Red Hat Satellite 6.11

Administering Red Hat Satellite

A guide to administering Red Hat Satellite.

Last Updated: 2024-01-31

Red Hat Satellite 6.11 Administering Red Hat Satellite

A guide to administering Red Hat Satellite.

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

Legal Notice

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

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

You can submit feedback by filing a ticket in Bugzilla:

1. Navigate to the Bugzilla website.

2. In the **Component** field, use **Documentation**.

3. In the **Description** field, enter your suggestion for improvement. Include a link to the relevant parts of the 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. INSTALLING THE KATELLO ROOT CA CERTIFICATE

The first time you log on 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 installed in the browser. Use the following procedure to locate the root CA certificate on Satellite and to install it in your browser.

Prerequisites

- 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.

Importing the Katello Root CA Certificate Manually

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 ON TO SATELLITE

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

Prerequisites

- Ensure that the Katello root CA certificate is installed in your browser. For more information, see Section 1.1, “Installing 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 on, 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 *Any Organization* and the default location is *Any Location*. 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

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 on 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 on 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 8, *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 or Red Hat Enterprise Linux 7 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 or Red Hat Enterprise Linux 7 server to host the external databases.

Subscriptions for Red Hat Software Collections and 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.

Prerequisites

- 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:
   - For Red Hat Enterprise Linux 7:

     ```
     # subscription-manager repos --disable '*'
     # subscription-manager repos --enable=rhel-server-rhscl-7-rpms \
     --enable=rhel-7-server-rpms --enable=rhel-7-server-satellite-6.11-rpms
     ```

   - For Red Hat Enterprise Linux 8:

     ```
     # subscription-manager repos --disable '*'
     # subscription-manager repos \
     --enable=satellite-6.11-for-rhel-8-x86_64-rpms \
     --enable=rhel-8-for-x86_64-baseos-rpms \
     --enable=rhel-8-for-x86_64-appstream-rpms
     ```

3. On Red Hat Enterprise Linux 8, 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. You can install PostgreSQL using Red Hat Enterprise Linux 8 or Red Hat Enterprise Linux Server 7 repositories. Satellite supports PostgreSQL version 12.

- Installing PostgreSQL on Red Hat Enterprise Linux 8
- Installing PostgreSQL on Red Hat Enterprise Linux 7

3.3.1. Installing PostgreSQL on Red Hat Enterprise Linux 8

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
```

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/24   md5
```

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

```
# systemctl start postgresql
# systemctl enable 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"
    ```

3.3.2. Installing PostgreSQL on Red Hat Enterprise Linux 7

Procedure

1. To install PostgreSQL, enter the following command:

   ```
   # yum install rh-postgresql12-postgresql-server \
   rh-postgresql12-syspaths \
   rh-postgresql12-postgresql-evr
   ```

2. To initialize PostgreSQL, enter the following command:

   ```
   # postgresql-setup initdb
   ```

3. Edit the **/var/opt/rh/rh-postgresql12/lib/pgsql/data/postgresql.conf** file:

```
# vi /var/opt/rh/rh-postgresql12/lib/pgsql/data/postgresql.conf
```

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

   ```
   listen_addresses = '*'
   ```

5. Edit the **/var/opt/rh/rh-postgresql12/lib/pgsql/data/pg_hba.conf** file:

   ```
   # vi /var/opt/rh/rh-postgresql12/lib/pgsql/data/pg_hba.conf
   ```

6. Add the following line to the file:

   ```
     host  all   all   Satellite_ip/24   md5
   ```

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

   ```
   # systemctl start postgresql
   # systemctl enable 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.

Prerequisites

- 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.

Prerequisite

- Ensure that the Ansible 2.9 or later repository is enabled and the ansible package is updated:

```
# subscription-manager repos --enable rhel-7-server-ansible-2.9-rpms
# satellite-maintain packages update ansible
```

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 listing the content of the following directory:

```
# ls /usr/share/ansible/collections/ansible_collections/redhat/satellite/plugins/modules/
```

NOTE

At the time of writing, the **ansible-doc -l** command does not list collections yet.

Alternatively, you can also see the complete list of Satellite Ansible modules and other related information at Red Hat Ansible Automation Platform.

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 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 5.1, “User Management” 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 5.4, “Creating and Managing User Groups” .

Roles define a set of permissions and access levels. Each role contains one on 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 5.5, “Creating and Managing Roles”.

5.1. USER MANAGEMENT

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*.

5.1.1. 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 Chapter 14, *Configuring External Authentication*.

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**.

5.1.2. 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
```

5.1.3. 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.

5.1.4. Creating an API-Only User

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

Prerequisite

1. 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 5.3.1, “Creating a Personal Access Token”.

5.2. SSH KEY MANAGEMENT

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 the *Provisioning* guide.

For information on SSH keys and SSH key creation, see Using SSH-based Authentication in the *Red Hat Enterprise Linux 7 System Administrator's Guide*.

5.2.1. 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
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNtYAAABBBHHS2KmNyIYa27Qaa7
EHp+2l99ucGStx4P77e03ZvE3yVRJEFikpoP3MJtYYfIe8k 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
```

5.3. 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.

5.3.1. 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 you 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":"6.11.0","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.

5.3.2. 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"}
   }
   ```

5.4. CREATING AND MANAGING USER GROUPS

5.4.1. 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 Section 14.4, “Configuring External User Groups” .

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

5.4.2. 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
  ```

5.4.3. 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.

5.5. CREATING AND MANAGING ROLES

Satellite provides a set of predefined roles with permissions sufficient for standard tasks, as listed in Section 5.6, “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.

5.5.1. 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 5.5.3, “Adding Permissions to a Role”.

5.5.2. 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.

5.5.3. 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 5.7, "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.

5.5.4. 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 5.5.5, "Creating a Complete Permission Table" .

5.5.5. 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('')}</ul></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.

5.5.6. 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.

5.6. 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 5.5.4, “Viewing Permissions of a Role” .

Table 5.1. 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	All permissions except permissions for managing organizations. An administrator role defined per organization. The role has no visibility into resources in other organizations. By cloning this role and assigning an organization, you can delegate administration of that organization to a user.
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.

Role	Permissions Provided by Role
Site manager	A restrained version of the Manager role.
System admin	- 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. Users with this role can create users and assign all roles to them. Therefore, ensure to give this role only to trusted users.
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.

5.7. GRANULAR PERMISSION FILTERING

5.7.1. Granular Permission Filter

As mentioned in Section 5.5.3, “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.

5.7.2. 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 5.7.4, "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**.

5.7.3. 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.

5.7.3.1. 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 5.7.3.2, "Creating an Organization Specific Manager Role".

5.7.3.2. 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.

5.7.4. Supported Operators for Granular Search

Table 5.2. Logical Operators

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

Table 5.3. Symbolic Operators

Operator	Description
=	*Is equal to.* An equality comparison that is case-sensitive for text fields.

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

CHAPTER 6. 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.

The events that trigger a notification are the following:

- Host build
- Content View promotion
- Error reported by host
- Repository sync

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.

NOTE

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

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 from a Connected Network* .

6.1. CONFIGURING EMAIL NOTIFICATIONS

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.

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.

6.2. 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.

6.3. 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.

6.4. 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.

6.5. 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.

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.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **All Hosts**, locate the host that you want to view, and click **Edit** in the **Actions** column.
2. Go to the **Additional Information** tab. If the checkbox **Include this host within Satellite reporting** is checked, then the email notifications are enabled on that host.
3. Optional: Toggle the checkbox to enable or disable the email notifications.

   NOTE

   If you want to receive email notifications, ensure that you have an email address set in your user settings.

CHAPTER 7. MANAGING SECURITY COMPLIANCE

Security compliance management is the ongoing process of defining security policies, auditing systems 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.

With Satellite, you can schedule compliance auditing and reporting on all registered hosts.

7.1. SECURITY CONTENT AUTOMATION PROTOCOL

Satellite uses the Security Content Automation Protocol (SCAP) standard to define security policies.

SCAP is a framework of several specifications based on XML, such as checklists described in the Extensible Checklist Configuration Description Format (XCCDF) and vulnerabilities described in the Open Vulnerability and Assessment Language (OVAL). These specifications are encapsulated as *data stream* files.

Checklist items in XCCDF, also known as *rules*, express the desired configuration of a system item. For example, a rule 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 *XCCDF profiles*, which allows multiple profiles to share a rule.

The OpenSCAP scanner tool evaluates system items on a host against the rules and generates a report in the Asset Reporting Format (ARF), which is then returned to Satellite for monitoring and analysis.

Table 7.1. Specifications in the SCAP Framework 1.3 supported by the OpenSCAP scanner

Title	Description	Version
SCAP	Security Content Automation Protocol	1.3
XCCDF	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	2.0
CVSS	Common Vulnerability Scoring System	2.0

Additional resources

- For more information about SCAP, see the OpenSCAP project.

7.2. SCAP CONTENT IN SATELLITE

SCAP content is a SCAP data-stream file that contains implementation of compliance, configuration, or security baselines. A single data stream usually includes multiple XCCDF profiles. An XCCDF profile defines an industry standard or custom security standard against which you can evaluate compliance of host configuration in Satellite, such as Protection Profile for General Purpose Operating Systems (OSPP), Health Insurance Portability and Accountability Act (HIPAA), and PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 9. You can adapt existing XCCDF profiles according to your requirements using *tailoring files*.

In Satellite, you use an XCCDF profile from SCAP content and, eventually, a tailoring file, to define a *compliance policy*. Satellite includes default SCAP contents from SCAP Security Guide provided by the OpenSCAP project.

For more information on how to download, deploy, modify, and create your own content, see:

- *Red Hat Enterprise Linux 9 Security hardening*
- *Red Hat Enterprise Linux 8 Security hardening*
- *Red Hat Enterprise Linux 7 Security Guide*
- *Red Hat Enterprise Linux 6 Security Guide*

7.2.1. Supported SCAP Versions

Satellite supports content of SCAP versions 1.2 and 1.3.

7.3. COMPLIANCE POLICY DEPLOYMENT OPTIONS

You can use one of the following methods to deploy compliance policies:

Ansible deployment

You use an Ansible role to configure hosts for compliance scans.

Puppet deployment

You use a Puppet class and the Puppet agent to configure hosts for compliance scans.

Manual deployment

You manually configure hosts for compliance scans.

7.4. CONFIGURING COMPLIANCE POLICY DEPLOYMENT METHODS

Use one the following procedures to configure Satellite for the method that you have selected to deploy compliance policies. You will select one of these methods when you later create a compliance policy .

Procedure for Ansible deployment

1. Import the **theforeman.foreman_scap_client** Ansible role.
   For more information, see *Configuring Red Hat Satellite to use Ansible* .

2. Assign the created policy and the **theforeman.foreman_scap_client** Ansible role to a host or host group.

3. To trigger the deployment, run the Ansible role on the host or host group either manually, or set up a recurring job by using remote execution for regular policy updates.
   For more information, see Configuring and Setting Up Remote Jobs in *Managing Hosts*.

Procedure for Puppet deployment

1. Ensure Puppet is enabled.

2. Ensure the Puppet agent is installed on hosts.

3. Import the Puppet environment that contains the **foreman_scap_client** Puppet module.
   For more information, see *Managing Configurations Using Puppet Integration in Red Hat Satellite*.

4. Assign the created policy and the **foreman_scap_client** Puppet class to a host or host group.
   Puppet triggers the deployment on the next regular run or you can run Puppet manually. Puppet runs every 30 minutes by default.

Procedure for manual deployment

- For the manual deployment method, no additional Satellite configuration is required.
  For information on manual deployment, see How to set up OpenSCAP Policies using Manual Deployment option in the *Red Hat Knowledgebase*.

7.5. LISTING AVAILABLE SCAP CONTENTS

Use this procedure to view what SCAP contents are already loaded in Satellite. To use the CLI instead of the Satellite web UI, see the CLI procedure.

Prerequisite

- Your user account has the **view_scap_contents** permission.

Procedure

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

CLI procedure

- Run the following Hammer command on Satellite Server:

```
# hammer scap-content list \
--location "My_Location" \
--organization "My_Organization"
```

7.6. CONFIGURING SCAP CONTENTS

You can upload SCAP data streams and tailoring files to define compliance policies.

7.6.1. Loading the Default SCAP Contents

By loading the default SCAP contents on Satellite Server, you ensure that the data streams from the SCAP Security Guide (SSG) are loaded and assigned to all organizations and locations.

SSG is provided by the operating system of Satellite Server and installed in **/usr/share/xml/scap/ssg/content/**. Note that the available data streams depend on the operating system version on which Satellite runs. You can only use this SCAP content to scan hosts that have the same minor RHEL version as your Satellite Server. For more information, see Section 7.6.2, "Getting Supported SCAP Contents for RHEL".

Prerequisites

- Your user account has a role assigned that has the **create_scap_contents** permission.

Procedure

- Use the following Hammer command on Satellite Server:

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

7.6.2. Getting Supported SCAP Contents for RHEL

You can get the latest SCAP Security Guide (SSG) for Red Hat Enterprise Linux on the Red Hat Customer Portal. You have to get a version of SSG that is designated for the minor RHEL version of your hosts.

