



Red Hat Satellite 6.1 User Guide

A guide to using Satellite entitlement management software.
Edition 3

Red Hat Satellite Documentation
Team

Red Hat Satellite 6.1 User Guide

A guide to using Satellite entitlement management software.
Edition 3

Red Hat Satellite Documentation Team

Legal Notice

Copyright © 2015 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

The Red Hat Satellite 6 User Guide describes how to use Satellite, including subscriptions, content management, provisioning, and system control.

Table of Contents

Chapter 1. Red Hat Satellite Server 6 Basic Configuration Workflow	4
Chapter 2. Accessing Red Hat Satellite	6
2.1. Logging in to Red Hat Satellite	6
2.2. Changing the Password in Red Hat Satellite	7
Chapter 3. Starting and Stopping Red Hat Satellite	9
Chapter 4. Configuring Organizations, Locations and Life Cycle Environments	10
4.1. Organizations	10
4.2. Locations	12
4.3. Life Cycle Environments	14
4.4. Viewing Import History	15
Chapter 5. Using the Red Hat Satellite Content Dashboard	17
Chapter 6. Using Content Views	19
6.1. Creating a Content View	19
6.2. Adding Repositories to the Content View	20
6.3. Filtering Content	20
6.4. Publishing a Content View	22
Chapter 7. Searching for Content	23
7.1. Basic Content Search	23
7.2. Content Comparison across Environments	23
Chapter 8. Viewing and Applying Errata	24
8.1. Inspecting Available Errata	24
8.2. Applying Errata to Content Hosts	26
8.3. Subscribing to Errata Notifications	27
Chapter 9. Security Compliance Management with OpenSCAP	28
9.1. Installation	28
9.2. SCAP Concepts	29
9.3. Compliance Reports Overview	33
9.4. Uploading Additional SCAP Content	39
Chapter 10. Working with Containers	40
10.1. Managing Containers	40
10.2. Managing Repositories	45
10.3. Importing External Registries	46
10.4. Importing Images to Compute Resources	46
10.5. Using Container Tags	47
Chapter 11. Configuring Activation Keys	48
11.1. Creating an Activation Key	48
11.2. Removing an Activation Key	49
11.3. Editing Activation Keys	49
11.4. Automated Host Registration with Activation Keys	52
Chapter 12. Configuring GPG Keys	53
12.1. Creating a GPG Key	53
12.2. Removing a GPG Key	53
Chapter 13. Configuring the Provisioning Environment	54

13.1. Creating a Host Group	54
13.2. Parameters	54
13.3. Configuring Provisioning Settings	57
13.4. Storing and Maintaining Host Information	64
Chapter 14. Configuring Hosts	69
14.1. Creating a Host	69
14.2. Configuring a Host for Registration	69
14.3. Registration	70
14.4. Configuring an Additional Network Interface	75
14.5. Removing a Host	80
Chapter 15. Discovering Bare-metal Hosts on Satellite	81
15.1. Configuring the Satellite Discovery Plug-in	81
15.2. Configuring the Satellite Capsule Discovery Plug-in	84
15.3. Provisioning Discovered Hosts	85
15.4. Extending the Discovery Image	88
15.5. Troubleshooting Satellite Discovery	89
Chapter 16. Configuring Host Collections	91
16.1. Creating a Host Collection	91
16.2. Adding Hosts to a Host Collection	91
16.3. Adding Content to Host Collections	91
16.4. Removing Content from a Host Collection	92
16.5. Changing the Life Cycle Environment or Content View of a Host Collection	93
16.6. Removing a Host from a Host Collection	93
16.7. Removing a Host Collection	93
16.8. Cloning a Host Collection	94
16.9. Reviewing Host Collection Details	94
Chapter 17. Users and Roles	95
17.1. Creating and Managing Users	95
17.2. Creating User Groups	99
17.3. Creating and Managing Roles	100
17.4. Granular Permission Filtering	103
Chapter 18. Backup and Disaster Recovery	106
18.1. Backing up Red Hat Satellite Server	106
18.2. Restoring Red Hat Satellite Server from a Backup	106
Chapter 19. Maintaining a Red Hat Satellite Server	108
19.1. Logging and Reporting	108
19.2. Collecting Information from Log Files	108
19.3. Enabling Debug Logging	109
19.4. Using Log Files in Support Cases	111
Chapter 20. Configuring External Authentication	112
20.1. Using LDAP	112
20.2. Using Identity Management	115
20.3. Using Active Directory with Cross-Forest Trust	117
20.4. Using Active Directory Directly	117
20.5. External Authentication for Provisioned Hosts	120
Chapter 21. Red Hat Satellite User Interface Plug-ins	125
21.1. Accessing Customer Portal Services from Red Hat Satellite	125

Chapter 22. Command Line Reference	129
22.1. Configuring hammer	129
Appendix A. Glossary of Terms	131
Appendix B. Revision History	136

Chapter 1. Red Hat Satellite Server 6 Basic Configuration Workflow

Prerequisites

Before continuing with this workflow you must have successfully installed a Red Hat Satellite 6 Server and any required Capsule Servers. See [Red Hat Satellite Installation Guide](#) for further information.

