:

Which hosts are managed by Capsule Server?

The number of associated hosts is displayed next to the **Hosts managed** label. Click the number to view the details of associated hosts.

How much storage space is available on Capsule Server?

The amount of storage space occupied by the Pulp content in **/var/lib/pulp** is displayed. Also the remaining storage space available on the Capsule can be ascertained.

17.4.2. Monitoring Services

In the Satellite web UI, navigate to **Infrastructure > Capsules** and click the name of the selected Capsule. At the **Services** tab, you can find basic information on Capsule services, such as the list of DNS domains, or the number of Pulp workers. The appearance of the page depends on what services are enabled on Capsule Server. Services providing more detailed status information can have dedicated tabs at the Capsule page. For more information, see [Section 17.4.3, "Monitoring Puppet"](#).

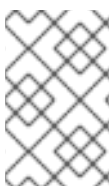
17.4.3. Monitoring Puppet

In the Satellite web UI, navigate to **Infrastructure > Capsules** and click the name of the selected Capsule. At the **Puppet** tab you can find the following:

- A summary of Puppet events, an overview of latest Puppet runs, and the synchronization status of associated hosts at the **General** sub-tab.
- A list of Puppet environments at the **Environments** sub-tab.

At the **Puppet CA** tab you can find the following:

- A certificate status overview and the number of autosign entries at the **General** sub-tab.
- A table of CA certificates associated with the Capsule at the **Certificates** sub-tab. Here you can inspect the certificate expiry data, or cancel the certificate by clicking **Revoke**.
- A list of autosign entries at the **Autosign entries** sub-tab. Here you can create an entry by clicking **New** or delete one by clicking **Delete**.



NOTE

The **Puppet** and **Puppet CA** tabs are available only if you have Puppet enabled in your Satellite. For more information, see [Enabling Puppet Integration with Satellite](#) in *Managing Configurations Using Puppet Integration in Red Hat Satellite*.

CHAPTER 18. USING WEBHOOKS

A webhook is a way for a web page or web application to provide other applications with information in real time. Webhooks are only triggered after an event occurs. The request usually contains details of the event. An event triggers callbacks, such as sending an e-mail confirming a host has been provisioned. Webhooks enable you to define a call to an external API based on Satellite internal event using a fire-and-forget message exchange pattern. The application sending the request does not wait for the response, or ignores it.

Payload of a webhook is created from webhook templates. Webhook templates use the same ERB syntax as Provisioning templates. Available variables:

- **@event_name**: Name of an event.
- **@webhook_id**: Unique event ID.
- **@payload**: Payload data, different for each event type. To access individual fields, use **@payload[:key_name]** Ruby hash syntax.
- **@payload[:object]**: Database object for events triggered by database actions (create, update, delete). Not available for custom events.
- **@payload[:context]**: Additional information as hash like request and session UUID, remote IP address, user, organization and location.

Because webhooks use HTTP, no new infrastructure needs be added to existing web services.

The typical use case for webhooks in Satellite is making a call to a monitoring system when a host is created or deleted.

Webhooks are useful where the action you want to perform in the external system can be achieved through its API. Where it is necessary to run additional commands or edit files, the shellhooks plugin for Capsules is available. The shellhooks plugin enables you to define a shell script on the Capsule that can be executed through the API.

You can use webhooks successfully without installing the shellhooks plugin.

For a list of available events, see [Available webhook events](#).

18.1. MIGRATING TO WEBHOOKS

The legacy **foreman_hooks** plugin provided full access to model objects that the webhooks plugin does not intentionally provide.

The scope of what is available is limited by the safemode and all objects and macros are both subject to an API stability promise and are fully documented.

The number of events triggered by webhooks is substantially fewer than with **foreman_hooks**.

Webhooks are processed asynchronously so there is minimal risk of tampering with internals of the system. It is not possible to migrate from **foreman_hooks** without creating payloads for each individual webhook script. However, the webhooks plugin comes with several example payload templates. You can also use the example payloads with shellhooks to simplify migration.

Both script and payload templates must be customized to achieve similar results.

18.2. INSTALLING WEBHOOKS

Use the following procedure to install webhooks. After installing webhooks, you can configure Satellite Server to send webhook requests.

Procedure

- Install webhooks using the following command:

```
# satellite-installer --enable-foreman-plugin-webhooks
```

- Optional: you can install the CLI plugin using the following command:

```
# satellite-installer --enable-foreman-cli-webhooks
```

18.3. CREATING A WEBHOOK TEMPLATE

Webhook templates are used to generate the body of HTTP request to a configured target when a webhook is triggered. Use the following procedure to create a webhook template in the Satellite web UI.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Webhook Templates**.
2. Click **Clone an existing template** or **Create Template**.
3. Enter a name for the template.
4. Use the editor to make changes to the template payload.
A webhook HTTP payload must be created using Satellite template syntax. The webhook template can use a special variable called **@object** that can represent the main object of the event. **@object** can be missing in case of certain events. You can determine what data are actually available with the **@payload** variable.

For more information, see [Template Writing Reference](#) in *Managing Hosts* and for available template macros and methods, visit [/templates_doc](#) on Satellite Server.

5. Optional: Enter the description and audit comment.
6. Assign organizations and locations.
7. Click **Submit**.

18.4. CREATING A WEBHOOK

You can customize events, payloads, HTTP authentication, content type, and headers through the Satellite web UI.

Use the following procedure to create a webhook in the Satellite web UI.

Procedure

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

2. Click **Create new**.
3. From the **Subscribe to** list, select an event.
4. Enter a **Name** for your webhook.
5. Enter a **Target URL**. Webhooks make HTTP requests to pre-configured URLs. The target URL can be a dynamic URL.
6. Click **Template** to select a template. Webhook templates are used to generate the body of the HTTP request to Satellite Server when a webhook is triggered.
7. Enter an HTTP method.
8. Optional: If you do not want activate the webhook when you create it, uncheck the **Enabled** flag.
9. Click the **Credentials** tab.
10. Optional: If HTTP authentication is required, enter **User** and **Password**.
11. Optional: Uncheck **Verify SSL** if you do not want to verify the server certificate against the system certificate store or Satellite CA.
12. On the **Additional** tab, enter the **HTTP Content Type**. For example, **application/json**, **application/xml** or **text/plain** on the payload you define. The application does not attempt to convert the content to match the specified content type.
13. Optional: Provide HTTP headers as JSON. ERB is also allowed.

When configuring webhooks with endpoints with non-standard HTTP or HTTPS ports, an SELinux port must be assigned, see [Configuring SELinux to Ensure Access to Satellite on Custom Ports](#) in *Installing Satellite Server in a Connected Network Environment*.

18.5. AVAILABLE WEBHOOK EVENTS

The following table contains a list of webhook events that are available from the Satellite web UI. **Action** events trigger webhooks only on **success**, so if an action fails, a webhook is not triggered.

For more information about payload, go to **Administer > About > Support > Templates DSL**. A list of available types is provided in the following table. Some events are marked as **custom**, in that case, the payload is an object object but a Ruby hash (key-value data structure) so syntax is different.

Event name	Description	Payload
Actions Katello Content View Promote Succeeded	A Content View was successfully promoted.	Actions::Katello::ContentView: :Promote
Actions Katello Content View Publish Succeeded	A repository was successfully synchronized.	Actions::Katello::ContentView: :Publish