Procedure

1. Access the SCAP Security Guide in the package browser .
2. From the **Version** menu, select the latest SSG version for the minor version of RHEL that your hosts are running. For example, for RHEL 8.6, select a version named ***.el8_6**.
3. Download the package RPM.
4. Extract the data-stream file (***-ds.xml**) from the RPM. For example:

```
$ rpm2cpio scap-security-guide-0.1.69-3.el8_6.noarch.rpm \
| cpio -iv --to-stdout ./usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml \
> ssg-rhel-8.6-ds.xml
```

5. Upload the data stream to Satellite. For more information, see Section 7.6.3, "Uploading Additional SCAP Content".

Additional resources

- Supported versions of the SCAP Security Guide in RHEL in the *Red Hat Knowledgebase*
- SCAP Security Guide profiles supported in RHEL 9 in *Red Hat Enterprise Linux 9 Security hardening*
- SCAP Security Guide profiles supported in RHEL 8 in *Red Hat Enterprise Linux 8 Security hardening*
- SCAP Security Guide profiles supported in RHEL 7 in *Red Hat Enterprise Linux 7 Security Guide*

7.6.3. Uploading Additional SCAP Content

You can upload additional SCAP content into Satellite Server, either content created by yourself or obtained elsewhere. Note that Red Hat only provides support for SCAP content obtained from Red Hat. To use the CLI instead of the Satellite web UI, see the CLI procedure.

Prerequisite

- Your user account has the **create_scap_contents** permission.
- You have acquired a SCAP data-stream file.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Compliance** > **SCAP contents**.
2. Click **Upload New SCAP Content**.
3. Enter a title in the **Title** text box, such as ***My SCAP Content***.
4. In **Scap File**, click **Choose file**, navigate to the location containing a SCAP data-stream file and click **Open**.
5. On the **Locations** tab, select locations.
6. On the **Organizations** tab, select organizations.
7. Click **Submit**.

If the SCAP content file is loaded successfully, a message similar to **Successfully created *My SCAP Content*** is displayed.

CLI procedure

1. Place the SCAP data-stream file to a directory on your Satellite Server, such as ***/usr/share/xml/scap/my_content/***.
2. Run the following Hammer command on Satellite Server:

```
# hammer scap-content bulk-upload --type directory \
--directory /usr/share/xml/scap/my_content/ \
--location "My_Location" \
--organization "My_Organization"
```

Verification

- List the available SCAP contents . The list of SCAP contents includes the new title.

7.6.4. Tailoring XCCDF Profiles

You can customize existing XCCDF profiles using *tailoring files* without editing the original SCAP content. A single tailoring file can contain customizations of multiple XCCDF profiles.

You can create a tailoring file using the SCAP Workbench tool. For more information on using the SCAP Workbench tool, see Customizing SCAP Security Guide for your use case .

Then you can assign a tailoring file to a compliance policy to customize an XCCDF profile in the policy.

7.6.5. Uploading a Tailoring File

After uploading a tailoring file, you can apply it in a compliance policy to customize an XCCDF profile.

Prerequisite

- Your user account has the **create_tailoring_files** permission.

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 tailoring file and select **Open**.
4. Click **Submit** to upload the chosen tailoring file.

7.7. MANAGING COMPLIANCE POLICIES

A *compliance policy* is a scheduled audit that checks the specified hosts for compliance against a specific XCCDF profile from a SCAP content.

You specify the schedule for scans on Satellite Server and the scans are performed on hosts. When a scan completes, a report in ARF format is generated and uploaded to Satellite Server. The compliance policy makes no changes to the scanned host.

A compliance policy defines a SCAP client configuration and a cron schedule. The policy is then deployed together with the SCAP client on hosts to which the policy is assigned.

7.7.1. Compliance Policy

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.

7.7.2. Creating a Compliance Policy

By creating a compliance policy, you can define and plan your security compliance requirements, and ensure that your hosts remain compliant to your security policies.

Prerequisites

- You have configured Satellite for your selected compliance policy deployment method.
- You have available SCAP contents, and eventually tailoring files, in Satellite.
  - To verify what SCAP contents are available, see Section 7.5, “Listing Available SCAP Contents”.
  - To upload SCAP contents and tailoring files, see Section 7.6, “Configuring SCAP Contents”.

- Your user account has the **view_policies** and **create_policies** permissions.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Compliance – Policies**.
2. Click **New Policy** or **New Compliance Policy**.
3. Select the deployment method: **Ansible**, **Puppet**, or **Manual**. Then click **Next**.
4. Enter a name for this policy, a description (optional), then click **Next**.
5. Select the **SCAP Content** and **XCCDF Profile** to be applied, then click **Next**.
   Note that Satellite does not detect whether the selected XCCDF profile contains any rules. An empty XCCDF profile, such as the **Default XCCDF Profile**, will return empty reports.
6. Optional: To customize the XCCDF profile, select a **Tailoring File** and a **XCCDF Profile in Tailoring File**, then click **Next**.
7. Specify the scheduled time when the policy is to be applied. Select **Weekly**, **Monthly**, or **Custom** from the **Period** list. The **Custom** option allows for greater flexibility in the policy's schedule.
   - 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.
8. Select the locations to which to apply the policy, then click **Next**.
9. Select the organizations to which to apply the policy, then click **Next**.
10. Optional: Select the host groups to which to assign the policy.
11. Click **Submit**.

7.7.3. Viewing a Compliance Policy

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

Prerequisite

- Your user account has the **view_policies** permission.

Procedure

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

7.7.4. Editing a Compliance Policy

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

Puppet agent applies an edited policy to the host on the next run. By default, this occurs every 30 minutes. If you use Ansible, you must run the Ansible role manually again or have configured a recurring remote execution job that runs the Ansible role on hosts.

Prerequisite

- Your user account has the **view_policies** and **edit_policies** permissions.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Compliance – Policies**.
2. Click the name of the required policy.
3. Edit the necessary attributes.
4. Click **Submit**.

7.7.5. Deleting a Compliance Policy

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

Prerequisite

- Your user account has the **view_policies** and **destroy_policies** permissions.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Compliance – Policies**.
2. In the **Actions** column of the required policy, select **Delete** from the list.
3. Click **OK** in the confirmation message.

7.8. DEPLOYING COMPLIANCE POLICIES

To deploy a compliance policy, you must install the SCAP client, update the cron schedule file, and upload the SCAP content selected in the policy onto a host.

7.8.1. Deploying a Policy in a Host Group Using Ansible

After you deploy a compliance policy in a host group using Ansible, the Ansible role installs the SCAP client and configures OpenSCAP scans on the hosts according to the selected compliance policy.

Prerequisites

- You have enabled OpenSCAP on your Capsule. For more information, see Enabling OpenSCAP on Capsule Servers in *Installing Capsule Server*.
- You have enabled and synced the Satellite Client 6 repository to Satellite, and enabled it on the hosts.
- You have created a compliance policy with the Ansible deployment option and assigned the host group.

Procedure

1. In the Satellite web UI, navigate to **Configure** > **Host Groups**.
2. Click the host group that you want to configure for OpenSCAP reporting.
3. From the **OpenSCAP Capsule** list, select the Capsule with OpenSCAP enabled that you want to use.
4. On the **Ansible Roles** tab, assign the **theforeman.foreman_scap_client** Ansible role.
5. Optional: On the **Parameters** tab, configure any Ansible variables of the role.
6. Click **Submit** to save your changes.
7. In the row of the required host group, navigate to the **Actions** column and select **Run all Ansible roles**.

7.8.2. Deploying a Policy on a Host Using Ansible

After you deploy a compliance policy on a host using Ansible, the Ansible role installs the SCAP client and configures OpenSCAP scans on the host according to the selected compliance policy.

Prerequisites

- You have enabled OpenSCAP on your Capsule. For more information, see Enabling OpenSCAP on Capsule Servers in *Installing Capsule Server*.
- You have enabled and synced the Satellite Client 6 repository to Satellite, and enabled it on the host.
- You have created a compliance policy with the Ansible deployment option.

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 **OpenSCAP Capsule** list, select the Capsule with OpenSCAP enabled that you want to use.
3. On the **Ansible Roles** tab, add the **theforeman.foreman_scap_client** Ansible role.
4. Optional: On the **Parameters** tab, configure any Ansible variables of the role.
5. Click **Submit** to save your changes.
6. Click the **Hosts** breadcrumbs link to navigate back to the host index page.
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 **Assign Compliance Policy** window, select **Remember hosts selection for the next bulk action**.

11. Select the required policy from the list of available policies and click **Submit**.

12. Click **Select Action**.

13. Select **Run all Ansible roles** from the list.

7.8.3. Deploying a Policy in a Host Group Using Puppet

After you deploy a compliance policy in a host group using Puppet, the Puppet agent installs the SCAP client and configures OpenSCAP scans on the hosts on the next Puppet run according to the selected compliance policy.

Prerequisites

- You have enabled OpenSCAP on your Capsule. For more information, see Enabling OpenSCAP on Capsule Servers in *Installing Capsule Server*.

- You have enabled and synced the Satellite Client 6 repository to Satellite, and enabled it on the hosts.

- You have created a compliance policy with the Puppet deployment option and assigned the host group.

Procedure

1. In the Satellite web UI, navigate to **Configure** > **Host Groups**.

2. Click the host group that you want to configure for OpenSCAP reporting.

3. In the **Environment** list, select the Puppet environment that contains the **foreman_scap_client*** Puppet classes.

4. In the **OpenSCAP Capsule** list, select the Capsule with OpenSCAP enabled that you want to use.

5. On the **Puppet ENC** tab, add the **foreman_scap_client** Puppet class.

6. Optional: Configure any **Puppet Class Parameters**.

7. Click **Submit** to save your changes.

7.8.4. Deploying a Policy on a Host Using Puppet

After you deploy a compliance policy on a host using Puppet, the Puppet agent installs the SCAP client and configures OpenSCAP scans on the host on the next Puppet run according to the selected compliance policy.

Prerequisites

- You have enabled OpenSCAP on your Capsule. For more information, see Enabling OpenSCAP on Capsule Servers in *Installing Capsule Server*.

- You have enabled and synced the Satellite Client 6 repository to Satellite, and enabled it on the host.

- You have created a compliance policy with the Puppet deployment option.

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 **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. On the **Puppet ENC** tab, add the **foreman_scap_client** Puppet class.
5. Optional: Configure any **Puppet Class Parameters**.
6. Click the **Hosts** breadcrumbs link to navigate back to the host index page.
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 **Assign Compliance Policy** window, select **Remember hosts selection for the next bulk action**.
11. Select the required policy from the list of available policies and click **Submit**.

7.9. RUNNING A SECURITY COMPLIANCE SCAN ON DEMAND

Hosts perform OpenSCAP scans regularly by the CRON schedule defined in the compliance policies assigned to hosts. However, you can also run a scan on a host for all configured compliance policies manually at any time.

Prerequisites

- Your user account has the **view_hosts**, **create_job_invocations**, and **view_job_invocations** permissions.
- You have created a compliance policy and deployed it on the host.
  - For more information about managing policies, see Section 7.7, “Managing Compliance Policies”.
  - For more information about deploying policies, see Section 7.8, “Deploying Compliance Policies”.

Procedure

1. Navigate to **Hosts** > **All Hosts**.
2. Click the hostname of the required host.
3. On the host details page, expand the **Schedule a job** dropdown menu.
4. Select **Run OpenSCAP scan**

Verification

1. In the host details overview, locate the **Recent jobs** card.
2. Select the **Running** tab. Unless the job has already finished, the table shows a job called **Run scan for all OpenSCAP policies**.
3. On the **Recent jobs** card, select the **Finished** tab.
4. If the job has finished successfully, you should see the **succeeded** status in the row of the job.
5. Optional: Click the job name to review invocation details.

7.10. MONITORING COMPLIANCE

With Satellite, you can centralize 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. By monitoring compliance with SCAP, you can verify policy compliance and detect changes in compliance.

7.10.1. Searching Compliance Reports

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

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Reports**.
2. Optional: To see a list of available search parameters, click the empty **Search** field.
3. Enter the search query in the **Search** field and click **Search**. The search query is case insensitive.

Search Query Examples

Find all compliance reports for which more than five rules failed

```
failed > 5
```

Find all compliance reports created after January 1, 2023, for hosts with hostnames that contain prod-

```
host ~ prod- AND date > "Jan 1, 2023"
```

Find all reports generated by therhel7_audit compliance policy from an hour ago

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

Find reports that pass an XCCDF rule

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

Find reports that fail an XCCDF rule

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

Find 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

- You can create complex queries with the following logical operators: **and**, **not** and **has**. For more information about logical operators, see Supported Operators for Granular Search in *Administering Red Hat Satellite*.
- 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 Supported Operators for Granular Search in *Administering Red Hat Satellite*.
- You can bookmark a search to reuse the same search query. For more information, see Creating Bookmarks in *Administering Red Hat Satellite*.

7.10.2. Compliance Email Notifications

Satellite Server sends an OpenSCAP Summary email to all users who subscribe to the **Compliance policy summary** email notifications. For more information on subscribing to email notifications, see Section 6.1, "Configuring Email Notifications".

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.

7.10.3. Viewing Compliance Policy Statistics

You can view a compliance policy dashboard to verify compliance reports of a particular policy. The compliance policy dashboard provides a statistical summary of compliance of hosts and the ability to view report details for each host within the scope of that policy.