Initial Configuration

These are the initial procedures to configure a basic Red Hat Satellite Server. You need the administrator privileges for the majority of the following actions:

1. Log in to the Satellite Server. See [Section 2.1, “Logging in to Red Hat Satellite”](#) for more information.

For information about changing the password, see [Section 2.2, “Changing the Password in Red Hat Satellite”](#).
2. Edit the Red Hat Satellite Integrated Capsule Server to select the desired organizations and locations. The name of the Satellite Integrated Capsule Server will be the same as the hostname of the server that Satellite 6 Server is installed on. See [Section 4.1.3, “Editing an Organization”](#) and [Section 4.2.2, “Editing a Location”](#) for more information.
3. Edit the desired location to select the resources to be associated with that location. See [Section 4.2, “Locations”](#) for more information.
4. Edit the default organization to select the resources to be associated with that organization. See [Section 4.1, “Organizations”](#) for more information.

Configuring a Red Hat Satellite Server

These are the initial procedures to configure a basic Red Hat Satellite Server:

1. Create a domain. See [Section 13.3.1, “Domains”](#) for more information.
2. Create a subnet. See [Section 13.3.2, “Subnets”](#) for more information.
3. Create the desired life cycle environments. See [Section 4.3, “Life Cycle Environments”](#) for more information.
4. Create any desired custom products. See the [Creating a Product](#) section in the [Installation Guide](#) for more information.
5. Choose the desired Red Hat Repositories.
 - a. Create a manifest from the Red Hat Customer Portal. See the [Setting up a Manifest](#) section in the [Installation Guide](#) for more information.
 - b. Upload the manifest in the Satellite Server web interface. This will propagate the subscription information into the Satellite Server. See the [Uploading a Subscription Manifest](#) section in the [Installation Guide](#) for more information.
 - c. Once the manifest has been uploaded, the Red Hat Repositories available from valid Red Hat Subscriptions are imported into the Satellite Server. Choose which repositories are relevant to your organization. See the [Uploading a Subscription Manifest](#) section in the [Installation Guide](#) for more information.

- d. Optional:
 - a. Red Hat source repositories update content based on security errata, bug fixes, and enhancements. To ensure that the Satellite Server is updated automatically, [Creating a Synchronization Plan](#) and [Creating a Synchronization Schedule](#) sections in the [Installation Guide](#) are recommended practices.
6. Manually synchronize content. See the [Synchronization Status](#) section in the [Installation Guide](#) for more information.
7. Create a content view with the desired repositories, puppet modules, and filters. Publish the content view then promote it to other life cycle environments as required. See [Chapter 6, Using Content Views](#) for more information.
8. Optional:
 - a. Create a host collection and assign it to the desired life cycle environment and content view. See [Chapter 16, Configuring Host Collections](#) for more information.
9. Create an activation key assigning it to the desired life cycle environment and content view. See [Section 11.1, “Creating an Activation Key”](#) for more information.
10. Edit an existing provisioning template and associate it with the previously created operating system. See [Section 13.3.8, “Provisioning Templates”](#) for more information.
11. Edit the operating system created by default when creating the content view with the desired details and ensure it is associated with the desired partition table and provisioning template. See [Section 13.3.10, “Operating Systems”](#) for more information.
12. Create a installation medium with the desired details. Ensure that the media is associated with the required locations and organizations. See [Section 13.3.6, “Installation Media”](#) for more information.
13. Create a host group with the desired details. See [Section 13.1, “Creating a Host Group”](#) for more information.

Creating a Backup of a Red Hat Satellite Server

To create a backup of the Red Hat Satellite Server, see [Section 18.1, “Backing up Red Hat Satellite Server”](#).

Chapter 2. Accessing Red Hat Satellite

2.1. Logging in to Red Hat Satellite

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

These steps show how to log in to Red Hat Satellite.

1. Access the **Satellite** server using a web browser pointed to the following address:

https://HOSTNAME/

To identify your hostname, use the **hostname** command at the prompt:

```
# hostname
```



Important

An untrusted connection warning appears on your web browser when accessing **Satellite** for the first time. 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.

Only do this if you are sure that the **Satellite** URL is a trusted source.



This Connection is Untrusted

You have asked Firefox to connect securely to [redacted] but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

- ▶ **Technical Details**
- ▶ **I Understand the Risks**

Figure 2.1. Untrusted Connection Warning

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*.

