Red Hat Satellite 6.14

Administering Red Hat Satellite

Administer users and permissions, manage organizations and locations, back up and restore Satellite, maintain Satellite, and more

Red Hat Satellite 6.14 Administering Red Hat Satellite

Administer users and permissions, manage organizations and locations, back up and restore
Satellite, maintain Satellite, and more

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

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

Prerequisites

- Ensure you have registered a Red Hat account.

Procedure

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

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

3. Click **Create**.

CHAPTER 1. ACCESSING RED HAT SATELLITE

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

1.1. IMPORTING THE KATELLO ROOT CA CERTIFICATE

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

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

Prerequisite

- Your Red Hat Satellite is installed and configured.

Procedure

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

   ```
   # hostname -f
   ```

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

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

3. When you access Satellite for the first time, an untrusted connection warning displays in your web browser. Accept the self-signed certificate and add the Satellite URL as a security exception to override the settings. This procedure might differ depending on the browser being used. Ensure that the Satellite URL is valid before you accept the security exception.

4. Select **katello-server-ca.crt**.

5. Import the certificate into your browser as a certificate authority and trust it to identify websites.

CLI procedure

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

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

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

1.2. LOGGING IN TO SATELLITE

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

Prerequisite

- Ensure that the Katello root CA certificate is installed in your browser. For more information, see Section 1.1, "Importing the Katello Root CA Certificate" .

Procedure

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

   > https://*satellite.example.com*/

2. Enter the user name and password created during the configuration process. If a user was not created during the configuration process, the default user name is *admin*. If you have problems logging in, you can reset the password. For more information, see Section 1.8, "Resetting the Administrative User Password".

1.3. USING RED HAT IDENTITY MANAGEMENT CREDENTIALS TO LOG IN TO THE SATELLITE HAMMER CLI

This section describes how to log in to your Satellite Hammer CLI with your Red Hat Identity Management (IdM) login and password.

Prerequisites

- You have enrolled your Satellite Server into Red Hat Identity Management and configured it to use Red Hat Identity Management for authentication. More specifically, you have enabled access both to the Satellite web UI and the Satellite API. For more information, see Using Red Hat Identity Management in *Installing Satellite Server in a Connected Network Environment* .

- The host on which you run this procedure is configured to use Red Hat Identity Management credentials to log users in to your Satellite Hammer CLI. For more information, see Configuring the Hammer CLI to Use Red Hat Identity Management User Authentication in *Installing Satellite Server in a Connected Network Environment*.

- The host is an Red Hat Identity Management client.

- An Red Hat Identity Management server is running and reachable by the host.

Procedure

1. Obtain a Kerberos ticket-granting ticket (TGT) on behalf of a Satellite user:

   > $ kinit idm_user

> **WARNING**
>
> If, when you were setting Red Hat Identity Management to be the authentication provider, you enabled access to both the Satellite API and the Satellite web UI, an attacker can now obtain an API session after the user receives the Kerberos TGT. The attack is possible even if the user did not previously enter the Satellite login credentials anywhere, for example in the browser.

2. If automatic negotiate authentication is not enabled, use the TGT to authenticate to Hammer manually:

```
$ hammer auth login negotiate
```

3. Optional: Destroy all cached Kerberos tickets in the collection:

```
$ kdestroy -A
```

You are still logged in, even after destroying the Kerberos ticket.

Verification

- Use any **hammer** command to ensure that the system does not ask you to authenticate again:

```
$ hammer host list
```

> **NOTE**
>
> To log out of Hammer, enter: **hammer auth logout**.

1.4. USING RED HAT IDENTITY MANAGEMENT CREDENTIALS TO LOG IN TO THE SATELLITE WEB UI WITH A FIREFOX BROWSER

This section describes how to use the Firefox browser to log in to your Satellite web UI with your Red Hat Identity Management (IdM) login and password.

Prerequisites

- You have enrolled your Satellite Server into Red Hat Identity Management and configured the server to use Red Hat Identity Management for authentication. For more information, see Using Red Hat Identity Management in *Installing Satellite Server in a Connected Network Environment* .

- The host on which you are using a Firefox browser to log in to the Satellite web UI is an Red Hat Identity Management client.

- You have a valid Red Hat Identity Management login and password.

- Red Hat recommends using the latest stable Firefox browser.

- Your Firefox browser is configured for Single Sign-On (SSO). For more information, see Configuring Firefox to use Kerberos for single sign-on in *Configuring authentication and authorization in Red Hat Enterprise Linux.*

- An Red Hat Identity Management server is running and reachable by the host.

Procedure

1. Obtain the Kerberos ticket granting ticket (TGT) for yourself using your Red Hat Identity Management credentials:

   ```
   $ kinit idm_user
   Password for idm_user@EXAMPLE.COM:
   ```

2. In your browser address bar, enter the URL of your Satellite Server.
   You are logged in automatically.

> **NOTE**
>
> Alternatively, you can skip the first two steps and enter your login and password in the fields displayed on the Satellite web UI. This is also the only option if the host from which you are accessing the Satellite web UI is not an Red Hat Identity Management client.

1.5. USING RED HAT IDENTITY MANAGEMENT CREDENTIALS TO LOG IN TO THE SATELLITE WEB UI WITH A CHROME BROWSER

This section describes how to use a Chrome browser to log in to your Satellite web UI with your Red Hat Identity Management login and password.

Prerequisites

- You have enrolled your Satellite Server into Red Hat Identity Management and configured the server to use Red Hat Identity Management for authentication. For more information, see Using Red Hat Identity Management in *Installing Satellite Server in a Connected Network Environment* .

- The host on which you are using the Chrome browser to log in to the Satellite web UI is an Red Hat Identity Management client.

- You have a valid Red Hat Identity Management login and password.

- Red Hat recommends using the latest stable Chrome browser.

- An Red Hat Identity Management server is running and reachable by the host.

Procedure

1. Enable the Chrome browser to use Kerberos authentication:

   ```
   $ google-chrome --auth-server-whitelist=".example.com" --auth-negotiate-delegate-whitelist=".example.com"
   ```

> **NOTE**
>
> Instead of allowlisting the whole domain, you can also allowlist a specific Satellite Server.

2. Obtain the Kerberos ticket-granting ticket (TGT) for yourself using your Red Hat Identity Management credentials:

```
$ kinit idm_user
Password for idm_user@_EXAMPLE.COM:
```

3. In your browser address bar, enter the URL of your Satellite Server.
   You are logged in automatically.

> **NOTE**
>
> Alternatively, you can skip the first three steps and enter your login and password in the fields displayed on the Satellite web UI. This is also the only option if the host from which you are accessing the Satellite web UI is not an Red Hat Identity Management client.

1.6. 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 Lifecycle 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.

1.7. CHANGING THE PASSWORD

These steps show how to change your password.

Procedure

1. In the Satellite web UI, 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 **Submit** to save your new password.

1.8. RESETTING THE ADMINISTRATIVE USER PASSWORD

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

To Reset the Administrative User Password

1. Log in to the base operating system where Satellite Server is installed.

2. Enter the following command to reset the password:

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

3. Use this password to reset the password in the Satellite web UI.

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

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

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

To Set a New Administrative User Password

1. Log in to the base operating system where Satellite Server is installed.

2. To set the password, enter the following command:

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

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

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

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

1.9. SETTING A CUSTOM MESSAGE ON THE LOGIN PAGE

Procedure

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

2. Click the edit button next to **Login page footer text**, and enter the desired text to be displayed on the login page. For example, this text may be a warning message required by your company.

3. Click **Save**.

4. Log out of the Satellite web UI and verify that the custom text is now displayed on the login page below the Satellite version number.