Consider prioritizing the following hosts when viewing compliance reports:

- Hosts which were evaluated as **Failed**
- Hosts labelled as **Never audited** because their status is unknown

Prerequisite

- Your user account has the **view_policies** permission.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Policies**.
2. In the row of the required policy, navigate to the **Actions** column and click **Dashboard**.

7.10.4. Examining Hosts per Rule Compliance Result

You can examine a simplified report and use policy rules to list hosts that have a certain compliance result, such as failing a particular rule.

Prerequisite

- Your user account has the **view_arf_reports** and **view_hosts** permissions.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Reports**.
2. In the **Reported At** column, navigate to the report of the required host and compliance policy, and click the time link.
3. Satellite displays a simplified list of policy rules with the results of the scan.
4. Optional: Filter the rules by check result. From the **Show log messages** dropdown list, select one of the following filters:
   - **Failed and othered** – to view rules that have failed or have not been checked during the scan,
   - **Failed only** – to view only rules that have failed.
5. Optional: Examine the details of the rule. In the **Message** column, click the icon next to the name of the rule.
6. In the row of the required rule, navigate to the **Actions** column and click **Hosts failing this rule**.

7.10.5. Examining Compliance Failures of a Host

You can examine a full compliance report, determine why a host failed compliance on a rule, and, in some cases, see how to remediate a case of non-compliance.

> **WARNING**
>
> Do not implement any of the recommended remedial actions or scripts without first testing them in a non-production environment. Remediation might render the system non-functional.

A compliance report consists of the following areas:

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

Prerequisite

- Your user account has the **view_arf_reports** and **view_hosts** permissions.

Procedure

1. In the Satellite web UI, navigate to **Hosts** > **Reports** to list all compliance reports.
2. In the row of the required host, navigate to the **Actions** column and click **Full Report** to view the complete details of an evaluation report.
3. Navigate to the **Evaluation Characteristics** area to review basic details about the evaluation of the host against a specific profile.
4. Navigate to the **Compliance and Scoring** area to review evaluation statistics and the host compliance score.
5. Navigate to the **Rule Overview** to examine the rules.
6. Optional: Deselect the check statuses that you want to hide, such as **pass**, **notapplicable**, or **fixed**.
7. Optional: From the **Group rule by** dropdown menu, select the criterion for the grouping of rules, such as **Severity**.
8. Optional: Enter a search string into the search field to filter rules by title. The search is case insensitive and applied dynamically as you type.
9. Click the title of a rule to inspect further result 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.

7.10.6. Deleting a Compliance Report

You can delete compliance reports on your Satellite.

Prerequisite

- Your user account has the **view_arf_reports** and **destroy_arf_reports** permissions.

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**.

7.10.7. 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 Deleting OpenSCAP Reports in the *API Guide*.

Prerequisite

- Your user account has the **view_arf_reports** and **destroy_arf_reports** permissions.

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.

CHAPTER 8. 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

The instances created using the backup tool are not supposed to run in parallel in a production environment. You must decommission any old instances after restoring the backup.

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 8.5, "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 System Backup and Recovery in the *Red Hat Enterprise Linux 7 System Administrator's Guide* .

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
```

8.1. 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:

   ### For Red Hat Enterprise Linux 8:

   ```
   # 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
   ```

   ### For Red Hat Enterprise Linux 7:

   ```
   # du -sh /var/opt/rh/rh-postgresql12/lib/pgsql/data /var/lib/pulp
   100G    /var/opt/rh/rh-postgresql12/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 8.1. Backup Data Compression Ratio for Red Hat Enterprise Linux 8

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

Table 8.2. Backup Data Compression Ratio for Red Hat Enterprise Linux 7

Data type	Directory	Ratio	Example results
PostgreSQL database files	**/var/opt/rh/rh-postgresql12/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.

8.2. 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 8.1, “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
  ```

8.3. 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. For production usecases, do not 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 8.1, “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
  ```

8.4. 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 8.1, “Estimating the Size of a Backup” .

Procedure

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

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

2. 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
   ```

3. 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
   ```

4. 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
   ```

8.5. 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.

8.6. 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 8.7, "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 8.1, "Estimating the Size of a Backup" .

Procedure

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

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

8.7. 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/opt/rh/rh-postgresql12/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.

8.8. 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 9. 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 8, *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.

9.1. 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 -Rnv /
  ```

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**.

9.2. 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 9.1, "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**.

9.3. 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 9.3.1, “Synchronizing an External Capsule” to synchronize from Satellite.

9.3.1. 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 10. RENAMING SATELLITE SERVER OR CAPSULE SERVER

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

If you rename Satellite Server, you must reregister all Satellite clients and configure each Capsule Server to point them to the new Satellite host name. If you use custom SSL certificates, you must regenerate them with the new host name. If you use virt-who, you must update the virt-who configuration files with the new host name.

If you rename Capsule Server, you must reregister all Capsule clients and update the Capsule host name in the Satellite web UI. If you use custom SSL certificates, you must regenerate them with the new host name.

> **WARNING**
>
> The renaming process shuts down all Satellite Server services on the host being renamed. When the renaming is complete, all services are restarted.

10.1. 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 you update all references to the new host name.

If you use external authentication, you must reconfigure Satellite Server for external authentication after you run the **satellite-change-hostname** script. The **satellite-change-hostname** script breaks external authentication for Satellite Server. For more information about configuring external authentication, see Chapter 14, *Configuring External Authentication*.

If you use virt-who, you must update the virt-who configuration files with the new host name after you run the **satellite-change-hostname** script. For more information, see Modifying a virt-who Configuration in *Configuring Virtual Machine Subscriptions in Red Hat Satellite* .

Prerequisites

- Both the **hostname** and **hostname -f** commands must return the FQDN of Satellite Server or 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** ***old_fqdn*** to set the old FQDN correctly before attempting to use the **satellite-change-hostname** script.
- Perform a backup of Satellite Server before changing a host name. If the renaming process is not successful, you must restore it from a backup. For more information, see Chapter 8, *Backing Up Satellite Server and Capsule Server*.
- Optional: If Satellite Server has a custom SSL certificate installed, a new certificate must be obtained for the host's new name. For more information, see Configuring Satellite Server with a Custom SSL Certificate in *Installing Satellite Server from a Connected Network* .

Procedure

1. On Satellite Server, choose the appropriate method to run the **satellite-change-hostname** script, providing the new host name and Satellite credentials:

    - If your Satellite Server is installed with 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. Optional: 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 from a Connected Network* .

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 capsule_id_number
    ```

10.2. 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 you update all references to the new host name.

NOTE

- Both the **hostname** and **hostname -f** commands must return the FQDN of Capsule Server or 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 *old_fqdn*** to set the old FQDN correctly before attempting to use the **satellite-change-hostname** script.

Prerequisites

- Backup Capsule Server. The **satellite-change-hostname** script makes irreversible changes to Capsule Server. If the renaming process is not successful, you must restore it from a backup. Perform a backup before changing a host name. For more information, see Chapter 8, *Backing Up Satellite Server and Capsule Server*.

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, providing the **root** user's password when prompted. In this example the archive file is copied to the **root** user's home directory, but you may prefer to copy it elsewhere.

   ```
   # scp /root/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 filename.

   ```
   # satellite-change-hostname new-capsule.example.com --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. Optional: If you have created a custom certificate for Capsule Server, on Capsule Server, to deploy the certificate, enter the **satellite-installer** command that the **capsule-certs-generate** command returns. 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 remote execution feature to perform this step. For more information, see Configuring and Setting up Remote Jobs in the *Managing Hosts Guide*.

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

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

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

# subscription-manager refresh
```

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

7. Locate Capsule Server in the list, and click **Edit** to the right of it.

8. Edit the **Name** and **URL** fields to match Capsule Server’s new host name, then click **Submit**.

9. On your DNS server, add a record for Capsule Server’s new host name, and delete the record for the previous host name.

CHAPTER 11. 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.

11.1. 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
```

11.2. 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
```

11.3. 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
```

11.4. 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 reenable regular task cleanup on Satellite, enter the following command:

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

11.5. 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'
```

11.6. 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.

11.7. 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 File System on a Logical Volume in the *Red Hat Enterprise Linux 7 Logical Volume Manager Administration Guide*.

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.

11.8. 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
```

11.9. 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
   ```

11.10. RECLAIMING SPACE FROM ON DEMAND REPOSITORIES

If you set the *download policy* to on demand, Satellite downloads packages only when the clients request them. You can clean up these packages to reclaim space.

For a single repository

- In the Satellite web UI, navigate to **Content** > **Products**.
- Select a product.

- On the **Repositories** tab, click the repository name.
- From the **Select Actions** list, select **Reclaim Space**.

For multiple repositories

- In the Satellite web UI, navigate to **Content** > **Products**.
- Select the product name.
- On the **Repositories** tab, select the checkbox of the repositories.
- Click **Reclaim Space** at the top right corner.

For Capsules

- In the Satellite web UI, navigate to **Infrastructure** > **Capsules**.
- Select the Capsule Server.
- Click **Reclaim space**.

CHAPTER 12. 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.

12.1. 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

1. 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.

12.2. 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-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--certs-tar-file "/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 13. 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
```

13.1. 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
   ```

13.2. 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/qpid/qpidd.conf                | syslog
rails      | /etc/foreman/settings.yaml          | /var/log/foreman/production.log
tomcat     | /etc/candlepin/candlepin.conf       | /var/log/candlepin/ /var/log/tomcat/
virt-who   | /etc/sysconfig/virt-who             | syslog
-----------|-------------------------------------|-------------------------------------
```

13.2.1. 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'
```

13.2.2. 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.

13.2.3. 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
```

13.2.4. 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.

13.2.5. 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.

13.2.6. 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/qpidd.conf**:

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

Ensure to restart the Qpid Broker afterwards:

```
# systemctl restart qpidd
```

CAUTION

Running the installer will revert this change.

13.2.7. 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
```

13.2.8. Increasing the Logging Level For Postgres

You can find the log for Postgres in **/var/opt/rh/rh-postgresql12/lib/pgsql/data/log/**. Uncomment the **log_statement** in **/var/opt/rh/rh-postgresql12/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.

13.2.9. 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
```

13.2.10. 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
```

13.2.11. 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/**

13.2.12. 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/**.

13.3. 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
```

13.4. RESTARTING SERVICES

Satellite uses a set of back-end services to perform tasks. You 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.