Result

When you have successfully logged in, you are taken to the **Satellite** dashboard. The dashboard contains an overview of the **Satellite** and the hosts registered.

The main navigation tabs are as follows:

Table 2.1. Navigation Tabs

Navigation Tabs	Description
Organization@Location	Clicking this tab changes the organization and location. If no organization or location is selected, the default organization is <i>Any Organization</i> and the default location is <i>Any Location</i> . Use this tab to change to different values.
Monitor	Provides summary dashboards and reports.
Content	Provides content management tools. This includes Content Views, Activation Keys, and Life Cycle Environments.
Hosts	Provides host inventory and provisioning configuration tools.
Configure	Provides general configuration tools and data including Host Groups and Puppet data.
Infrastructure	Provides tools on configuring how Satellite 6 interacts with the environment.
Administer	Provides advanced configuration for settings such as Users and RBAC, as well as general settings.
User Name	Provides user administration where users can edit their personal information.



Note

If you have forgotten the administrative password, log on to the **Satellite** command-line interface to reset the administration user and password:

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

This will reset the password of the default user *admin* to the one printed on the command line. Change this password upon logging in to prevent any security issues from occurring.

2.2. Changing the Password in Red Hat Satellite

These steps show how to change your password.

Procedure 2.1. Changing Password

1. Click your user name at the top right corner.
2. Select **My Account** from the menu.
3. Type in a new password in the **Password** field.
4. Type in the new password again in the **Verify** field.

5. Click the **Submit** button to save your new password.

Chapter 3. Starting and Stopping Red Hat Satellite

Satellite provides the **katello-service** command to manage Satellite services from the command line. This is useful when upgrading Satellite or when creating a backup, see the [Red Hat Satellite Installation Guide](#) for details on these use cases.

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

```
# katello-service list
```

To see the status of running services, execute:

```
# katello-service status
```

To stop all Satellite services, execute:

```
# katello-service stop
```

To start all Satellite services, execute:

```
# katello-service start
```

To restart all Satellite services, execute:

```
# katello-service restart
```

Chapter 4. Configuring Organizations, Locations and Life Cycle Environments

Red Hat Satellite 6 takes a consolidated approach to Organization and Location management. System administrators define multiple Organizations and multiple Locations in a single Satellite server. For example, a company might have three Organizations (Finance, Marketing, and Sales) across three countries (United States, United Kingdom, and Japan). In this example, the Satellite server manages all Organizations across all geographical Locations, creating nine distinct contexts for managing systems. In addition, users can define specific locations and nest them to create a hierarchy. For example, Satellite administrators might divide the United States into specific cities, such as Boston, Phoenix, or San Francisco.