CHAPTER 2. STARTING AND STOPPING RED HAT SATELLITE

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

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

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

To see the status of running services, execute:

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

To stop Satellite services, execute:

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

To start Satellite services, execute:

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

To restart Satellite services, execute:

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

CHAPTER 3. CLONING SATELLITE SERVER

You can clone your Satellite Server to create instances to test upgrades and migration of instances to a different machine or operating system. This is an optional step to provide more flexibility during the upgrade or migration.

You cannot use the Satellite clone tool on a Capsule Server. Instead, you must backup the existing Capsule Server, restore it on the target server, and then reconfigure Capsule Server.

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

Terminology

Ensure that you understand the following terms:

Source server

The origin of the clone.

Target server

The new server that you copy files to and clone the source server to.

3.1. CLONING PROCESS OVERVIEW

1. Back up the source server.

2. Clone the source server to the target server.

3. Power off the source server.

4. Update the network configuration on the target server to match the target server's IP address with its new host name.

5. If any content hosts use the deprecated Katello Agent, restart the **goferd** service on all those content hosts and their Capsules to refresh the connection.

6. Test the new target server.

3.2. PREREQUISITES

To clone Satellite Server, ensure that you have the following resources available:

- A minimal install of Red Hat Enterprise Linux 8 to become the target server. Do not install Red Hat Enterprise Linux 8 software groups or third-party applications. Ensure that your server complies with all the required specifications. For more information, see Preparing your Environment for Installation in *Installing Satellite Server in a Connected Network Environment* .

- A backup of your Satellite Server that you make using the **satellite-maintain backup** script. You can use a backup with or without Pulp data.

- A Satellite subscription for the target server.

Before you begin cloning, ensure the following conditions exist:

- The target server is on an isolated network. This avoids unwanted communication with Capsule Servers and hosts.

- The target server has at least the same storage capacity as the source server.

Customized configuration files

If you have any customized configurations on your source server that are not managed by the **satellite-installer** tool or Satellite backup process, you must manually back up these files.

3.3. PULP DATA CONSIDERATIONS

You can clone Satellite server without including Pulp data. However, for your cloned environment to work, you do require Pulp data. If the target server does not have Pulp data. it is not a fully working Satellite.

To transfer Pulp data to a target server, you have two options:

- Clone using backup with Pulp data

- Clone using backup without Pulp data and copy **/var/lib/pulp** manually from the source server.

If your **pulp_data.tar** file is greater than 500 GB, or if you use a slow storage system, such as NFS, and your **pulp_data.tar** file is greater than 100 GB, do not include **pulp_data.tar** in the backup because this can cause memory errors during extraction. Copy the **pulp_data.tar** file from the source server to the target server.

To back up without Pulp data

Follow the steps in the procedure in Section 3.4, "Cloning Satellite Server" and replace the steps that involve cloning with Pulp data with the following steps:

1. Perform a backup with PostgreSQL databases active excluding the Pulp data:

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

2. Stop and disable Satellite services:

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

3. Copy the Pulp data to the target server:

    ```
    # rsync --archive --partial --progress --compress \
    /var/lib/pulp/ target_server.example.com:/var/lib/pulp/
    ```

Proceed to Section 3.4.2, "Cloning to the Target Server" .

3.4. CLONING SATELLITE SERVER

Use the following procedures to clone Satellite Server. Note that because of the high volume of data that you must copy and transfer as part of these procedures, it can take a significant amount of time to complete.

3.4.1. Preparing the source server for cloning

On the source server, complete the following steps:

1. Verify the Pool ID of your Satellite subscription:

   ```
   # subscription-manager list --consumed \
   --matches 'Red Hat Satellite'|grep "Pool ID:"|awk '{print $3}'
   ```

   Note the *Pool ID* for later use.

2. Remove the Red Hat Satellite subscription:

   ```
   # subscription-manager remove --serial=$(subscription-manager list \
   --consumed \
   --matches 'Red Hat Satellite'|grep "Serial:"|awk '{print $2}')
   ```

3. Determine the size of the Pulp data:

   ```
   # du -sh /var/lib/pulp/
   ```

4. If you have less than 500 GB of Pulp data, perform a backup with PostgreSQL databases active including the Pulp data. If you have more than 500 GB of Pulp data, skip the following steps and complete the steps in Section 3.3, "Pulp Data Considerations" before you continue.

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

5. Stop and disable Satellite services:

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

Proceed to Section 3.4.2, "Cloning to the Target Server" .

3.4.2. Cloning to the Target Server

To clone your server, complete the following steps on your target server:

1. The **satellite-clone** tool defaults to using /**backup**/ as the backup folder. If you copy to a different folder, update the **backup_dir** variable in the /**etc/satellite-clone/satellite-clone-vars.yml** file.

2. Place the backup files from the source Satellite in the /**backup**/ folder on the target server. You can either mount the shared storage or copy the backup files to the /**backup**/ folder on the target server.

3. Power off the source server.

4. Enter the following commands to register to the Customer Portal, attach subscriptions, and enable only the required subscriptions:

```
# subscription-manager register your_customer_portal_credentials
# subscription-manager attach --pool=pool_ID
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=satellite-maintenance-6.14-for-rhel-8-x86_64-rpms
# dnf module enable satellite-maintenance:el8
```

5. Install the **satellite-clone** package:

   ```
   # dnf install satellite-clone
   ```

   After you install the **satellite-clone** tool, you can adjust any configuration to suit your own deployment in the **/etc/satellite-clone/satellite-clone-vars.yml** file.

6. Run the **satellite-clone** tool:

   ```
   # satellite-clone
   ```

7. Reconfigure DHCP, DNS, TFTP, and remote execution services. The cloning process disables these services on the target Satellite Server to avoid conflict with the source Satellite Server.

8. Reconfigure and enable DHCP, DNS, and TFTP in the Satellite web UI. For more information, see Configuring External Services on Satellite Server in *Installing Satellite Server in a Connected Network Environment*.

9. Log in to the Satellite web UI, with the username **admin** and the password **changeme**. Immediately update the admin password to secure credentials.

10. Ensure that the correct organization is selected.

11. In the Satellite web UI, navigate to **Content** > **Subscriptions**.

12. Click **Manage Manifest**.

13. Click **Refresh** and then click **Close** to return to the list of subscriptions.

14. Verify that the available subscriptions are correct.

15. Follow the instructions in the **/usr/share/satellite-clone/logs/reassociate_capsules.txt** file to restore the associations between Capsules and their lifecycle environments.

16. Update your network configuration, for example, DNS, to match the target server's IP address with its new host name. The **satellite-clone** tool changes the host name to the source server's host name. If you want to change the host name to something different, you can use the **satellite-change-hostname** tool. For more information, see Renaming a Satellite or Capsule Server in *Administering Red Hat Satellite*.

17. If the source server uses the **virt-who** daemon, install and configure it on the target server. Copy all the **virt-who** configuration files in the **/etc/virt-who.d/** directory from the source server to the same directory on the target server. For more information, see *Configuring Virtual Machine Subscriptions in Red Hat Satellite*. After you perform an upgrade using the following chapters, you can safely decommission the source server.

CHAPTER 4. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES

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

Note that you cannot use tuning profiles on Capsules.

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

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

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

1. The default tuning profile defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** file

2. The tuning profile that you want to apply to your deployment and is defined in the **/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** directory

3. Optional: If you have configured a **/etc/foreman-installer/custom-hiera.yaml** file, Satellite applies these configuration settings.

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

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

default

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

Number of CPU cores: 4

medium

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

Number of CPU cores: 8

large

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

Number of CPU cores: 16

extra-large

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

Number of CPU cores: 32

extra-extra-large

Number of managed hosts: 60000+
RAM: 256G

Number of CPU cores: 48+

Procedure

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

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

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

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

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

CHAPTER 5. 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 5.2, "Preparing a Host for External Databases" . Prepare a Red Hat Enterprise Linux 8 server to host the external databases.