13.5. 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
```

13.6. 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*.

13.7. LOG FILE DIRECTORIES PROVIDED BY SATELLITE

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

Table 13.1. 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.

13.8. 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.

13.9. 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 13.2. 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: `# awk '/add_loggers/,/^$/' /usr/share/foreman/config/application.rb`
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 14. CONFIGURING EXTERNAL AUTHENTICATION

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

Important User and Group Account Information

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

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

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

Scenarios for Configuring External Authentication

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

- Using *Lightweight Directory Access Protocol* (LDAP) server as an external identity provider. LDAP is a set of open protocols used to access centrally stored information over a network. With Satellite, you can manage LDAP entirely through the Satellite web UI. For more information, see Section 14.1, “Using LDAP”. Though you can use LDAP to connect to a Red Hat Identity Management or AD server, the setup does not support server discovery, cross-forest trusts, or single sign-on with Kerberos in Satellite’s web UI.

- Using a Red Hat Identity Management server as an external identity provider. Red Hat Identity Management deals with the management of individual identities, their credentials and privileges used in a networking environment. Configuration using Red Hat Identity Management cannot be completed using only the Satellite web UI and requires some interaction with the CLI. For more information see Section 14.2, “Using Red Hat Identity Management” .

- Using *Active Directory* (AD) integrated with Red Hat Identity Management through cross-forest Kerberos trust as an external identity provider. For more information see Section 14.3.5, “Active Directory with Cross-Forest Trust”.

- Using Red Hat Single Sign-On as an OpenID provider for external authentication to Satellite. For more information, see Section 14.8, “Configuring Satellite with Red Hat Single Sign-On Authentication”.

- Using Red Hat Single Sign-On as an OpenID provider for external authentication to Satellite with TOTP. For more information, see Section 14.9, “Configuring Red Hat Single Sign-On Authentication with TOTP”.

As well as providing access to Satellite Server, hosts provisioned with Satellite can also be integrated with Red Hat Identity Management realms. Red Hat Satellite has a realm feature that automatically manages the life cycle of any system registered to a realm or domain provider. For more information, see Section 14.7, “External Authentication for Provisioned Hosts” .

Table 14.1. Authentication Overview

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

14.1. USING LDAP

Satellite supports LDAP authentication using one or multiple LDAP directories.

If you require Red Hat Satellite to use **TLS** to establish a secure LDAP connection (LDAPS), first obtain certificates used by the LDAP server you are connecting to and mark them as trusted on the base operating system of your Satellite Server as described below. If your LDAP server uses a certificate chain with intermediate certificate authorities, all of the root and intermediate certificates in the chain must be trusted, so ensure all certificates are obtained. If you do not require secure LDAP at this time, proceed to Section 14.1.2, “Configuring Red Hat Satellite to use LDAP” .

Using SSSD Configuration

Though direct LDAP integration is covered in this section, Red Hat recommends that you use SSSD and configure it against Red Hat Identity Management, AD, or an LDAP server. SSSD improves the consistency of the authentication process. For more information about the preferred configurations, see Section 14.3, “Using Active Directory” . You can also cache the SSSD credentials and use them for LDAP authentication. For more information on SSSD, see Configuring SSSD in the *Red Hat Enterprise Linux 7 System-Level Authentication Guide*.

IMPORTANT

Users cannot use both Red Hat Identity Management and LDAP as an authentication method. Once a user authenticates using one method, they cannot use the other method.

To change the authentication method for a user, you have to remove the automatically created user from Satellite.

For more information on using Red Hat Identity Management as an authentication method, see Section 14.2, “Using Red Hat Identity Management” .

14.1.1. Configuring TLS for Secure LDAP

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

Procedure

1. Obtain the Certificate from the LDAP Server.

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

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

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

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

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

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

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

   c. Restart the **httpd** service:

      ```
      # systemctl restart httpd
      ```

14.1.2. Configuring Red Hat Satellite to use LDAP

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

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

- Section 14.2, “Using Red Hat Identity Management”
- Section 14.3, “Using Active Directory”

Procedure

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

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

2. In the Satellite web UI, navigate to **Administer** > **Authentication Sources**.

3. Click **Create LDAP Authentication Source**

4. On the **LDAP server** tab, enter the LDAP server’s name, host name, port, and server type. The default port is 389, the default server type is POSIX (alternatively you can select FreeIPA or Active Directory depending on the type of authentication server). For **TLS** encrypted connections, select the **LDAPS** checkbox to enable encryption. The port should change to 636, which is the default for LDAPS.

5. On the **Account** tab, enter the account information and domain name details. See Section 14.1.3, "Description of LDAP Settings" for descriptions and examples.

6. On the **Attribute mappings** tab, map LDAP attributes to Satellite attributes. You can map login name, first name, last name, email address, and photo attributes. See Section 14.1.4, "Example Settings for LDAP Connections" for examples.

7. On the **Locations** tab, select locations from the left table. Selected locations are assigned to users created from the LDAP authentication source, and available after their first login.

8. On the **Organizations** tab, select organizations from the left table. Selected organizations are assigned to users created from the LDAP authentication source, and available after their first login.

9. Click **Submit**.

10. Configure new accounts for LDAP users:

    - If you did not select **Automatically Create Accounts In Satellite**checkbox, see Section 5.1.1, "Creating a User" to create user accounts manually.

    - If you selected the **Automatically Create Accounts In Satellite**checkbox, LDAP users can now log in to Satellite using their LDAP accounts and passwords. After they log in for the first time, the Satellite administrator has to assign roles to them manually. See Section 5.1.2, "Assigning Roles to a User" to assign user accounts appropriate roles in Satellite.

14.1.3. Description of LDAP Settings

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

Table 14.2. Account Tab Settings

Setting	Description
Account	The user name of the LDAP account that has read access to the LDAP server. User name is not required if the server allows anonymous reading, otherwise use the full path to the user's object. For example: `uid=$login,cn=users,cn=accounts,dc=example,dc=com` The **$login** variable stores the username entered on the login page as a literal string. The value is accessed when the variable is expanded. The variable cannot be used with external user groups from an LDAP source because Satellite needs to retrieve the group list without the user logging in. Use either an anonymous, or dedicated service user.
Account password	The LDAP password for the user defined in the **Account username** field. This field can remain blank if the **Account username** is using the **$login** variable.
Base DN	The top level domain name of the LDAP directory.
Groups base DN	The top level domain name of the LDAP directory tree that contains groups.

Setting	Description
LDAP filter	A filter to restrict LDAP queries.
Automatically Create Accounts In Satellite	If this checkbox is selected, Satellite creates user accounts for LDAP users when they log in to Satellite for the first time. After they log in for the first time, the Satellite administrator has to assign roles to them manually. See Section 5.1.2, "Assigning Roles to a User" to assign user accounts appropriate roles in Satellite.
Usergroup Sync	If this option is selected, the user group membership of a user is automatically synchronized when the user logs in, which ensures the membership is always up to date. If this option is cleared, Satellite relies on a cron job to regularly synchronize group membership (every 30 minutes by default). For more information, see Section 14.4, "Configuring External User Groups".

14.1.4. Example Settings for LDAP Connections

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

Table 14.3. Example Settings for Active Directory, Free IPA or Red Hat Identity Management and POSIX LDAP Connections

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

Setting	Active Directory	FreeIPA or Red Hat Identity Management	POSIX (OpenLDAP)
Email address attribute	mail	mail	mail
Photo attribute	thumbnailPhoto	-	-

NOTE

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

14.1.5. Example LDAP Filters

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

Table 14.4. Example filters for allowing specific users to login

User	Filter
User1	(distinguishedName=cn=User1,cn=Users,dc=domain,dc=example)
User1, User3	(memberOf=cn=Group1,cn=Users,dc=domain,dc=example)
User2, User3	(memberOf=cn=Group2,cn=Users,dc=domain,dc=example)
User1, User2, User3	(|(memberOf=cn=Group1,cn=Users,dc=domain,dc=example) (memberOf=cn=Group2,cn=Users,dc=domain,dc=example))
User1, User2, User3	(memberOf:1.2.840.113556.1.4.1941:=cn=Users,dc=domain,dc=example)

NOTE

Group **Users** is a nested group that contains groups **Group1** and **Group2**. If you want to filter all users from a nested group, you must add **memberOf:1.2.840.113556.1.4.1941:=** before the nested group name. See the last example in the table above.

LDAP directory structure

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

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

LDAP group membership

The group membership that the filters in the example use:

Group	Members
Group1	User1, User3
Group2	User2, User3

14.2. USING RED HAT IDENTITY MANAGEMENT

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

NOTE

You can attach Red Hat Identity Management as an external authentication source with no single sign-on support. For more information, see Section 14.1, “Using LDAP”.

IMPORTANT

Users cannot use both Red Hat Identity Management and LDAP as an authentication method. Once a user authenticates using one method, they cannot use the other method.

To change the authentication method for a user, you have to remove the automatically created user from Satellite.

Prerequisite

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

The examples in this chapter assume separation between Red Hat Identity Management and Satellite configuration. However, if you have administrator privileges for both servers, you can configure Red Hat Identity Management as described in Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide.

14.2.1. Configuring Red Hat Identity Management Authentication on Satellite Server

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

Procedure

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

   ```
   # kinit admin
   ```

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

   ```
   # klist
   ```

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

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

   NOTE

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

   For more information on host configuration properties, see About Host Entry Configuration Properties in the *Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy* guide.

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

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

   For more information on managing services, see Managing Services in the *Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy* guide.

5. On Satellite Server, install the IPA client:

   WARNING

   This command might restart Satellite services during the installation of the package. For more information about installing and updating packages on Satellite, see Section 11.8, "Managing Packages on the Base Operating System of Satellite Server or Capsule Server".

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

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

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

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

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

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

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

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

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

9. Restart Satellite services:

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

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

14.2.2. Configuring Host-Based Authentication Control

HBAC rules define which machine within the domain a Red Hat Identity Management user is allowed to access. You can configure HBAC on the Red Hat Identity Management server to prevent selected users from accessing Satellite Server. With this approach, you can prevent Satellite from creating database entries for users that are not allowed to log in. For more information on HBAC, see Configuring Host-Based Access Control in the *Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy* guide.

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

Procedure

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

   ```
   # kinit admin
   ```

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

   ```
   # klist
   ```

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

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

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

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

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

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

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

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

7. Configure the Red Hat Identity Management integration with Satellite Server as described in Section 14.2.1, “Configuring Red Hat Identity Management Authentication on Satellite Server” . On Satellite Server, define the PAM service as root:

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

14.3. USING ACTIVE DIRECTORY

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

NOTE

You can attach Active Directory as an external authentication source with no single sign-on support. For more information, see Section 14.1, “Using LDAP”. For an example configuration, see How to configure Active Directory authentication with TLS on Satellite.

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

- Enrolling Satellite Server with the Active Directory server as described in Section 14.3.2, “Enrolling Satellite Server with the AD Server”.

- Configuring direct Active Directory integration with GSS-proxy as described in Section 14.3.3, “Configuring Direct AD Integration with GSS-Proxy”.

14.3.1. GSS-Proxy

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

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

14.3.2. Enrolling Satellite Server with the AD Server

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

Prerequisite

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

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

Procedure

1. Install the required packages:

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

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

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

14.3.3. Configuring Direct AD Integration with GSS-Proxy

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

Prerequisite

- Satellite is enrolled with the Active Directory server. For more information, see Section 14.3.2, “Enrolling Satellite Server with the AD Server”.

Procedure

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

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

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

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

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

   ```
   [global]
   workgroup = EXAMPLE
   ```

```
realm = EXAMPLE.ORG
kerberos method = system keytab
security = ads
```

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

   ```
   # id apache
   ```

   Apache user must not have access to the keytab file.

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

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

6. Create a keytab entry:

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

7. Enable IPA authentication in Satellite:

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

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

   ```
   # systemctl restart gssproxy.service
   # systemctl enable gssproxy.service
   ```

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

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

10. Apply changes to the service:

    ```
    # systemctl daemon-reload
    ```

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

    ```
    # systemctl restart httpd.service
    ```

IMPORTANT

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

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

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

Verification

Verify that SSO is working as expected.

With a running Apache server, users making HTTP requests against the server are authenticated if the client has a valid Kerberos ticket.

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

```
# kinit ldapuser
```

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

```
# klist
```

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

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

This returns the following response:

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

14.3.4. Kerberos Configuration in Web Browsers

For information on configuring the Firefox browser see Configuring Firefox to Use Kerberos for Single Sign-On in the *Red Hat Enterprise Linux System-Level Authentication* guide.

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

14.3.5. Active Directory with Cross-Forest Trust

Kerberos can create **cross-forest trust** that defines a relationship between two otherwise separate domain forests. A domain forest is a hierarchical structure of domains; both AD and Red Hat Identity Management constitute a forest. With a trust relationship enabled between AD and Red Hat Identity

Management, users of AD can access Linux hosts and services using a single set of credentials. For more information on cross-forest trusts, see Creating Cross-forest Trusts with Active Directory and Identity Management in the *Red Hat Enterprise Linux Windows Integration* guide.

From the Satellite point of view, the configuration process is the same as integration with Red Hat Identity Management server without cross-forest trust configured. Satellite Server has to be enrolled in the IdM domain and integrated as described in Section 14.2, "Using Red Hat Identity Management" .

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

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

Procedure

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

```
[nss]
user_attributes=+mail, +sn, +givenname
[domain/EXAMPLE.com]
...
krb5_store_password_if_offline = True
ldap_user_extra_attrs=email:mail, lastname:sn, firstname:givenname

[ifp]
allowed_uids = ipaapi, root
user_attributes=+email, +firstname, +lastname
```

   - Verify the AD attributes value.

```
# dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe
/org/freedesktop/sssd/infopipe org.freedesktop.sssd.infopipe.GetUserAttr string:ad-
user@ad-domain array:string:email,firstname,lastname
```

14.4. CONFIGURING EXTERNAL USER GROUPS

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

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

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

Prerequisites

- If you use an LDAP server, configure Satellite to use LDAP authentication. For more information see Section 14.1, "Using LDAP".
  When using external user groups from an LDAP source, you cannot use the **$login** variable as a substitute for the account user name. You must use either an anonymous or dedicated service user.

- If you use a Red Hat Identity Management or AD server, configure Satellite to use Red Hat Identity Management or AD authentication. For more information, see Chapter 14, *Configuring External Authentication*.

- Ensure that at least one external user authenticates for the first time.

- Retain a copy of the external group names you want to use. To find the group membership of external users, enter the following command:

  ```
  # id username
  ```

Procedure

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

2. Specify the name of the new user group. Do not select any users to avoid adding users automatically when you refresh the external user group.

3. Click the **Roles** tab and select the roles you want to assign to the user group. Alternatively, select the **Administrator** checkbox to assign all available permissions.

4. Click the **External groups** tab, then click **Add external user group**, and select an authentication source from the **Auth source** drop-down menu.
   Specify the exact name of the external group in the **Name** field.

5. Click **Submit**.

14.5. REFRESHING EXTERNAL USER GROUPS FOR LDAP

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

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

Use this procedure to refresh the LDAP source manually.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Usergroups** and select a user group.

2. On the **External Groups** tab, click **Refresh** to the right of the required user group.

CLI procedure

- Enter the following command:

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

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

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

14.7. EXTERNAL AUTHENTICATION FOR PROVISIONED HOSTS

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

Prerequisites

- Satellite Server that is registered to the Content Delivery Network or an external Capsule Server that is registered to Satellite Server.

- A deployed realm or domain provider such as Red Hat Identity Management.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Procedure

1. In the Satellite web UI, navigate to **Infrastructure** > **Realms** and click **Create Realm**.

2. In the **Name** field, enter a name for the realm.

3. From the **Realm Type** list, select the type of realm.

4. From the **Realm Capsule** list, select Capsule Server where you have configured Red Hat Identity Management.

5. Click the **Locations** tab and from the **Locations** list, select the location where you want to add the new realm.

6. Click the **Organizations** tab and from the **Organizations** list, select the organization where you want to add the new realm.

7. Click **Submit**.

Updating Host Groups with Realm Information

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

1. In the Satellite web UI, navigate to **Configure** > **Host Groups**, select the host group that you want to update, and click the **Network** tab.

2. From the **Realm** list, select the realm you create as part of this procedure, and then click **Submit**.

Adding Hosts to a Red Hat Identity Management Host Group

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

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

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

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

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

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

2. Create an **automembership** rule:

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

   Where you can use the following options:

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

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

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