2. Section 5.3, "Installing PostgreSQL". Prepare PostgreSQL with databases for Satellite, Pulp and Candlepin with dedicated users owning them.

3. Section 5.4, "Migrating to External Databases" . Edit the parameters of **satellite-installer** to point to the new databases, and run **satellite-installer**.

5.1. 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.

5.2. PREPARING A HOST FOR EXTERNAL DATABASES

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

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

Prerequisite

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

Procedure

1. Use the instructions in Attaching the Satellite Infrastructure Subscription to attach a Satellite subscription to your server.

2. Disable all repositories and enable only the following repositories:

   ```
   # subscription-manager repos --disable '*'
   # subscription-manager repos \
   --enable=satellite-6.14-for-rhel-8-x86_64-rpms \
   --enable=satellite-maintenance-6.14-for-rhel-8-x86_64-rpms \
   --enable=rhel-8-for-x86_64-baseos-rpms \
   --enable=rhel-8-for-x86_64-appstream-rpms
   ```

3. Enable the following modules:

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

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

5.3. INSTALLING POSTGRESQL

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

Procedure

1. To install PostgreSQL, enter the following command:

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

2. To initialize PostgreSQL, enter the following command:

   ```
   # postgresql-setup initdb
   ```

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

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

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

   - checkpoint_completion_target: 0.9

   - max_connections: 500

   - shared_buffers: 512MB

   - work_mem: 4MB

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

   ```
   listen_addresses = '*'
   ```

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

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

6. Add the following line to the file:

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

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

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

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

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

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

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

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

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

11. Connect to the Pulp database:

```
postgres=# \c pulpcore
You are now connected to database "pulpcore" as user "postgres".
```

12. Create the **hstore** extension:

```
pulpcore=# CREATE EXTENSION IF NOT EXISTS "hstore";
CREATE EXTENSION
```

13. Exit the **postgres** user:

```
# \q
```

14. 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"
```

5.4. MIGRATING TO EXTERNAL DATABASES

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

Prerequisite

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

Procedure

1. On Satellite Server, stop Satellite services:

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

2. Start the **PostgreSQL** services:

```
# systemctl start postgresql
```

3. Back up the internal databases:

```
# satellite-maintain backup online \
--preserve-directory \
--skip-pulp-content \
/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-database foreman \
--foreman-db-host postgres.example.com \
--foreman-db-manage false \
--foreman-db-password Foreman_Password \
--foreman-db-username foreman \
--foreman-proxy-content-pulpcore-manage-postgresql false \
--foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \
--foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \
--foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password \
--foreman-proxy-content-pulpcore-postgresql-user pulp \
--katello-candlepin-db-host postgres.example.com \
--katello-candlepin-db-name candlepin \
--katello-candlepin-db-password Candlepin_Password \
--katello-candlepin-db-user candlepin \
--katello-candlepin-manage-db false
```

CHAPTER 6. 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.

6.1. INSTALLING THE SATELLITE ANSIBLE MODULES

Use this procedure to install the Satellite Ansible modules.

Procedure

- Install the package using the following command:

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

6.2. VIEWING THE SATELLITE ANSIBLE MODULES

You can view the installed Satellite Ansible modules by running:

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

Alternatively, you can also see the complete list of Satellite Ansible modules and other related information at 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 7. MANAGING ORGANIZATIONS

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

Single Organization

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

Multiple Organizations

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

External Organizations

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

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

New Users

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

7.1. CREATING AN ORGANIZATION

Use this procedure to create an organization. To use the CLI instead of the Satellite web UI, see the CLI procedure.

Procedure

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

2. Click **New Organization**.

3. In the **Name** field, enter a name for the organization.

4. In the **Label** field, enter a unique identifier for the organization. This is used for creating and mapping certain assets, such as directories for content storage. Use letters, numbers, underscores, and dashes, but no spaces.

5. Optional: If you do not wish to enable Simple Content Access (SCA), uncheck the **Simple Content Access** checkbox. For more information on SCA, see Simple Content Access .

> **NOTE**
>
> Red Hat does not recommend disabling SCA as entitlement mode is deprecated.

6. Optional: In the **Description** field, enter a description for the organization.

7. Click **Submit**.

8. If you have hosts with no organization assigned, select the hosts that you want to add to the organization, then click **Proceed to Edit**

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

10. Click **Submit**.

CLI procedure

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

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

   > **NOTE**
   >
   > Organizations created this way have Simple Content Access (SCA) enabled by default. If you wish to disable SCA, add the **--simple-content-access false** parameter to the command. Red Hat does not advise you to disable SCA because entitlement mode (not using SCA) is deprecated.

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

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

7.2. SETTING THE ORGANIZATION CONTEXT

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

Procedure

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

CLI procedure

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

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

This command outputs subscriptions allocated for the *My_Organization*.

7.3. CREATING AN ORGANIZATION DEBUG CERTIFICATE

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

Procedure

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

2. Select an organization that you want to generate a debug certificate for.

3. Click **Generate and Download**.

4. Save the certificate file in a secure location.

Debug Certificates for Provisioning Templates

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

7.4. BROWSING REPOSITORY CONTENT USING AN ORGANIZATION DEBUG CERTIFICATE

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

Prerequisite

- You created and downloaded an organization certificate. For more information, see Section 7.3, "Creating an Organization Debug Certificate".

Procedure

1. Split the private and public keys from the certificate into two files.

   a. Open the X.509 certificate, for example, for the default organization:

   ```
   $ vi 'Default Organization-key-cert.pem'
   ```

   b. Copy the contents of the file from **-----BEGIN RSA PRIVATE KEY-----** to **-----END RSA PRIVATE KEY-----**, into a **key.pem** file.

   c. Copy the contents of the file from **-----BEGIN CERTIFICATE-----** to **-----END CERTIFICATE-----**, into a **cert.pem** file.

2. To use a browser, you must first convert the X.509 certificate to a format your browser supports and then import the certificate.

For Firefox Users

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

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

2. In the Firefox browser, navigate to **Edit** > **Preferences** > **Advanced Tab**.

3. Select **View Certificates** and click the **Your Certificates** tab.

4. Click **Import** and select the **.pfx** file to load.

5. Enter the following URL in the address bar to browse the accessible paths for all the repositories and check their contents:

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

For CURL Users

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

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

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

7.5. DELETING AN ORGANIZATION

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

Procedure

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

2. From the list to the right of the name of the organization you want to delete, select **Delete**.

3. Click **OK** to delete the organization.

CLI procedure

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

   ```
   # hammer organization list
   ```

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

2. Enter the following command to delete an organization:

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

CHAPTER 8. MANAGING LOCATIONS

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

- Locations are based on physical or geographical settings.

- Locations have a hierarchical structure.

8.1. CREATING A LOCATION

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

Procedure

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

2. Click **New Location**.

3. Optional: From the **Parent** list, select a parent location. This creates a location hierarchy.

4. In the **Name** field, enter a name for the location.

5. Optional: In the **Description** field, enter a description for the location.

6. Click **Submit**.

7. If you have hosts with no location assigned, add any hosts that you want to assign to the new location, then click **Proceed to Edit**.

8. Assign any infrastructure resources that you want to add to the location. This includes networking resources, installation media, kickstart templates, and other parameters. You can return to this page at any time by navigating to **Administer** > **Locations** and then selecting a location to edit.

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

CLI procedure

- Enter the following command to create a location:

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

8.2. CREATING MULTIPLE LOCATIONS

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

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

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

8.3. SETTING THE LOCATION CONTEXT

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

Procedure

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

CLI procedure

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

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

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

8.4. DELETING A LOCATION

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

Procedure

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

2. Select **Delete** from the list to the right of the name of the location you want to delete.

3. Click **OK** to delete the location.

CLI procedure

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

   ```
   # hammer location list
   ```

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

2. Enter the following command to delete the location:

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

CHAPTER 9. 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 9.1, "Managing Users" for more information on working with users.

You can manage permissions of several users at once by organizing them into user groups. User groups themselves can be further grouped to create a hierarchy of permissions. For more information on creating user groups, see Section 9.4, "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 9.5, "Creating and Managing Roles".

9.1. MANAGING USERS

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

9.1.1. Creating a User

Use this procedure to create a user. To use the CLI instead of the Satellite web UI, see the CLI procedure.

Procedure

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

2. Click **Create User**.

3. In the **Login** field, enter a username for the user.

4. In the **Firstname** and **Lastname** fields, enter the real first name and last name of the user.

5. In the **Mail** field, enter the user's email address.

6. In the **Description** field, add a description of the new user.

7. Select a specific language for the user from the **Language** list.

8. Select a timezone for the user from the **Timezone** list.
   By default, Satellite Server uses the language and timezone settings of the user's browser.

9. Set a password for the user:

   a. From the **Authorized by** list, select the source by which the user is authenticated.

      - **INTERNAL**: to enable the user to be managed inside Satellite Server.

      - **EXTERNAL**: to configure external authentication as described in Configuring External Authentication in *Installing Satellite Server in a Connected Network Environment* .

b. Enter an initial password for the user in the **Password** field and the **Verify** field.

10. Click **Submit** to create the user.

CLI procedure

- To create a user, enter the following command:

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

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

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

9.1.2. 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
  ```