```
----------------------------
Number of conditions added 1
----------------------------
```

Where you can use the following options:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Procedure

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

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

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

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

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

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

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

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

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

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

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

3. Restart the **httpd** service:

   ```
   # systemctl restart httpd
   ```

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

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

Procedure

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

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

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

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

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

11. In the **Name** field, enter a name for the group mapper.

12. From the **Mapper Type** list, select **Group Membership**.

13. In the **Token Claim Name** field, enter **groups**.

14. Set the **Full group path** setting to OFF.

15. Click **Save**.

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

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

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

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

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

Prerequisite

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

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **Authentication** tab.

2. Locate the **Authorize login delegation** row, and in the **Value** column, set the value to **Yes**.

3. Locate the **Authorize login delegation auth source user autocreate** row, and in the **Value** column, set the value to **External**.

4. Locate the **Login delegation logout URL** row, and in the **Value** column, set the value to **https://satellite.example.com/users/extlogout**.

5. Locate the **OIDC Algorithm** row, and in the **Value** column, set the algorithm for encoding on Red Hat Single Sign-On to **RS256**.

6. Locate the **OIDC Audience** row, and in the **Value** column, set the value to the client ID for Red Hat Single Sign-On.

7. Locate the **OIDC Issuer** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm***.

8. Locate the **OIDC JWKs URL** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs***.

9. In the Satellite web UI, navigate to **Administer** > **Authentication Sources**, click the vertical ellipsis on the **External** card, and select **Edit**.

10. Click the **Locations** tab and add locations that can use the Red Hat Single Sign-On authentication source.

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

12. Click **Submit**.

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

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

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

Prerequisite

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

Procedure

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

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

2. Set the login delegation logout URL:

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

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

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

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

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

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

NOTE

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

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

6. Set the value for the Open IDC issuer:

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

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

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

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

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

9. Set the location and organization:

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

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

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

Procedure

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

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

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

Procedure

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

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

The command prompts you to enter a success code.

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

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

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

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

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

Procedure

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

2. In the **Name** field, enter a name for the user group. The name should not be the same as in the Active Directory.

3. Do not add users and user groups to the right-hand columns. Click the **Roles** tab.

4. Select the **Administer** checkbox.

5. Click the **External Groups** tab.

6. Click **Add external user group**.

7. In the **Name** field, enter the name of the Active Directory group.

8. From the list, select **EXTERNAL**.

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

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

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

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

- A working installation of Red Hat Single Sign-On server that uses HTTPS instead of HTTP.

- A Red Hat Single Sign-On account with admin privileges.

- A realm for Satellite user accounts created in Red Hat Single Sign-On.

- If the certificates or the CA are self-signed, ensure that they are added to the end-user certificate trust store.

- Users imported or added to Red Hat Single Sign-On.
  If you have an existing user database configured such as LDAP or Kerberos, you can import users from it by configuring user federation. For more information, see User Storage Federation in the *Red Hat Single Sign-On Server Administration Guide* .

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

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

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

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

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

2. Users authenticate to Satellite using the Satellite CLI.

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

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

Procedure

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

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

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

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

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

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

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

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

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

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

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

3. Restart the **httpd** service:

```
# systemctl restart httpd
```

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

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

Procedure

1. In the Red Hat Single Sign-On web UI, navigate to **Clients** and click the Satellite client.

2. Configure access type:

   - If you want your users to authenticate to Satellite using the Satellite web UI, from the **Access Type** list, select **confidential**.

   - If you want your users to authenticate to Satellite using the CLI, from the **Access Type** list, select **public**.

3. In the **Valid redirect URI** fields, add a valid redirect URI.

   - If you want your users to authenticate to Satellite using the Satellite web UI, in the blank field below the existing URI, enter a URI in the form **https://satellite.example.com/users/extlogin**. Note that you must add the string **/users/extlogin** after the Satellite FQDN.
   After completing this step, the Satellite client for logging in using the Satellite web UI must have the following **Valid Redirect URIs**:

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

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

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

4. Click **Save**.

5. Click the **Mappers** tab and click **Create** to add an audience mapper.

6. In the **Name** field, enter a name for the audience mapper.

7. From the **Mapper Type** list, select **Audience**.

8. From the **Included Client Audience** list, select the Satellite client.

9. Click **Save**.

10. Click **Create** to add a group mapper so that you can specify authorization in Satellite based on group membership.

11. In the **Name** field, enter a name for the group mapper.

12. From the **Mapper Type** list, select **Group Membership**.

13. In the **Token Claim Name** field, enter **groups**.

14. Set the **Full group path** setting to OFF.

15. Click **Save**.

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

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

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

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

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

Prerequisite

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

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Settings**, and click the **Authentication** tab.

2. Locate the **Authorize login delegation** row, and in the **Value** column, set the value to **Yes**.

3. Locate the **Authorize login delegation auth source user autocreate**row, and in the **Value** column, set the value to **External**.

4. Locate the **Login delegation logout URL** row, and in the **Value** column, set the value to **https://satellite.example.com/users/extlogout**.

5. Locate the **OIDC Algorithm** row, and in the **Value** column, set the algorithm for encoding on Red Hat Single Sign-On to **RS256**.

6. Locate the **OIDC Audience** row, and in the **Value** column, set the value to the client ID for Red Hat Single Sign-On.

7. Locate the **OIDC Issuer** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm***.

8. Locate the **OIDC JWKs URL** row, and in the **Value** column, set the value to ***https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs***.

9. In the Satellite web UI, navigate to **Administer** > **Authentication Sources**, click the vertical ellipsis on the **External** card, and select **Edit**.

10. Click the **Locations** tab and add locations that can use the Red Hat Single Sign-On authentication source.

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

12. Click **Submit**.

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

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

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

Prerequisite

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

Procedure

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

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

2. Set the login delegation logout URL:

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

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

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

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

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

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

   > **NOTE**
   >
   > If you register several Red Hat Single Sign-On clients to Satellite, ensure that you append all audiences in the array. For example:
   >
   > ```
   > # hammer settings set --name oidc_audience \
   > --value "['satellite.example.com-foreman-openidc', 'satellite.example.com-hammer-openidc']"
   > ```

6. Set the value for the Open IDC issuer:

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

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

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

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

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

9. Set the location and organization:

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

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

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

Procedure

1. In the Red Hat Single Sign-On web UI, navigate to the Satellite realm.

2. Navigate to **Authentication**, and click the **OTP Policy** tab.

3. Ensure that the **Supported Applications** field includes FreeOTP or Google Authenticator.

4. Configure the OTP settings to suit your requirements.

5. Optional: If you want to use TOTP authentication as a default authentication method for all users, click the **Flows** tab, and to the right of the **OTP Form** setting, select **REQUIRED**.

6. Click the **Required Actions** tab.

7. To the right of the **Configure OTP** row, select the **Default Action** checkbox.

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

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

Procedure

1. Log in to Satellite, Satellite redirects you to the Red Hat Single Sign-On login screen.

2. Enter your username and password, and click **Log In**.

3. The first attempt to log in, Red Hat Single Sign-On requests you to configure your client by scanning the barcode and entering the pin displayed.

4. After you configure your client and enter a valid PIN, Red Hat Single Sign-On redirects you to Satellite and logs you in.

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

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

Procedure

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

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

   The command prompts you to enter a success code.

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

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

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

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

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

Procedure

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

14.10. DISABLING RED HAT SINGLE SIGN-ON AUTHENTICATION

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

Procedure

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

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

CHAPTER 15. 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.

15.1. 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 15.1. 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 15.2. 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 15.3. 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.

15.1.1. 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**.

15.2. 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.

15.3. MONITORING SATELLITE SERVER

From the **About** page in Satellite web UI, you can find an overview of the following:

- System Status, including Capsules, Available Providers, Compute Resources, and Plug-ins
- Support Information
- System Information
- Backend System Status
- Installed Packages

To navigate to the **About** page:

- In the upper right corner of Satellite web UI, click **Administer** > **About**.

NOTE

After Pulp failure, the status of Pulp might show **OK** instead of **Error** for up to 10 minutes due to synchronization delay.

15.4. MONITORING CAPSULE SERVER

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

15.4.1. 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.

15.4.2. 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 15.4.3, “Monitoring Puppet”.

15.4.3. 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 16. 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.

16.1. 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.

16.2. 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
```

16.3. 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**.

16.4. 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*.

16.5. 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

Event name	Description	Payload
Actions Remote Execution Run Host Job Succeeded	A generic remote execution job succeeded for a host. This event is emitted for all Remote Execution jobs, when complete.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Errata Install Succeeded	Install errata using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Group Install Succeeded	Install package group using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Install Succeeded	Install package using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Group Remove	Remove package group using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Remove Succeeded	Remove package using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Service Restart Succeeded	Restart Services using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Group Update Succeeded	Update package group using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Update Succeeded	Update package using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Foreman OpenSCAP Run Scans Succeeded	Run OpenSCAP scan.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Host Succeeded	Runs an Ansible playbook containing all the roles defined for a host.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Capsule Upgrade Succeeded	Upgrade Capsules on given Capsule server hosts.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Configure Cloud Connector Succeeded	Configure Cloud Connector on given hosts.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Insights Plan Succeeded	Runs a given maintenance plan from Red Hat Access Insights given an ID.	Actions::RemoteExecution::RunHostJob

Event name	Description	Payload
Actions Remote Execution Run Host Job Ansible Run Playbook Succeeded	Run an Ansible playbook against given hosts.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Enable Web Console Succeeded	Run an Ansible playbook to enable the web console on given hosts.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Puppet Run Host Succeeded	Perform a single Puppet run.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Module Stream Action Succeeded	Perform a module stream action using the Katello interface.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Pre-upgrade Succeeded	Upgradeability check for RHEL 7 host.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Remediation Plan Succeeded	Run Remediation plan with Leapp.	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Upgrade Succeeded	Run Leapp upgrade job for RHEL 7 host.	Actions::RemoteExecution::RunHostJob
Build Entered	A host entered the build mode.	Custom event: **@payload[:id]** (host id), **@payload[:hostname]** (host name).
Build Exited	A host build mode was canceled, either it was successfully provisioned or the user canceled the build manually.	Custom event: **@payload[:id]** (host id), **@payload[:hostname]** (host name).
Content View Created/Updated/Destroyed	Common database operations on a Content View.	Katello::ContentView
Domain Created/Updated/Destroyed	Common database operations on a domain.	Domain
Host Created/Updated/Destroyed	Common database operations on a host.	Host
Hostgroup Created/Updated/Destroyed	Common database operations on a hostgroup.	Hostgroup

Event name	Description	Payload
Model Created/Updated/Destroyed	Common database operations on a model.	Model
Status Changed	Global host status of a host changed.	Custom event: **@payload[:id]** (host id), **@payload[:hostname]**, **@payload[:global_status]** (hash)
Subnet Created/Updated/Destroyed	Common database operations on a subnet.	Subnet
Template Render Performed	A report template was rendered.	Template
User Created/Updated/Destroyed	Common database operations on a user.	User

16.6. SHELLHOOKS

With webhooks, you can only map one Satellite event to one API call. For advanced integrations, where a single shell script can contain multiple commands, you can install a Capsule shellhooks plugin that exposes executables using a REST HTTP API.

You can then configure a webhook to reach out to a Capsule API to run a predefined shellhook. A shellhook is an executable script that can be written in any language as long as it can be executed. The shellhook can for example contain commands or edit files.

You must place your executable scripts in **/var/lib/foreman-proxy/shellhooks** with only alphanumeric characters and underscores in their name.

You can pass input to shellhook script through the webhook payload. This input is redirected to standard input of the shellhook script. You can pass arguments to shellhook script using HTTP headers in format **X-Shellhook-Arg-1** to **X-Shellhook-Arg-99**. For more information on passing arguments to shellhook script, see:

- Section 16.8, “Passing Arguments to Shellhook Script Using Webhooks”
- Section 16.9, “Passing Arguments to Shellhook Script Using Curl”

The HTTP method must be POST. An example URL would be: **https://capsule.example.com:9090/shellhook/My_Script**.

NOTE

Unlike the **shellhooks** directory, the URL must contain **/shellhook/** in singular to be valid.

You must enable **Capsule Authorization** for each webhook connected to a shellhook to enable it to authorize a call.

Standard output and standard error output are redirected to the Capsule logs as messages with debug or warning levels respectively.

The shellhook HTTPS calls do not return a value.

For an example on creating a shellhook script, see Section 16.10, “Creating a Shellhook to Print Arguments”.

16.7. INSTALLING THE SHELLHOOKS PLUGIN

Optionally, you can install and enable the shellhooks plugin on each Capsule used for shellhooks, using the following command:

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

16.8. PASSING ARGUMENTS TO SHELLHOOK SCRIPT USING WEBHOOKS

Use this procedure to pass arguments to a shellhook script using webhooks.