9.1.3. 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.

9.1.4. Creating an API-Only User

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

Prerequisite

- You have created a user and assigned roles to them. Note that this user must be authorized internally. For more information, see] and xref:Assigning_Roles_to_a_User_admin[.

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 9.3.1, "Creating a Personal Access Token".

9.2. MANAGING SSH KEYS

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

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

9.2.1. 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
```

9.3. 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.

9.3.1. 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":"6.14.0","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.

9.3.2. 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"}
   }
   ```

9.4. CREATING AND MANAGING USER GROUPS

9.4.1. User Groups

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

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

9.4.2. 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
  ```

9.4.3. Removing a User Group

Use the following procedure to remove a user group from Satellite.

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. Click **Confirm** to delete the user group.

9.5. CREATING AND MANAGING ROLES

Satellite provides a set of predefined roles with permissions sufficient for standard tasks, as listed in Section 9.6, "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.

9.5.1. 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 9.5.3, "Adding Permissions to a Role".

9.5.2. 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.

9.5.3. 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 9.7, "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.

9.5.4. 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 9.5.5, "Creating a Complete Permission Table" .

9.5.5. 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.

9.5.6. Removing a Role

Use the following procedure to remove a role from Satellite.

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. Click **Confirm** to delete the role.

9.6. PREDEFINED ROLES AVAILABLE IN SATELLITE

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

For a complete set of predefined roles and the permissions they grant, log in to Satellite web UI as the privileged user and navigate to **Administer** > **Roles**. For more information, see Section 9.5.4, "Viewing Permissions of a Role".

Predefined role	Permissions the role provides	Additional information
Auditor	View the Audit log.	

Predefined role	Permissions the role provides	Additional information
Default role	View tasks and jobs invocations.	Satellite automatically assigns this role to every user in the system.
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.
Site manager	View permissions for various items. Permissions to manage hosts in the infrastructure.	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. Give this role only to trusted users.
Viewer	View the configuration of every element of the Satellite structure, logs, reports, and statistics.	

9.7. GRANULAR PERMISSION FILTERING

As mentioned in Section 9.5.3, "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.

9.7.1. Creating a Granular Permission Filter

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

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

Procedure

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

> *field_name operator value*

- *field_name* marks the field to be queried. The range of available field names depends on the resource type. For example, the *Partition Table* resource type offers *family*, *layout*, and *name* as query parameters.

- *operator* specifies the type of comparison between *field_name* and *value*. See Section 9.7.3, "Supported Operators for Granular Search" for an overview of applicable operators.

- *value* is the value used for filtering. This can be for example a name of an organization. Two types of wildcard characters are supported: underscore (_) provides single character replacement, while percent sign (%) replaces zero or more characters.

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

CLI procedure

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

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

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

9.7.2. Examples of Using Granular Permission Filters

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

9.7.2.1. Applying Permissions for the Host Resource Type

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

> hostgroup = host-editors

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

> name ^ (XXXX, Yyyy, zzzz)

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

> **Dev**

You can limit user permissions to a certain organization or location with the use of the granular permission filter in the **Search** field. However, some resource types provide a GUI alternative, an **Override** checkbox that provides the **Locations** and **Organizations** tabs. On these tabs, you can select from the list of available organizations and locations. For more information, see Section 9.7.2.2, "Creating an Organization Specific Manager Role".

9.7.2.2. Creating an Organization Specific Manager Role

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

Procedure

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

2. Clone the existing **Organization admin** role. Select **Clone** from the drop-down list next to the **Filters** button. You are then prompted to insert a name for the cloned role, for example *org-1 admin*.

3. Click the desired locations and organizations to associate them with the role.

4. Click **Submit** to create the role.

5. Click *org-1 admin*, and click **Filters** to view all associated filters. The default filters work for most use cases. However, you can optionally click **Edit** to change the properties for each filter. For some filters, you can enable the **Override** option if you want the role to be able to access resources in additional locations and organizations. For example, by selecting the **Domain** resource type, the **Override** option, and then additional locations and organizations using the **Locations** and **Organizations** tabs, you allow this role to access domains in the additional locations and organizations that is not associated with this role. You can also click **New filter** to associate new filters with this role.

9.7.3. Supported Operators for Granular Search

Table 9.1. Logical Operators

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

Table 9.2. 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 10. CONFIGURING EMAIL NOTIFICATIONS

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

For an overview of available notification types, see Section 10.1, "Email Notification Types".

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

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

10.1. EMAIL NOTIFICATION TYPES

Satellite can create the following email notifications:

- **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.

- **Compliance 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.

- **Sync errata**: A notification sent only after synchronizing a repository. It contains a summary of new errata introduced by the synchronization.

For a complete list of email notification types, navigate to **Administer** > **Users** in the Satellite web UI, click the **Username** of the required user, and select the **Email Preferences** tab.

10.2. CONFIGURING EMAIL NOTIFICATION PREFERENCES

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

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

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

Procedure

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

2. Click the **Username** of the user you want to edit.

3. On the **User** tab, verify the value of the **Mail** field. Email notifications will be sent to the address in this field.

4. On the **Email Preferences** tab, select **Mail Enabled**.

5. Select the notifications you want the user to receive using the drop-down menus next to the notification types.
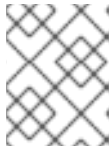
   > **NOTE**
   >
   > The **Audit Summary** notification can be filtered by entering the required query in the **Mail Query** text box.

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

10.3. TESTING EMAIL DELIVERY

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

Procedure

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

2. Click on the username.

3. On the **Email Preferences** tab, click **Test email**.
   A test email message is sent immediately to the user's email address.

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

   a. Verify the user's email address.

   b. Verify Satellite Server's email configuration.

   c. Examine firewall and mail server logs.

10.4. TESTING EMAIL NOTIFICATIONS

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

Procedure

- To trigger the notifications, execute the following command:

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

Replace *My_Frequency* with one of the following:

- daily

- weekly

- monthly

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

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

10.5. 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 11. BACKING UP SATELLITE SERVER AND CAPSULE SERVER

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

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

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

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

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

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

Conventional Backup Methods

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

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

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

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

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

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

11.1. ESTIMATING THE SIZE OF A BACKUP

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

A full backup requires space to store the following data:

- Uncompressed Satellite database and configuration files

- Compressed Satellite database and configuration files

- An extra 20% of the total estimated space to ensure a reliable backup

Procedure

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

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

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

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

   Table 11.1. Backup Data Compression Ratio

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

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

3. To calculate the amount of available space you require to store a backup, calculate the sum of the estimated values of compressed and uncompressed backup data, and add an extra 20% to ensure a reliable backup.

This example requires 201 GB plus 120 GB for the uncompressed and compressed backup data, 321 GB in total. With 64 GB of extra space, 385 GB must be allocated for the backup location.

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

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

There are three main methods of backing up Satellite Server:

- Offline backup

- Online backup

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

Offline backups

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

Online backups

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

Snapshots backups

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

Directory creation

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

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

- **satellite-backup** on Satellite

- **foreman-proxy-backup** on Capsule

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

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

Remote databases

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

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

Backing up to a remote NFS share

To enable Satellite to save the backup to an NFS share, ensure that the **root** user of your Satellite Server or Capsule Server can write to the NFS share. NFS export options such as **root_squash** and **all_squash** are known to prevent this.

For more information, see Red Hat Enterprise Linux *Configuring and using network files services* and Red Hat Enterprise Linux *Securing network services*.

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see Section 11.1, "Estimating the Size of a Backup".

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

Procedure

- On Satellite Server, enter the following command:

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

- On Capsule Server, enter the following command:

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

11.3. PERFORMING A BACKUP WITHOUT PULP CONTENT

You can perform an offline backup that excludes the contents of the Pulp directory. The backup without Pulp content is useful for debugging purposes and is only intended to provide access to configuration files without backing up the Pulp database. 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 11.1, "Estimating the Size of a Backup" .

Procedure

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

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

11.4. PERFORMING AN INCREMENTAL BACKUP

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

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

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

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see Section 11.1, "Estimating the Size of a Backup" .

Procedure

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
```