Procedure

- When creating a webhook, on the **Additional** tab, create HTTP headers in the following format:

  ```
  {
    "X-Shellhook-Arg-1": "VALUE",
    "X-Shellhook-Arg-2": "VALUE"
  }
  ```

  Ensure that the headers have a valid JSON or ERB format. Only pass safe fields like database ID, name, or labels that do not include new lines or quote characters.

  For more information, see Section 16.4, “Creating a Webhook” .

Example

```
{
  "X-Shellhook-Arg-1": "<%= @object.content_view_version_id %>",
  "X-Shellhook-Arg-2": "<%= @object.content_view_name %>"
}
```

16.9. PASSING ARGUMENTS TO SHELLHOOK SCRIPT USING CURL

Use this procedure to pass arguments to a shellhook script using curl.

Procedure

- When executing a shellhook script using **curl**, create HTTP headers in the following format:

  ```
  "X-Shellhook-Arg-1: VALUE"
  "X-Shellhook-Arg-2: VALUE"
  ```

Example

```
# curl -sX POST -H 'Content-Type: text/plain' \
-H "X-Shellhook-Arg-1: Version 1.0" \
-H "X-Shellhook-Arg-2: My Content View" \
--data "" https://capsule.example.com:9090/shellhook/My_Script
```

16.10. CREATING A SHELLHOOK TO PRINT ARGUMENTS

Create a simple shellhook script that prints "Hello World!" when you run a remote execution job.

Prerequisite

- You have the **webhooks** and **shellhooks** plug-ins installed. For more information, see:
  - Section 16.2, “Installing Webhooks”
  - Section 16.7, “Installing the Shellhooks Plugin”

Procedure

1. Modify the **/var/lib/foreman-proxy/shellhooks/print_args** script to print arguments to standard error output so you can see them in the Capsule logs:

   ```
   #!/bin/sh
   #
   # Prints all arguments to stderr
   #
   echo "$@" >&2
   ```

2. In the Satellite web UI, navigate to **Administer** > **Webhooks**.
3. Click **Create new**.
4. From the **Subscribe to** list, select **Actions Remote Execution Run Host Job Succeeded**
5. Enter a **Name** for your webhook.
6. In the **Target URL** field, enter the URL of your Capsule Server followed by **:9090/shellhook/print_args**:

   ```
   https://capsule.example.com:9090/shellhook/print_args
   ```

   Note that **shellhook** in the URL is singular, unlike the **shellhooks** directory.
7. From the **Template** list, select **Empty Payload**.
8. On the **Credentials** tab, check **Capsule Authorization**.
9. On the **Additional** tab, enter the following text in the **Optional HTTP headers** field:

   ```
   {
      "X-Shellhook-Arg-1": "Hello",
      "X-Shellhook-Arg-2": "World!"
   ```

```
}
```

10. Click **Submit**. You now have successfully created a shellhook that prints "Hello World!" to Capsule logs every time you a remote execution job succeeds.

Verification

1. Run a remote execution job on any host. You can use **time** as a command. For more information, see Executing a Remote Job in *Managing Hosts*.

2. Verify that the shellhook script was triggered and printed "Hello World!" to Capsule Server logs:

   ```
   # tail /var/log/foreman-proxy/proxy.log
   ```

   You should find the following lines at the end of the log:

   ```
   [I] Started POST /shellhook/print_args
   [I] Finished POST /shellhook/print_args with 200 (0.33 ms)
   [I] [3520] Started task /var/lib/foreman-proxy/shellhooks/print_args\ Hello\ World\!
   [W] [3520] Hello World!
   ```

CHAPTER 17. SEARCHING AND BOOKMARKING

Satellite features powerful search functionality on most pages of the Satellite web UI. It enables you to search all kinds of resources that Satellite manages. Searches accept both free text and syntax-based queries, which can be built using extensive input prediction. Search queries can be saved as bookmarks for future reuse.

17.1. BUILDING SEARCH QUERIES

As you start typing a search query, a list of valid options to complete the current part of the query appears. You can either select an option from the list and keep building the query using the prediction, or continue typing. To learn how free text is interpreted by the search engine, see Section 17.2, “Using Free Text Search”.

17.1.1. Query Syntax

```
parameter operator value
```

Available fields, resources to search, and the way the query is interpreted all depend on context, that is, the page where you perform the search. For example, the field "hostgroup" on the Hosts page is equivalent to the field "name" on the Host Groups page. The field type also determines available operators and accepted values.

For a list of all operators, see Operators. For descriptions of value formats, see Values.

17.1.2. Query Operators

All operators that can be used between *parameter* and *value* are listed in the following table. Other symbols and special characters that might appear in a prediction-built query, such as colons, do not have special meaning and are treated as free text.

Table 17.1. Comparison Operators Accepted by Search

Operator	Short Name	Description	Example
=	EQUALS	Accepts numerical, temporal, or text values. For text, exact case sensitive matches are returned.	**hostgroup = RHEL7**
!=	NOT EQUALS		
~	LIKE	Accepts text or temporal values. Returns case insensitive matches. Accepts the following wildcards: _ for a single character, % or * for any number of characters including zero. If no wildcard is specified, the string is treated as if surrounded by wildcards: %rhel7%	**hostgroup ~ rhel%**
!~	NOT LIKE		

Operator	Short Name	Description	Example
>	GREATER THAN	Accepts numerical or temporal values. For temporal values, the operator > is interpreted as "later than", and < as "earlier than". Both operators can be combined with EQUALS: >= <=	**registered_at > 10-January-2017** The search will return hosts that have been registered after the given date, that is, between 10th January 2017 and now. **registered_at <= Yesterday** The search will return hosts that have been registered yesterday or earlier.
<	LESS THAN		
^	IN	Compares an expression against a list of values, as in SQL. Returns matches that contain or not contain the values, respectively.	**release_version !^ 7**
!^	NOT IN		
HAS or set?		Returns values that are present or not present, respectively.	**has hostgroup** or **set? hostgroup** On the Puppet Classes page, the search will return classes that are assigned to at least one host group. **not has hostgroup** or **null? hostgroup** On the Dashboard with an overview of hosts, the search will return all hosts that have no assigned host group.
NOT HAS or null?			

Simple queries that follow the described syntax can be combined into more complex ones using logical operators AND, OR, and NOT. Alternative notations of the operators are also accepted:

Table 17.2. Logical Operators Accepted by Search

Operator	Alternative Notations			Example
and	&	&&	<whitespace>	**class = motd AND environment ~ production**
or	|	||		**errata_status = errata_needed || errata_status = security_needed**
not	–	!		**hostgroup ~ rhel7 not status.failed**

17.1.3. Query Values

Text Values

Text containing whitespaces must be enclosed in quotes. A whitespace is otherwise interpreted as the AND operator.
Examples:

hostgroup = "Web servers"

The search will return hosts with assigned host group named "Web servers".

hostgroup = Web servers

The search will return hosts in the host group Web with any field matching %servers%.

Temporal Values

Many date and time formats are accepted, including the following:

- "10 January 2017"
- "10 Jan 2017"
- 10-January-2017
- 10/January/2017
- "January 10, 2017"
- Today, Yesterday, and the like.

WARNING

Avoid ambiguous date formats, such as 02/10/2017 or 10-02-2017.

17.2. USING FREE TEXT SEARCH

When you enter free text, it will be searched for across multiple fields. For example, if you type "64", the search will return all hosts that have that number in their name, IP address, MAC address, and architecture.

NOTE

Multi-word queries must be enclosed in quotes, otherwise the whitespace is interpreted as the AND operator.

Because of searching across all fields, free text search results are not very accurate and searching can be slow, especially on a large number of hosts. For this reason, we recommend that you avoid free text and use more specific, syntax-based queries whenever possible.

17.3. MANAGING BOOKMARKS

You can save search queries as bookmarks for reuse. You can also delete or modify a bookmark.

Bookmarks appear only on the page on which they were created. On some pages, there are default bookmarks available for the common searches, for example, all **active** or **disabled** hosts.

17.3.1. Creating Bookmarks

This section details how to save a search query as a bookmark. You must save the search query on the relevant page to create a bookmark for that page, for example, saving a host related search query on the Hosts page.

Procedure

1. In the Satellite web UI, navigate to the page where you want to create a bookmark.
2. In the **Search** field, enter the search query you want to save.
3. Select the arrow to the right of the **Search** button and then select **Bookmark this search**.
4. In the **Name** field, enter a name for the new bookmark.
5. In the **Search query** field, ensure your search query is correct.
6. Ensure the **Public** checkbox is set correctly:
   - Select the **Public** checkbox to set the bookmark as public and visible to all users.
   - Clear the **Public** checkbox to set the bookmark as private and only visible to the user who created it.
7. Click **Submit**.

To confirm the creation, either select the arrow to the right of the **Search** button to display the list of bookmarks, or navigate to **Administer** > **Bookmarks** and then check the **Bookmarks** list for the name of the bookmark.

17.3.2. Deleting Bookmarks

You can delete bookmarks on the Bookmarks page.

Procedure

1. In the Satellite web UI, navigate to **Administer** > **Bookmarks**.
2. On the Bookmarks page, click **Delete** for the Bookmark you want to delete.
3. When the confirmation window opens, click **OK** to confirm the deletion.

To confirm the deletion, check the **Bookmarks** list for the name of the bookmark.

APPENDIX A. ADMINISTRATION SETTINGS

This section contains information about settings that you can edit in the Satellite web UI by navigating to **Administer** > **Settings**.

A.1. GENERAL SETTINGS

Setting	Default Value	Description
Administrator email address		The default administrator email address
Satellite URL		URL where your Satellite instance is reachable. See also **Provisioning > Unattended URL**.
Entries per page	20	Number of records shown per page in Satellite
Fix DB cache	No	Satellite maintains a cache of permissions and roles. When set to **Yes**, Satellite recreates this cache on the next restart.
DB pending seed	No	Should the **foreman-rake db:seed** be executed on the next run of the installer modules?
Capsule request timeout	60	Open and read timeout for HTTP requests from Satellite to Capsule (in seconds).
Login page footer text		Text to be shown in the login-page footer.
Show host power status	Yes	Show power status on the host index page. This feature calls to compute resource providers which may lead to decreased performance on the host listing page.
HTTP(S) proxy		Set a proxy for outgoing HTTP(S) connections from the Satellite product. System-wide proxies must be configured at the operating system level.
HTTP(S) proxy except hosts	[]	Set hostnames to which requests are not to be proxied. Requests to the local host are excluded by default.

Setting	Default Value	Description
Show Experimental Labs	No	Whether or not to show a menu to access experimental lab features (requires reload of page).
Append domain names to the host	Yes	If set to **Yes**, Satellite appends domain names when new hosts are provisioned.
Out of sync interval	30	Managed hosts report periodically, and if the time between reports exceeds this duration in minutes, hosts are considered out of sync. You can override this on your hosts by adding the **outofsync_interval** parameter, per host, at **Hosts > All hosts > $host > Edit > Parameters > Add Parameter**.
Satellite UUID		Satellite instance ID. Uniquely identifies a Satellite instance.
Default language		The UI for new users uses this language.
Default timezone		The timezone to use for new users.
Instance title		The instance title is shown on the top navigation bar (requires a page reload).
Saved audits interval		Duration in days to preserve audit data. Leave empty to disable the audits cleanup.
New host details UI	Yes	Satellite loads the new UI for host details.

A.2. SATELLITE TASK SETTINGS

Setting	Default Value	Description
Sync task timeout	120	Number of seconds to wait for a synchronous task to finish before an exception is raised.
Enable dynflow console	Yes	Enable the dynflow console (**/foreman_tasks/dynflow**) for debugging.
Require auth for dynflow console	Yes	The user must be authenticated as having administrative rights before accessing the dynflow console.

Setting	Default Value	Description
Capsule action retry count	4	Number of attempts permitted to start a task on the Capsule before failing.
Capsule action retry interval	15	Time in seconds between retries.
Allow Capsule batch tasks	Yes	Enable batch triggering of tasks on the Capsule.
Capsule tasks batch size	100	Number of tasks included in one request to the Capsule if **foreman_tasks_proxy_batch_trigger** is enabled.
Tasks troubleshooting URL		URL pointing to the task troubleshooting documentation. It should contain a **%{label}** placeholder that is replaced with a normalized task label (restricted to only alphanumeric characters)). A **%{version}** placeholder is also available.
Polling intervals multiplier	1	Polling multiplier used to multiply the default polling intervals. You can use this to prevent polling too frequently for long running tasks.

A.3. TEMPLATE SYNC SETTINGS

Setting	Default Value	Description
Associate	New	Associate templates with OS, organization and location.
Branch		Default branch in Git repo.
Commit message	Templates export made by a Satellite user	Custom commit message for exported templates.
Dirname	/	The directory within the Git repo containing the templates.
Filter		Import or export of names matching this regex. Case-insensitive. Snippets are not filtered.
Force import	No	If set to **Yes**, locked templates are overwritten during an import.