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

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

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

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

11.6. PERFORMING AN ONLINE BACKUP

Perform an online backup only for debugging purposes.

Risks Associated with Online Backups

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

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

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

Prerequisites

- Ensure that your backup location has sufficient available disk space to store the backup. For more information, see Section 11.1, "Estimating the Size of a Backup" .

Procedure

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

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

11.7. PERFORMING A SNAPSHOT BACKUP

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

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

To view the usage statement, enter the following command:

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

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

Prerequisites

- The system uses LVM for the directories that you snapshot: **/var/lib/pulp/** and **/var/lib/pgsql**.

- The free disk space in the relevant volume group (VG) is three times the size of the snapshot. More precisely, the VG must have enough space unreserved by the member logical volumes (LVs) to accommodate new snapshots. In addition, one of the LVs must have enough free space for the backup directory.

- The target backup directory is on a different LV than the directories that you snapshot.

Procedure

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

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

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

11.8. WHITE-LISTING AND SKIPPING STEPS WHEN PERFORMING BACKUPS

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

Procedure

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

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

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

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

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

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

12.1. RESTORING FROM A FULL BACKUP

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

Prerequisites

- Ensure that you are restoring to the correct instance. The Red Hat Satellite instance must have the same host name, configuration, and be the same minor version (X.Y) as the original system.

- Ensure that you have an existing target directory. The target directory is read from the configuration files contained within the archive.

- Ensure that you have enough space to store this data on the base system of Satellite Server or Capsule Server as well as enough space after the restoration to contain all the data in the **/etc/** and **/var/** directories contained within the backup.
  To check the space used by a directory, enter the following command:

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

  To check for free space, enter the following command:

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

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

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

  ```
  # restorecon -Rv /
  ```

Procedure

1. Choose the appropriate method to install Satellite or Capsule:

   - To install Satellite Server from a connected network, follow the procedures in Installing Satellite Server in a Connected Network Environment.

   - To install Satellite Server from a disconnected network, follow the procedures in Installing Satellite Server in a Disconnected Network Environment.

   - To install a Capsule Server, follow the procedures in Installing Capsule Server.

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

3. Run the restoration script.

   > # satellite-maintain restore */var/backup_directory*

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

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

Additional Resources

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

12.2. RESTORING FROM INCREMENTAL BACKUPS

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

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

Procedure

1. Restore the last full backup using the instructions in Section 12.1, "Restoring from a Full Backup" .

2. Remove the full backup data from Satellite Server's local file system, for example, **/var/** or **/var/tmp/**.

3. Copy the incremental backup data to Satellite Server's local file system, for example, **/var/** or **/var/tmp/**.

4. Restore the incremental backups in the same sequence that they are made:

   > # satellite-maintain restore */var/backup_directory*/FIRST_INCREMENTAL
   > # satellite-maintain restore */var/backup_directory*/SECOND_INCREMENTAL

Additional Resources

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

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

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

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

confirm, in the Satellite web UI, that Capsule Server is connected to Satellite Server. Then use the procedure Section 12.3.1, "Synchronizing Content from Satellite Server to Capsule Servers" to synchronize from Satellite.

12.3.1. Synchronizing Content from Satellite Server to Capsule Servers

Use this procedure to synchronize content from your Satellite Server to your Capsule Server.

Prerequisite

- Ensure that you either select the relevant organization and location context of your Capsule Server or choose **Any Organization** and **Any Location**.

Procedure

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

2. Select your Capsule Server.

3. On the **Overview** tab, click **Synchronize**.

   - Select **Optimized Sync** to synchronize content from your Satellite Server to your Capsule Server that bypasses unnecessary steps to speed up performance.

   - Select **Complete Sync** to perform a complete sync from your Satellite Server to your Capsule Server that synchronizes content even if the metadata has not changed.

CHAPTER 13. RENAMING SATELLITE SERVER OR CAPSULE SERVER

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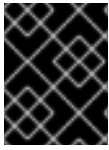
> **IMPORTANT**
>
> When changing the domain name of your Satellite Server or Capsule Server, update the hostname using **satellite-change-hostname** to avoid networking issues.

13.1. RENAMING SATELLITE SERVER

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

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

Prerequisites

- Back up your Satellite Server before changing its host name. If you fail to successfully rename it, restore it from the backup. For more information, see Chapter 11, *Backing Up Satellite Server and Capsule Server*.

- Run the **hostname** and **hostname -f** commands on Satellite Server. If both commands do not return the FQDN of Satellite Server, the **satellite-change-hostname** script will fail to complete. If the **hostname** command returns the shortname of Satellite Server instead of the FQDN, use **hostnamectl set-hostname** *My_Old_FQDN* to set the old FQDN correctly before using the **satellite-change-hostname** script.

- If Satellite Server has a custom SSL certificate installed, obtain a new certificate for the new FQDN of the host. For more information, see Configuring Satellite Server with a Custom SSL Certificate in *Installing Satellite Server in a Connected Network Environment* .

Procedure

1. On Satellite Server, run the **satellite-change-hostname** script, and provide the new host name. Choose one of the following methods:

   - If your Satellite Server is installed with the default self-signed SSL certificates, enter the following command:

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

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

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

2. If you have created a custom SSL certificate for the new Satellite Server host name, run the Satellite installation script to install the certificate. For more information about installing a custom SSL certificate, see Deploying a Custom SSL Certificate to Satellite Server in *Installing Satellite Server in a Connected Network Environment*.

3. Reregister all Satellite hosts. For more information, see Registering Hosts in *Managing Hosts*.

4. On all Capsule Servers, run the Satellite installation script to update references to the new host name:

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

5. On Satellite Server, list all Capsule Servers:

```
# hammer capsule list
```

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

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

7. If you use the *virt-who* agent, update the *virt-who* configuration files with the new host name. For more information, see Modifying a virt-who Configuration in *Configuring Virtual Machine Subscriptions in Red Hat Satellite*.

8. If you use external authentication, reconfigure Satellite Server for external authentication after you run the **satellite-change-hostname** script. For more information, see Configuring External Authentication in *Installing Satellite Server in a Connected Network Environment* .