Setting	Default Value	Description
Lock templates	Keep, do not lock new	How to handle lock for imported templates.
Metadata export mode	Refresh	Default metadata export mode. Possible options: **refresh** re-renders metadata. **keep** keeps existing metadata. **remove** exports the template without metadata.
Negate	No	Negate the filter for import or export.
Prefix		A string added as a prefix to imported templates.
Repo		Target path from where to import or export templates. Different protocols can be used, for example: **/tmp/dir** **git://example.com** **https://example.com** **ssh://example.com** When exporting to **/tmp**, note that production deployments may be configured to use **private tmp**.
Verbosity	No	Choose verbosity for Rake task importing templates.

A.4. DISCOVERED SETTINGS

Setting	Default Value	Description
Discovery location		Indicates the default location to place discovered hosts in.
Discovery organization		Indicates the default organization to which discovered hosts are added.

Setting	Default Value	Description
Interface fact	discovery_bootif	Fact name to use for primary interface detection.
Create bond interfaces	No	Automatically create a bond interface if another interface is detected on the same VLAN using LLDP.
Clean all facts	No	Clean all reported facts (except discovery facts) during provisioning.
Hostname facts	discovery_bootif	List of facts to use for the hostname (comma separated, first wins).
Auto provisioning	No	Use the provisioning rules to automatically provision newly discovered hosts.
Reboot	Yes	Automatically reboot or kexec discovered hosts during provisioning.
Hostname prefix	mac	The default prefix to use for the hostname. Must start with a letter.
Fact columns		Extra facter columns to show in host lists (comma separated).
Highlighted facts		Regex to organize facts for highlights section – e.g. **^(abc|cde)$**.
Storage facts		Regex to organize facts for the storage section.
Software facts		Regex to organize facts for the software section.
Hardware facts		Regex to organize facts for the hardware section.
Network facts		Regex to organize facts for the network section.
IPMI facts		Regex to organize facts for the *Intelligent Platform Management Interface* (IPMI) section.

Setting	Default Value	Description
Lock PXE	No	Automatically generate a *Preboot Execution Environment* (PXE) configuration to pin a newly discovered host to discovery.
Locked PXELinux template name	pxelinux_discovery	PXELinux template to be used when pinning a host to discovery.
Locked PXEGrub template name	pxegrub_discovery	PXEGrub template to be used when pinning a host to discovery.
Locked PXEGrub2 template name	pxegrub2_discovery	PXEGrub2 template to be used when pinning a host to discovery.
Force DNS	Yes	Force the creation of DNS entries when provisioning a discovered host.
Error on existing NIC	No	Do not permit to discover an existing managed host matching the MAC of a provisioning *Network Interface Card* (NIC) (errors out early).
Type of name generator	Fact + prefix	Discovery hostname naming pattern.
Prefer IPv6	No	Prefer IPv6 to IPv4 when calling discovered nodes.

A.5. BOOT DISK SETTINGS

Setting	Default Value	Description
iPXE directory	**/usr/share/ipxe**	Path to directory containing iPXE images.
ISOLINUX directory	**/usr/share/syslinux**	Path to directory containing ISOLINUX images.
SYSLINUX directory	**/usr/share/syslinux**	Path to directory containing SYSLINUX images.
Grub2 directory	**/var/lib/tftpboot/grub2**	Path to directory containing **grubx64.efi** and **shimx64.efi**.
Host image template	Boot disk iPXE - host	iPXE template to use for host-specific boot disks.

Setting	Default Value	Description
Generic image template	Boot disk iPXE - generic host	iPXE template to use for generic host boot disks.
Generic Grub2 EFI image template	Boot disk Grub2 EFI - generic host	Grub2 template to use for generic *Extensible Firmware Interface* (EFI) host boot disks.
ISO generation command	genisoimage	Command to generate ISO image, use **genisoimage** or **mkisofs**.
Installation media caching	Yes	Installation media files are cached for full host images.
Allowed bootdisk types	[generic, host, full_host, subnet]	List of permitted bootdisk types. Leave blank to disable it.

A.6. RED HAT CLOUD SETTINGS

Setting	Default Value	Description
Automatic inventory upload	Yes	Enable automatic upload of your host inventory to the Red Hat cloud.
Synchronize recommendations Automatically	No	Enable automatic synchronization of Insights recommendations from the Red Hat cloud.
Obfuscate host names	No	Obfuscate hostnames sent to the Red Hat cloud.
Obfuscate host ipv4 addresses	No	Obfuscate IPv4 addresses sent to the Red Hat cloud.
ID of the RHC daemon	*****	RHC daemon id.

A.7. CONTENT SETTINGS

Setting	Default Value	Description
Default HTTP Proxy		Default HTTP Proxy for syncing content.
CDN SSL version		SSL version used to communicate with the CDN.

Setting	Default Value	Description
Default synced OS provisioning template	Kickstart default	Default provisioning template for operating systems created from synced content.
Default synced OS finish template	Kickstart default finish	Default finish template for new operating systems created from synced content.
Default synced OS user-data	Kickstart default user data	Default user data for new operating systems created from synced content.
Default synced OS PXELinux template	Kickstart default PXELinux	Default PXELinux template for new operating systems created from synced content.
Default synced OS PXEGrub template	Kickstart default PXEGrub	Default PXEGrub template for new operating systems created from synced content.
Default synced OS PXEGrub2 template	Kickstart default PXEGrub2	Default PXEGrub2 template for new operating systems created from synced content.
Default synced OS iPXE template	Kickstart default iPXE	Default iPXE template for new operating systems created from synced content.
Default synced OS partition table	Kickstart default	Default partitioning table for new operating systems created from synced content.
Default synced OS kexec template	Discovery Red Hat kexec	Default kexec template for new operating systems created from synced content.
Default synced OS Atomic template	Atomic Kickstart default	Default provisioning template for new atomic operating systems created from synced content.
Manifest refresh timeout	1200	Timeout when refreshing a manifest (in seconds).
Accept action timeout	20	Time in seconds to wait for a host to pick up a remote action.
Finish action timeout	3600	Time in seconds to wait for a host to finish a remote action.

Setting	Default Value	Description
Subscription connection enabled	Yes	Can communicate with the Red Hat Portal for subscriptions.
Installable errata from Content View	No	Calculate errata host status based only on errata in a host's Content View and Lifecycle Environment.
Restrict Composite Content View promotion	No	If this is enabled, a composite content view cannot be published or promoted, unless the component content view versions that it includes exist in the target environment.
Check services before actions	Yes	Check the status of backend services such as pulp and candlepin before performing actions?
Batch size to sync repositories in	100	How many repositories should be synced concurrently on a Capsule. A smaller number may lead to longer sync times. A larger number will increase dynflow load.
Sync Capsules after Content View promotion	Yes	Whether or not to auto sync Capsules after a Content View promotion.
Default Custom Repository download policy	**immediate**	Default download policy for custom repositories. Either **immediate** or **on_demand**.
Default Red Hat Repository download policy	**on_demand**	Default download policy for enabled Red Hat repositories. Either **immediate** or **on_demand**.
Default Capsule download policy	**on_demand**	Default download policy for Capsule syncs. Either **inherit**, **immediate**, or **on_demand**.
Pulp export destination filepath	**/var/lib/pulp/katello-export**	On-disk location for exported repositories.
Pulp 3 export destination filepath	**/var/lib/pulp/exports**	On-disk location for Pulp 3 exported repositories.
Pulp client key	**/etc/pki/katello/private/pulp-client.key**	Path for SSL key used for Pulp server authentication.

Setting	Default Value	Description
Pulp client cert	**/etc/pki/katello/certs/pulp-client.crt**	Path for SSL certificate used for Pulp server authentication.
Sync Connection Timeout	300	Total timeout in seconds for connections when syncing.
Use remote execution by default	No	If enabled, remote execution is used instead of katello-agent for remote actions.
Delete Host upon unregister	No	When unregistering a host using subscription-manager, also delete the host record. Managed resources linked to the host such as virtual machines and DNS records might also be deleted.
Subscription manager name registration fact		When registering a host using subscription-manager, force use the specified fact for the host name (in the form of **fact.fact**).
Subscription manager name registration fact strict matching	No	If this is enabled, and **register_hostname_fact** is set and provided, registration looks for a new host by name only using that fact, and skips all hostname matching.
Default Location subscribed hosts	Default Location	Default location where new subscribed hosts are stored after registration.
Expire soon days	120	The number of days remaining in a subscription before you are reminded about renewing it.
Content View Dependency Solving Default	No	The default dependency solving value for new content views.
Host Duplicate DMI UUIDs	[]	If hosts fail to register because of duplicate *Desktop Management Interface* (DMI) UUIDs, add their comma-separated values here. Subsequent registrations generate a unique DMI UUID for the affected hosts.
Host Profile Assume	Yes	Enable new host registrations to assume registered profiles with matching hostname as long as the registering DMI UUID is not used by another host.

Setting	Default Value	Description
Host Profile Can Change In Build	No	Enable host registrations to bypass **Host Profile Assume** as long as the host is in build mode.
Host Can Re-Register Only In Build	No	Enable hosts to re-register only when they are in build mode.
Host Tasks Workers Pool Size	5	Number of workers in the pool to handle the execution of host-related tasks. When set to 0, the default queue is used. Restart of the dynflowd/foreman-tasks service is required.
Applicability Batch Size	50	Number of host applicability calculations to process per task.
Autosearch	Yes	For pages that support it, automatically perform the search while typing in search input.
Autosearch delay	500	If Autosearch is enabled, delay in milliseconds before executing searches while typing.
Pulp bulk load size	2000	The number of items fetched from a single paged Pulp API call.
Upload profiles without Dynflow	Yes	Enable Katello to update host installed packages, enabled repositories, and module inventory directly instead of wrapped in Dynflow tasks (try turning off if Puma processes are using too much memory).
Orphaned Content Protection Time	1440	Time in minutes to consider orphan content as orphaned.
Prefer registered through Capsule for remote execution	No	Prefer using a proxy to which a host is registered when using remote execution.
Allow deleting repositories in published content views	Yes	Enable removal of repositories that the user has previously published in one or more Content View versions.

A.8. AUTHENTICATION SETTINGS

Setting	Default Value	Description
OAuth active	Yes	Satellite will use OAuth for API authorization.
OAuth consumer key	*****	OAuth consumer key.
OAuth consumer secret	*****	OAuth consumer secret.
OAuth map users	No	Satellite maps users by username in the request-header. If this is disabled, OAuth requests have administrator rights.
Failed login attempts limit	30	Satellite blocks user logins from an incoming IP address for 5 minutes after the specified number of failed login attempts. Set to 0 to disable brute force protection.
Restrict registered Capsules	Yes	Only known Capsules can access features that use Capsule authentication.
Require SSL for capsules	Yes	Client SSL certificates are used to identify Capsules (**:require_ssl** should also be enabled).
Trusted hosts	[]	List of hostnames, IPv4, IPv6 addresses or subnets to be trusted in addition to Capsules for access to fact/report importers and ENC output.
SSL certificate	**/etc/foreman/client_cert.pem**	SSL Certificate path that Satellite uses to communicate with its proxies.
SSL CA file	**/etc/foreman/proxy_ca.pem**	SSL CA file path that Satellite uses to communicate with its proxies.
SSL private key	**/etc/foreman/client_key.pem**	SSL Private Key path that Satellite uses to communicate with its proxies.
SSL client DN env	HTTP_SSL_CLIENT_S_DN	Environment variable containing the subject DN from a client SSL certificate.
SSL client verify env	HTTP_SSL_CLIENT_VERIFY	Environment variable containing the verification status of a client SSL certificate.
SSL client cert env	HTTP_SSL_CLIENT_CERT	Environment variable containing a client’s SSL certificate.

Setting	Default Value	Description
Server CA file		SSL CA file path used in templates to verify the connection to Satellite.
Websockets SSL key	**etc/pki/katello/private/katello-apache.key**	Private key file path that Satellite uses to encrypt websockets.
Websockets SSL certificate	**/etc/pki/katello/certs/katello-apache.crt**	Certificate path that Satellite uses to encrypt websockets.
Websockets encryption	Yes	VNC/SPICE websocket proxy console access encryption (**websockets_ssl_key/cert** setting required).
Login delegation logout URL		Redirect your users to this URL on logout. Enable **Authorize login delegation** also.
Authorize login delegation auth source user autocreate	External	Name of the external authentication source where unknown externally authenticated users (see **Authorize login delegation**) are created. Empty means no autocreation.
Authorize login delegation	No	Authorize login delegation with **REMOTE_USER HTTP** header.
Authorize login delegation API	No	Authorize login delegation with **REMOTE_USER HTTP** header for API calls too.
Idle timeout	60	Log out idle users after the specified number of minutes.
BCrypt password cost	9	Cost value of bcrypt password hash function for internal auth-sources (4–30). A higher value is safer but verification is slower, particularly for stateless API calls and UI logins. A password change is needed to affect existing passwords.
BMC credentials access	Yes	Permits access to BMC interface passwords through ENC YAML output and in templates.

Setting	Default Value	Description
OIDC JWKs URL		OpenID Connect *JSON Web Key Set* (JWKS) URL. Typically **https://keycloak.example.com/auth/realms/<realm name>/protocol/openid-connect/certs** when using Keycloak as an OpenID provider.
OIDC Audience	[]	Name of the OpenID Connect Audience that is being used for authentication. In the case of Keycloak this is the Client ID.
OIDC Issuer		The issuer claim identifies the principal that issued the *JSON Web tokens* (JWT), which exists at a **/.well-known/openid-configuration** in case of most of the OpenID providers.
OIDC Algorithm		The algorithm used to encode the JWT in the OpenID provider.