13.2. RENAMING CAPSULE SERVER

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

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

Prerequisites

- Back up your Capsule Server before renaming. If you fail to successfully rename it, restore it from the backup. For more information, see Chapter 11, *Backing Up Satellite Server and Capsule Server*.

- Run the **hostname** and **hostname -f** commands on Capsule Server. If both commands do not return the FQDN of Capsule Server, the **satellite-change-hostname** script will fail to complete. If the **hostname** command returns the shortname of Capsule Server instead of the FQDN, use **hostnamectl set-hostname *My_Old_FQDN*** to set the old FQDN correctly before attempting to use the **satellite-change-hostname** script.

Procedure

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

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

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

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

   - If you are using a custom SSL certificate, create a new SSL certificate for Capsule Server. For more information, see Configuring Capsule Server with a Custom SSL Certificate in *Installing Capsule Server* .

2. On Satellite Server, copy the certificates archive file to Capsule Server. For example, to copy the archive file to the **root** user's home directory:

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

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

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

5. On all Capsule clients, enter the following commands to reinstall the bootstrap RPM, reregister clients, and refresh their subscriptions.
   You can use the remote execution feature to perform this step. For more information, see Configuring and Setting up Remote Jobs in *Managing Hosts*.

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

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

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

   # subscription-manager refresh
   ```

6. Update the Capsule host name in the Satellite web UI.

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

   b. Locate Capsule Server in the list, and click **Edit**.

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

   d. On your DNS server, add a record for the new hostname of your Capsule Server, and delete the record of the previous host name.

CHAPTER 14. MAINTAINING SATELLITE SERVER

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

14.1. DELETING AUDIT RECORDS MANUALLY

You can use the **foreman-rake audits:expire** command to remove audit records at any time.

Procedure

- Delete the audit records using the **foreman-rake audits:expire** command:

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

  This command deletes all audit records older than **Number_Of_Days**.

14.2. DELETING AUDIT RECORDS AUTOMATICALLY

You can automatically delete audit records using the **Saved audits interval** setting. This setting is empty by default, meaning Satellite does not automatically delete the audit records.

Procedure

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

2. On the **General** tab, find the **Saved audits interval** setting.

3. Set the value of the setting to the number of days after which you want Satellite to delete the audit records.

14.3. ANONYMIZING AUDIT RECORDS

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

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

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

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

14.4. DELETING REPORT RECORDS

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

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

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

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

14.5. 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
  ```

14.6. DELETING TASK RECORDS

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

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

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

14.7. DELETING A TASK BY ID

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

Procedure

1. Connect to your Satellite Server using SSH:

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

2. Optional: View the task:

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

3. Delete the task:

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

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

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

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

14.8. RECOVERING FROM A FULL DISK

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

Procedure

1. Let running Pulp tasks finish but do not trigger any new ones as they can fail due to the full disk.

2. Ensure that the LV with the **/var/lib/pulp** directory on it has sufficient free space. Here are some ways to achieve that:

   a. Remove orphaned content:

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

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

   b. Change the download policy from **Immediate** to **On Demand** for as many repositories as possible and remove already downloaded packages. See the Red Hat Knowledgebase solution How to change syncing policy for Repositories on Satellite from "Immediate" to "On-Demand" on the Red Hat Customer Portal for instructions.

   c. Grow the file system on the LV with the **/var/lib/pulp** directory on it. For more information, see Growing a logical volume and file system in *Red Hat Enterprise Linux 8 Configuring and managing logical volumes*.

> **NOTE**
>
> If you use an untypical file system (other than for example ext3, ext4, or xfs), you might need to unmount the file system so that it is not in use. In that case, complete the following steps:
>
> 1. Stop Satellite services:
>
>    ```
>    # satellite-maintain service stop
>    ```
>
> 2. Grow the file system on the LV.
>
> 3. Start Satellite services:
>
>    ```
>    # satellite-maintain service start
>    ```

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

14.9. 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.

You can manage packages using the **satellite-maintain packages** command as follows:

- To install packages on Satellite Server or Capsule Server:

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

- To check for available package updates on Satellite Server or Capsule Server:

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

- To update all packages on Satellite Server or Capsule Server:

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

- To update specific packages on Satellite Server or Capsule Server:

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

Updating packages individually can lead to package inconsistencies on Satellite Server or Capsule Server. For more information about updating packages on Satellite Server, see Updating Satellite Server to the Next Minor Version in *Updating Red Hat Satellite* .

14.10. 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
   ```

14.11. 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.

> **NOTE**
>
> Space reclamation requires an existing repository. For deleted repositories, wait for the next scheduled orphan cleanup or remove orphaned content manually:
>
> ```
> # foreman-rake katello:delete_orphaned_content
> ```

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 15. RENEWING THE CUSTOM SSL CERTIFICATE

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

15.1. RENEWING A CUSTOM SSL CERTIFICATE ON SATELLITE SERVER

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

Prerequisite

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

Procedure

- Before deploying a renewed custom certificate on your Satellite Server, validate the custom SSL input files. Note that for the **katello-certs-check** command to work correctly, Common Name (CN) in the certificate must match the FQDN of Satellite Server:

```
# katello-certs-check -t satellite \
-b /root/satellite_cert/ca_cert_bundle.pem \
-c /root/satellite_cert/satellite_cert.pem \
-k /root/satellite_cert/satellite_cert_key.pem
```

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

```
# satellite-installer --scenario satellite \
--certs-server-cert "/root/satellite_cert/satellite_cert.pem" \
--certs-server-key "/root/satellite_cert/satellite_key.pem" \
--certs-server-ca-cert "/root/satellite_cert/ca_cert_bundle.pem" \
--certs-update-server \
--certs-update-server-ca
```
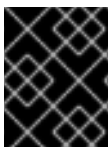
> **IMPORTANT**
>
> Do not delete the certificate files after you deploy the certificate. They are required when upgrading Satellite Server.
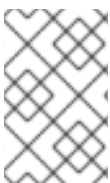
> **NOTE**
>
> If a new consumer package **katello-ca-consumer-latest.noarch.rpm** is generated due to a different Certificate Signing Authority, all the clients registered to Satellite Server must be updated.

Verification

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.

15.2. RENEWING A CUSTOM SSL CERTIFICATE ON CAPSULE SERVER

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

Prerequisite

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

Procedure

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

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

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

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

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 \
    --certs-tar-file "/root/My_Certificates/capsule.example.com-certs.tar" \
    --certs-update-server \
    --foreman-proxy-foreman-base-url "https://satellite.example.com" \
    --foreman-proxy-register-in-foreman "true"
    ```

IMPORTANT

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

NOTE

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

CHAPTER 16. LOGGING AND REPORTING PROBLEMS

The following sections describe the most commonly used log files and debugging tools available in Satellite.

16.1. INCREASING LOGGING LEVELS TO HELP WITH DEBUGGING

You can increase the logging level of Satellite components to troubleshoot Satellite. Increasing the logging level to **debug** provides the most detailed information. The default logging level is **info**.

16.1.1. Increasing the Logging Level for the Foreman Application

To increase logging for the Foreman Application:

1. Set the logging level to **debug**:

   ```
   # 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
   ```

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

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

16.1.2. Increasing the Logging Level for Hammer

You can find the log in ~/**.hammer/log/hammer.log**.

To increase logging for Hammer:

- In **/etc/hammer/cli_config.yml**, set the **:log_level:** option to **debug**:

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

16.1.3. Increasing the Logging Level for Capsule

You can find the log in **/var/log/foreman-proxy/proxy.log**.

To increase logging for Capsule:

1. Set the logging level to **debug**:

   ```
   # satellite-installer --foreman-proxy-log-level DEBUG
   ```

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

   ```
   # satellite-installer --reset-foreman-proxy-log-level
   ```

> **NOTE**
>
> Running satellite-installer will revert the setting back to default.

16.1.4. Increasing the Logging Level for Candlepin

You can find the logs in **/var/log/candlepin/candlepin.log** and **/var/log/candlepin/error.log**.

To increase logging for Candlepin:

1. In **/etc/candlepin/candlepin.conf**, set the logging level to **DEBUG**:

   log4j.logger.org.candlepin=DEBUG

2. Restart the **tomcat** service:

   # systemctl restart tomcat

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

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

16.1.5. Increasing the Logging Level for Qpid Dispatch Router

Qpid logs to **syslog**. You can view the log in **/var/log/messages** or with **journalctl**.

To increase logging for Qpid Dispatch Router:

1. In **/etc/qpid-dispatch/qdrouterd.conf**, set the logging level to **debug**:

   enable: debug+

2. Restart the Qpid Dispatch Router:

   # systemctl restart qdrouterd

> **NOTE**
>
> Running satellite-installer will revert the setting to default.

16.1.6. Increasing the Logging Level For Qpid Broker

Qpid logs to **syslog**. You can view the log in **/var/log/messages** or with **journalctl**.

To increase logging for Qpid Broker:

1. In **/etc/qpid/qpidd.conf**, set the logging level to **debug**:

   log-enable=debug+

2. Restart Qpid Broker:

```
# systemctl restart qpidd
```

> **NOTE**
>
> Running satellite-installer will revert the setting to default.

16.1.7. Increasing the Logging Level For Redis

You can find the log for Redis in **/var/log/redis/redis.log**.

To increase logging for Redis:

1. In **/etc/redis.conf**, set the logging level to **debug**:

   ```
   loglevel debug
   ```

2. Restart the Redis service:

   ```
   # systemctl restart redis
   ```

> **NOTE**
>
> Running satellite-installer will revert the setting to default.

16.1.8. Increasing the Logging Level For Satellite Installer

You can find the log files in **/var/log/foreman-installer**/.

To increase the logging level of the Satellite installer, add the **--verbose-log-level debug** option during the installation:

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

16.1.9. Increasing the Logging Level For Pulp

By default, Pulp logs to **syslog**. You can view the log in **/var/log/messages** or with **journalctl**.

To increase logging for Pulp:

1. In **/etc/pulp/settings.py**, set the logging level to **DEBUG**:

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

2. Restart the Pulp services:

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

16.1.10. Increasing the Logging Level For Puppet Agent

You can find the logs in **/var/log/puppetlabs/puppet/**.

To increase logging for Puppet agent:

1. Ensure Puppet is enabled in your Satellite. For more information, see Enabling Puppet Integration with Satellite in *Managing Configurations Using Puppet Integration in Red Hat Satellite*.

2. Set the logging level to **debug**:

   ```
   # satellite-installer --puppet-agent-additional-settings log_level:debug
   ```

16.1.11. Increasing the Logging Level For Puppet Server

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

To increase logging for Puppet server:

1. Ensure Puppet is enabled in your Satellite. For more information, see Enabling Puppet Integration with Satellite in *Managing Configurations Using Puppet Integration in Red Hat Satellite*.

2. Set the logging level to **debug**:

   ```
   # satellite-installer --puppet-server-additional-settings log_level:debug
   ```

3. Restart the Puppet server:

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

16.2. CONFIGURING LOGGING LEVELS WITH HAMMER CLI

You can use Hammer to log debugging information for various Satellite components. The default logging level is **INFO**.

- To list the components whose logging level you can change by using **hammer**:

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

- To set **DEBUG** level logging for all components:

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

- To set **PRODUCTION** level logging for all components:

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

- To set **DEBUG** level logging for a specific component:

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

- To list all available logging options:

```
# hammer admin logging --help
```

Additional resources

- Hammer CLI Guide

16.3. CONFIGURING LOGGING TO JOURNAL OR FILE-BASED LOGGING

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

Procedure for configuring logging with Journal

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

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

2. Optional: To inspect the log messages, use the **journalctl** utility. For example:

   - **journalctl --unit foreman** and **journalctl --unit foreman-proxy** show messages for the **foreman** and **foreman-proxy** units

   - **journalctl REQUEST=***request_ID* shows messages for a specified request

Procedure for configuring file-based logging

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

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

2. Optional: To inspect the log messages, view these files:

   - **/var/log/foreman/production.log**

   - **/var/log/foreman-proxy.log**

Additional resources

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

16.4. SELECTIVE DEBUGGING WITH INDIVIDUAL LOGGERS

You can enable individual loggers for selective debugging.

Procedure

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

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

2. Optional: Reset all loggers to their default values:

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

16.5. OVERVIEW OF INDIVIDUAL LOGGERS

The following list provides an overview of selected loggers. You can find the complete list of loggers with their default values in **/usr/share/foreman/config/application.rb** under **Foreman::Logging.add_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.

16.6. RETRIEVING THE STATUS OF SERVICES

Satellite uses a set of back-end services. When troubleshooting, you can check the status of Satellite services.

Procedure

- In the Satellite web UI, navigate to **Administer** > **About**.

  - On the **Smart Proxies** tab, view the status of all Capsules.

  - On the **Compute Resources** tab, view the status of attached compute resource providers.

  - In the **Backend System Status** table, view the status of all back-end services.

CLI procedure

- Get information from the database and Satellite services:

  ```
  # hammer ping
  ```

- Check the status of the services running in systemd:

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

  Run **satellite-maintain service --help** for more information.

- Perform a health check:

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

  Run **satellite-maintain health --help** for more information.

16.7. RESTARTING SERVICES

Satellite uses a set of back-end services. When troubleshooting, you can restart the services if needed.

Procedure

- Restart Satellite services:

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

16.8. UTILITIES FOR PROCESSING 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?* .
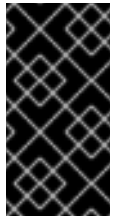
> **IMPORTANT**
>
> The **sosreport** command removes security information such as passwords, tokens, and keys while collecting information. However, the tar files can still contain sensitive information about the Satellite Server. Send the tar files directly to the intended recipient and not to a public target.

foreman-tail

The **foreman-tail** command displays Satellite logs in real time.
For more information, see the **foreman-tail(8)** man page.

16.9. LOG FILE DIRECTORIES

The following table provides an overview of selected log directories on Satellite Server:

Log file directory	Description of log file content
/var/log/candlepin/	Subscription management
/var/log/foreman-installer/	Installer
/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/	Puppet
/var/log/tomcat/	Candlepin web service logs

16.10. SYSTEM JOURNAL METADATA

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

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

ORG_LABEL

Organization label

REMOTE_IP

Remote IP address of a client

REQUEST

Request ID generated by the Action Dispatch module

SESSION

Random ID generated per session or a request for a sessionless request

TEMPLATE_NAME

Template name

TEMPLATE_DIGEST

Digest (SHA256) of rendered template contents

TEMPLATE_HOST_NAME

Host name for a rendered template if present

TEMPLATE_HOST_ID

Host database ID for a rendered template if present

USER_LOGIN

User login name

CHAPTER 17. MONITORING RESOURCES

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

17.1. USING THE RED HAT SATELLITE CONTENT DASHBOARD

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

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

Host Configuration Status

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

Table 17.1. Host Configuration States

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

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

Host Configuration Chart

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

Latest Events

A list of messages produced by hosts including administration information, product and subscription changes, and any errors.
Monitor this section for global notifications sent to all users and to detect any unusual activity or errors.

Run Distribution (last 30 minutes)

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

New Hosts

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

Task Status

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

Latest Warning/Error Tasks

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

Discovered Hosts

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

Latest Errata

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

Content Views

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

Sync Overview

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

Host Subscription Status

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

Table 17.2. Host Subscription States

Icon	State	Description
![red]	Invalid	Hosts that have products installed, but are not correctly subscribed. These hosts need attention immediately.
![orange]	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.
![green]	Valid	Hosts that have a valid entitlement and are using their full entitlements.

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

Subscription Status

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

Host Collections

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

Virt-who Configuration Status

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

Table 17.3. Virt-who Configuration States

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

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

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

Latest Compliance Reports

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

Compliance Reports Breakdown

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

Red Hat Insights Actions

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

Red Hat Insights Risk Summary

A table shows the distribution of the actions according to the risk levels. Risk level represents how critical the action is and how likely it is to cause an actual issue. The possible risk levels are: Low, Medium, High, and Critical.
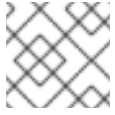
> **NOTE**
>
> It is not possible to change the date format displayed in the Satellite web UI.

17.1.1. Managing Tasks

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

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

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

To Adjust Timeout Settings:

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

2. Enter *%_timeout* in the search box and click **Search**. The search should return four settings, including a description.

3. In the **Value** column, click the icon next to a number to edit it.

4. Enter the desired value in seconds, and click **Save**.

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

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

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

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

To Disable Checking for Services:

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

2. Enter *check_services_before_actions* in the search box and click **Search**.

3. In the **Value** column, click the icon to edit the value.

4. From the drop-down menu, select **false**.

5. Click **Save**.

17.2. CONFIGURING RSS NOTIFICATIONS

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

By default, the Notifications area displays RSS feed events published in the Red Hat Satellite Blog .

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

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

To Configure RSS Feed Notifications:

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

2. In the RSS URL row, click the edit icon in the **Value** column and type the required URL.

3. In the RSS enable row, click the edit icon in the **Value** column to enable or disable this feature.

17.3. MONITORING SATELLITE SERVER

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

Procedure

1. In the Satellite web UI, navigate to **Monitor** > **Audits** to view the audit records.

2. To obtain a list of all the audit attributes, use the following command:

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

17.4. MONITORING CAPSULE SERVER

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

17.4.1. Viewing General Capsule Information

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

Is Capsule Server running?

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

What services are enabled on Capsule Server?

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

What organizations and locations is Capsule Server assigned to?

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

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

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

Which hosts are managed by Capsule Server?

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

How much storage space is available on Capsule Server?

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

17.4.2. Monitoring Services

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

17.4.3. Monitoring Puppet

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

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

- A list of Puppet environments at the **Environments** sub-tab.

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

- A certificate status overview and the number of autosign entries at the **General** sub-tab.

- A table of CA certificates associated with the Capsule at the **Certificates** sub-tab. Here you can inspect the certificate expiry data, or cancel the certificate by clicking **Revoke**.

- A list of autosign entries at the **Autosign entries** sub-tab. Here you can create an entry by clicking **New** or delete one by clicking **Delete**.

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

CHAPTER 18. USING WEBHOOKS

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

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

- **@event_name**: Name of an event.

- **@webhook_id**: Unique event ID.

- **@payload**: Payload data, different for each event type. To access individual fields, use **@payload[:key_name]** Ruby hash syntax.

- **@payload[:object]**: Database object for events triggered by database actions (create, update, delete). Not available for custom events.

- **@payload[:context]**: Additional information as hash like request and session UUID, remote IP address, user, organization and location.

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

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

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

You can use webhooks successfully without installing the shellhooks plugin.

For a list of available events, see Available webhook events.

18.1. MIGRATING TO WEBHOOKS

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

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

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

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

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

18.2. INSTALLING WEBHOOKS

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

Procedure

- Install webhooks using the following command:

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

- Optional: Install the webhooks CLI plug-in:

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

18.3. CREATING A WEBHOOK TEMPLATE

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

Procedure

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

2. Click **Clone an existing template** or **Create Template**.

3. Enter a name for the template.

4. Use the editor to make changes to the template payload.
   A webhook HTTP payload must be created using Satellite template syntax. The webhook template can use a special variable called **@object** that can represent the main object of the event. **@object** can be missing in case of certain events. You can determine what data are actually available with the **@payload** variable.

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

5. Optional: Enter the description and audit comment.

6. Assign organizations and locations.

7. Click **Submit**.

Examples

When creating a webhook template, you must follow the format of the target application for which the template is intended. For example, an application can expect a **"text"** field with the webhook message. Refer to the documentation of your target application to find more about how your webhook template format should look like.

Running remote execution jobs

This webhook template defines a message with the ID and result of a remote execution job. The webhook which uses this template can be subscribed to events such as **Actions Remote Execution Run Host Job Succeeded** or **Actions Remote Execution Run Host Job Failed**.

```
{
    "text": "job invocation <%= @object.job_invocation_id %> finished with result <%=
@object.task.result %>"
}
```

Creating users

This webhook template defines a message with the login and email of a created user. The webhook which uses this template should be subscribed to the **User Created** event.

```
{
    "text": "user with login <%= @object.login %> and email <%= @object.mail %> created"
}
```

18.4. CREATING A WEBHOOK

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

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

Procedure

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

2. Click **Create new**.

3. From the **Subscribe to** list, select an event.

4. Enter a **Name** for your webhook.

5. Enter a **Target URL**. Webhooks make HTTP requests to pre-configured URLs. The target URL can be a dynamic URL.

6. Click **Template** to select a template. Webhook templates are used to generate the body of the HTTP request to Satellite Server when a webhook is triggered.

7. Enter an HTTP method.

8. Optional: If you do not want activate the webhook when you create it, uncheck the **Enabled** flag.

9. Click the **Credentials** tab.

10. Optional: If HTTP authentication is required, enter **User** and **Password**.

11. Optional: Uncheck **Verify SSL** if you do not want to verify the server certificate against the system certificate store or Satellite CA.

12. On the **Additional** tab, enter the **HTTP Content Type**. For example, **application/json**, **application/xml** or **text/plain** on the payload you define. The application does not attempt to convert the content to match the specified content type.

13. Optional: Provide HTTP headers as JSON. ERB is also allowed.

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

18.5. AVAILABLE WEBHOOK EVENTS

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

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

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

Event name	Description	Payload
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
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).

Event name	Description	Payload
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
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

18.6. 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 18.8, "Passing Arguments to Shellhook Script Using Webhooks"](#)

- [Section 18.9, "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 18.10, "Creating a Shellhook to Print Arguments"](#).

18.7. 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
```

18.8. 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 18.4, "Creating a Webhook"](#).

Example

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

18.9. 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
```

18.10. 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 18.2, "Installing Webhooks"

  - Section 18.7, "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 19. 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.

19.1. 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 19.2, "Using Free Text Search".

19.1.1. 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.

19.1.2. 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 19.1. 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 19.2. 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**

19.1.3. 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.

19.2. 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.

19.3. 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.

19.3.1. 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.

19.3.2. 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.
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.
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.

Setting	Default Value	Description
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.
Capsule action retry count	4	Number of attempts permitted to start a task on the Capsule before failing.

Setting	Default Value	Description
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.
Lock templates	Keep, do not lock new	How to handle lock for imported templates.

Setting	Default Value	Description
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.
Interface fact	discovery_bootif	Fact name to use for primary interface detection.

Setting	Default Value	Description
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.
Lock PXE	No	Automatically generate a *Preboot Execution Environment* (PXE) configuration to pin a newly discovered host to discovery.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
Restrict Composite Content View promotion	No	If this is enabled, a composite content view cannot be published or promoted, unless the 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.
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.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
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.

Setting	Default Value	Description
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.
Server CA file		SSL CA file path used in templates to verify the connection to Satellite.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
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.
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.

Setting	Default Value	Description
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.
Time to pickup	86400	Time in seconds within which the host has to pick up a job. If the job is not picked up within this limit, the job will be cancelled. Applies only to pull-mqtt based jobs. Defaults to one day.

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.

Setting	Default Value	Description
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.
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.