A.9. EMAIL SETTINGS

Setting	Default Value	Description
Email reply address		Email reply address for emails that Satellite is sending.
Email subject prefix		Prefix to add to all outgoing email.
Send welcome email	No	Send a welcome email including username and URL to new users.
Delivery method	Sendmail	Method used to deliver email.
SMTP enable StartTLS auto	Yes	SMTP automatically enables StartTLS.
SMTP OpenSSL verify mode	Default verification mode	When using TLS, you can set how OpenSSL checks the certificate.
SMTP address		SMTP address to connect to.
SMTP port	25	SMTP port to connect to.
SMTP HELO/EHLO domain		HELO/EHLO domain.

Setting	Default Value	Description
SMTP username		Username to use to authenticate, if required.
SMTP password	*****	Password to use to authenticate, if required.
SMTP authentication	none	Specify authentication type, if required.
Sendmail arguments	-i	Specify additional options to sendmail. Only used when the delivery method is set to sendmail.
Sendmail location	**/usr/sbin/sendmail**	The location of the sendmail executable. Only used when the delivery method is set to sendmail.

A.10. NOTIFICATIONS SETTINGS

Setting	Default Value	Description
RSS enable	Yes	Pull RSS notifications.
RSS URL	https://www.redhat.com/en/rss/blog/channel/red-hat-satellite	URL from which to fetch RSS notifications.

A.11. PROVISIONING SETTINGS

Setting	Default Value	Description
Host owner		Default owner on provisioned hosts, if empty Satellite uses the current user.
Root password	*****	Default encrypted root password on provisioned hosts.
Unattended URL		URL that hosts retrieve templates from during the build. When it starts with https, unattended, or userdata, controllers cannot be accessed using HTTP.

Setting	Default Value	Description
Safemode rendering	**Yes**	Enables safe mode rendering of provisioning templates. The default and recommended option **Yes** denies access to variables and any object that is not listed in Satellite. When set to **No**, any object may be accessed by a user with permission to use templating features, either by editing templates, parameters or smart variables. This permits users full remote code execution on Satellite Server, effectively disabling all authorization. This is not a safe option, especially in larger companies.
Access unattended without build	No	Enable access to unattended URLs without build mode being used.
Query local nameservers	No	Satellite queries the locally configured resolver instead of the SOA/NS authorities.
Installation token lifetime	360	Time in minutes that installation tokens should be valid for. Set to 0 to disable the token.
SSH timeout	120	Time in seconds before SSH provisioning times out.
Libvirt default console address	0.0.0.0	The IP address that should be used for the console listen address when provisioning new virtual machines using libvirt.
Update IP from built request	No	Satellite updates the host IP with the IP that made the build request.
Use short name for VMs	No	Satellite uses the short hostname instead of the FQDN for creating new virtual machines.
DNS timeout	[5, 10, 15, 20]	List of timeouts (in seconds) for DNS lookup attempts such as the **dns_lookup** macro and DNS record conflict validation.

Setting	Default Value	Description
Clean up failed deployment	Yes	Satellite deletes the virtual machine if the provisioning script ends with a non-zero exit code.
Type of name generator	**Random-based**	Specifies the method used to generate a hostname when creating a new host. The default **Random-based** option generates a unique random hostname which you can but do not have to use. This is useful for users who create many hosts and do not know how to name them. The **MAC-based** option is for bare-metal hosts only. If you delete a host and create it later on, it receives the same hostname based on the MAC address. This can be useful for users who recycle servers and want them to always get the same hostname. The **Off** option disables the name generator function and leaves the hostname field blank.
Default PXE global template entry		Default PXE menu item in a global template – **local**, **discovery** or custom, use blank for template default.
Default PXE local template entry		Default PXE menu item in local template – **local**, **local_chain_hd0**, or custom, use blank for template default.
iPXE intermediate script	iPXE intermediate script	Intermediate iPXE script for unattended installations.
Destroy associated VM on host delete	No	Destroy associated VM on host delete. When enabled, VMs linked to hosts are deleted on Compute Resource. When disabled, VMs are unlinked when the host is deleted, meaning they remain on Compute Resource and can be re-associated or imported back to Satellite again. This does not automatically power off the VM
Maximum structured facts	100	Maximum number of keys in structured subtree, statistics stored in **satellite::dropped_subtree_facts**.

Setting	Default Value	Description
Default Global registration template	Global Registration	Global Registration template.
Default 'Host initial configuration' template	Linux host_init_config default	Default 'Host initial configuration' template, automatically assigned when a new operating system is created.
Global default PXEGrub2 template	PXEGrub2 global default	Global default PXEGrub2 template. This template is deployed to all configured TFTP servers. It is not affected by upgrades.
Global default PXELinux template	PXELinux global default	Global default PXELinux template. This template is deployed to all configured TFTP servers. It is not affected by upgrades.
Global default PXEGrub template	PXEGrub global default	Global default PXEGrub template. This template is deployed to all configured TFTP servers. It is not affected by upgrades.
Global default iPXE template	iPXE global default	Global default iPXE template. This template is deployed to all configured TFTP servers. It is not affected by upgrades.
Local boot PXEGrub2 template	PXEGrub2 default local boot	Template that is selected as PXEGrub2 default for local boot.
Local boot PXELinux template	PXELinux default local boot	Template that is selected as PXELinux default for local boot.
Local boot PXEGrub template	PXEGrub default local boot	Template that is selected as PXEGrub default for local boot.
Local boot iPXE template	iPXE default local boot	Template that is selected as iPXE default for local boot.
Manage PuppetCA	Yes	Satellite automates certificate signing upon provision of a new host.
Use UUID for certificates	No	Satellite uses random UUIDs for certificate signing instead of hostnames.

Setting	Default Value	Description
Show unsupported provisioning templates	No	Show unsupported provisioning templates. When enabled, all the available templates are shown. When disabled, only Red Hat supported templates are shown.

A.12. FACTS SETTINGS

Setting	Default Value	Description
Create new host when facts are uploaded	Yes	Satellite creates the host when new facts are received.
Location fact	satellite_location	Hosts created after a Puppet run are placed in the location specified by this fact.
Organization fact	satellite_organization	Hosts created after a Puppet run are placed in the organization specified by this fact. The content of this fact should be the full label of the organization.
Default location	Default Location	Hosts created after a Puppet run that did not send a location fact are placed in this location.
Default organization	Default Organization	Hosts created after a Puppet run that did not send an organization fact are placed in this organization.
Update hostgroup from facts	Yes	Satellite updates a host’s hostgroup from its facts.
Ignore facts for operating system	No	Stop updating operating system from facts.
Ignore facts for domain	No	Stop updating domain values from facts.
Update subnets from facts	None	Satellite updates a host’s subnet from its facts.
Ignore interfaces facts for provisioning	No	Stop updating IP and MAC address values from facts (affects all interfaces).

Setting	Default Value	Description
Ignore interfaces with matching identifier	[**lo**, **en*v***, **usb***, **vnet***, **macvtap***, **;vdsmdummy;**, **veth***, **tap***, **qbr***, **qvb***, **qvo***, **qr-***, **qg-***, **vlinuxbr***, **vovsbr***, **br-int**]	Skip creating or updating host network interfaces objects with identifiers matching these values from incoming facts. You can use a * wildcard to match identifiers with indexes, e.g. **macvtap***. The ignored interface raw facts are still stored in the database, see the **Exclude pattern** setting for more details.
Exclude pattern for facts stored in satellite	[**lo**, **en*v***, **usb***, **vnet***, **macvtap***, **;vdsmdummy;**, **veth***, **tap***, **qbr***, **qvb***, **qvo***, **qr-***, **qg-***, **vlinuxbr***, **vovsbr***, **br-int**, **load_averages::***, **memory::swap::available***, **memory::swap::capacity**, **memory::swap::used***, **memory::system::available***, **memory::system::capacity**, **memory::system::used***, **memoryfree**, **memoryfree_mb**, **swapfree**, **swapfree_mb**, **uptime_hours**, **uptime_days**]	Exclude pattern for all types of imported facts (Puppet, Ansible, rhsm). Those facts are not stored in the satellite database. You can use a * wildcard to match names with indexes, e.g. **ignore*** filters out ignore, ignore123 as well as a::ignore or even a::ignore123::b.

A.13. CONFIGURATION MANAGEMENT SETTINGS

Setting	Default Value	Description
Create new host when report is uploaded	Yes	Satellite creates the host when a report is received.
Matchers inheritance	Yes	Satellite matchers are inherited by children when evaluating smart class parameters for hostgroups, organizations, and locations.
Default parameters lookup path	[**fqdn**, **hostgroup**, **os**, **domain**]	Satellite evaluates host smart class parameters in this order by default.
Interpolate ERB in parameters	Yes	Satellite parses ERB in parameters value in the ENC output.

Setting	Default Value	Description
Always show configuration status	No	All hosts show a configuration status even when a Puppet Capsule is not assigned.

A.14. REMOTE EXECUTION SETTINGS

Setting	Default Value	Description
Fallback to Any Capsule	No	Search the host for any proxy with Remote Execution. This is useful when the host has no subnet or the subnet does not have an execution proxy.
Enable Global Capsule	Yes	Search for Remote Execution proxy outside of the proxies assigned to the host. The search is limited to the host's organization and location.
SSH User	root	Default user to use for SSH. You can override per host by setting the **remote_execution_ssh_user** parameter.
Effective User	root	Default user to use for executing the script. If the user differs from the SSH user, su or sudo is used to switch the user.
Effective User Method	sudo	The command used to switch to the effective user. One of [**sudo**, **dzdo**, **su**]
Effective user password	*****	Effective user password. See **Effective User**.
Sync Job Templates	Yes	Whether to sync templates from disk when running **db:seed**.
SSH Port	22	Port to use for SSH communication. Default port 22. You can override per host by setting the **remote_execution_ssh_port** parameter.

Setting	Default Value	Description
Connect by IP	No	Whether the IP addresses on host interfaces are preferred over the FQDN. It is useful when the DNS is not resolving the FQDNs properly. You can override this per host by setting the **remote_execution_connect_by_ip** parameter. For dual-stacked hosts, consider the **remote_execution_connect_by_ip_prefer_ipv6** setting.
Prefer IPv6 over IPv4	No	When connecting using an IP address, are IPv6 addresses preferred? If no IPv6 address is set, it falls back to IPv4 automatically. You can override this per host by setting the **remote_execution_connect_by_ip_prefer_ipv6** parameter. By default and for compatibility, IPv4 is preferred over IPv6.
Default SSH password	*****	Default password to use for SSH. You can override per host by setting the **remote_execution_ssh_password** parameter.
Default SSH key passphrase	*****	Default key passphrase to use for SSH. You can override per host by setting the **remote_execution_ssh_key_passphrase** parameter.
Workers pool size	5	Number of workers in the pool to handle the execution of the remote execution jobs. Restart of the **dynflowd/satellite-tasks** service is required.
Cleanup working directories	Yes	Whether working directories are removed after task completion. You can override this per host by setting the **remote_execution_cleanup_working_dirs** parameter.
Cockpit URL		Where to find the Cockpit instance for the Web Console button. By default, no button is shown.
Form Job Template	Run Command - SSH Default	Choose a job template that is pre-selected in job invocation form.

Setting	Default Value	Description
Job Invocation Report Template	Jobs - Invocation report template	Select a report template used for generating a report for a particular remote execution job.

A.15. ANSIBLE SETTINGS

Setting	Default Value	Description
Private Key Path		Use this to supply a path to an SSH Private Key that Ansible uses instead of a password. Override with the **ansible_ssh_private_key_file** host parameter.
Connection type	ssh	Use this connection type by default when running Ansible playbooks. You can override this on hosts by adding the **ansible_connection** parameter.
WinRM cert Validation	validate	Enable or disable WinRM server certificate validation when running Ansible playbooks. You can override this on hosts by adding the **ansible_winrm_server_cert_validation** parameter.
Default verbosity level	Disabled	Satellite adds this level of verbosity for additional debugging output when running Ansible playbooks.
Post-provision timeout	360	Timeout (in seconds) to set when Satellite triggers an Ansible roles task playbook after a host is fully provisioned. Set this to the maximum time you expect a host to take until it is ready after a reboot.
Ansible report timeout	30	Timeout (in minutes) when hosts should have reported.
Ansible out of sync disabled	No	Disable host configuration status turning to out of sync for Ansible after a report does not arrive within the configured interval.
Default Ansible inventory report template	Ansible - Ansible Inventory	Satellite uses this template to schedule the report with Ansible inventory.

Setting	Default Value	Description
Ansible roles to ignore	[]	The roles to exclude when importing roles from Capsule. The expected input is comma separated values and you can use * wildcard metacharacters. For example: **foo***, ***b***, ***bar**.
Capsule tasks batch size for Ansible		Number of tasks which should be sent to the Capsule in one request if **satellite_tasks_proxy_batch_trigger** is enabled. If set, it overrides **satellite_tasks_proxy_batch_size** setting for Ansible jobs.