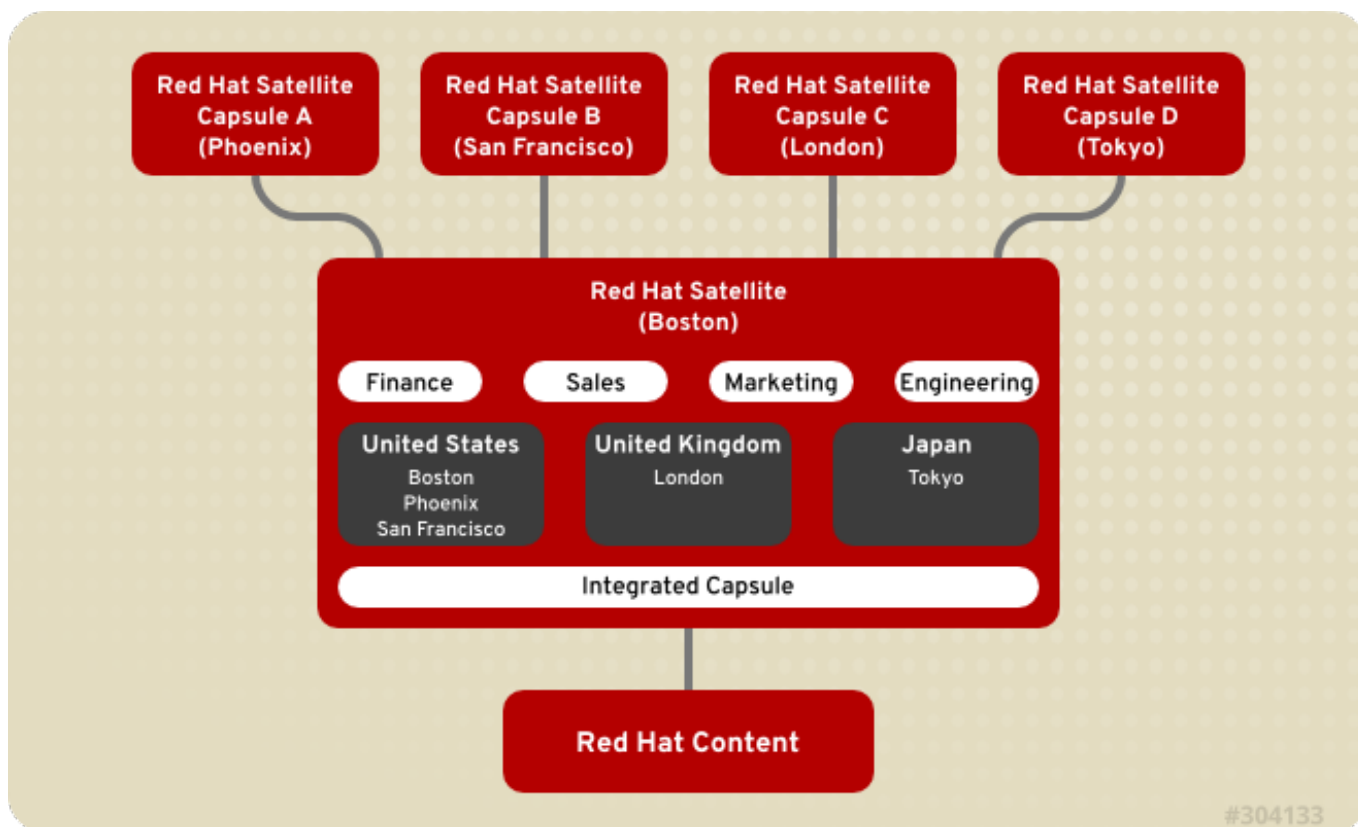


Figure 4.1. Example Topology for Red Hat Satellite 6

The main Satellite server retains the management function, while the content and configuration is synchronized between the main Satellite server and a Satellite Capsule assigned to certain locations.

4.1. Organizations

Organizations divide hosts into logical groups based on ownership, purpose, content, security level, or other divisions.

Multiple organizations can be viewed, created, and managed within the web interface. Software and host entitlements can be allocated across many organizations, and access to those organizations controlled.

Each organization must be created and used by a single Red Hat customer account, however each account can manage multiple organizations. Subscription manifests can only be imported into a single organization and Satellite will not upload a certificate that has already been uploaded into a different organization.

By default, **Red Hat Satellite** will have one organization already created, called "Default Organization", which can be modified to suit your own installation, or deleted. The organization name has a corresponding label **Default_Organization**.



Important

If a new user is not assigned a default organization their access will be limited. To grant systems rights to users, assign them to a default organization and have them log out and log back in again.

4.1.1. Creating an Organization

These steps show how to create a new organization.

Procedure 4.1. To Create an Organization:

1. Navigate to **Administer** → **Organizations**.
2. Click **New Organization**.
3. Insert the name of the new organization in the **Name** field.
4. Insert the label of the new organization in the **Label** field.
5. Insert a description of the new organization in the **Description** field.
6. Click **Submit**.
7. Select the hosts to assign to the new organization.
 - ✦ Click **Assign All** to assign all hosts with no organization to the new organization.
 - ✦ Click **Manually Assign** to manually select and assign the hosts with no organization.
 - ✦ Click **Proceed to Edit** to skip assigning hosts.
8. Specify the configuration details of the organization such as Capsules, subnets or compute resources. You can modify these settings later as described in [Section 4.1.3, "Editing an Organization"](#).
9. Click **Submit**.

4.1.2. Creating an Organization Debug Certificate

These steps show how to generate and download a debug certificate for an organization. Debug certificates unlock all content from an organization and are required for exporting provisioning templates.

Procedure 4.2. To Create a New Organization Debug Certificate:

1. Navigate to **Administer** → **Organizations**.

2. Select an organization for which you want to generate a debug certificate.
3. Click **Generate and Download**. This generates a debug certificate. Save the certificate file in a secure location.



Note

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

4.1.3. Editing an Organization

Procedure 4.3. To Edit an Organization:

1. Navigate to **Administer** → **Organizations**.
2. Click the name of the organization to be edited.
3. Select the resource to edit from the list on the left.
4. Click the name of the desired items to add them to the **Selected Items** list.
5. Click **Submit**.



Note

Users with administrator privileges are not listed under the **Users** tab when editing an organization.

4.1.4. Removing an Organization

Procedure 4.4. To Remove an Organization:

1. Navigate to **Administer** → **Organizations**.
2. Select **Delete** from the drop-down menu to the right of the name of the organization you want to remove.
3. An alert box appears:

Delete *Organization*?

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

4.2. Locations

Locations divide organizations into logical groups based on geographical location. Each location is created and used by a single Red Hat customer account, although each account can manage multiple locations and organizations.

The Red Hat Satellite installation process creates one location, called **Default Location**, which you can modify to suit your own needs. If a new user is not assigned a default location their access will be limited. To grant system rights to users, assign a default location and have them log out and log in again.



Important

You cannot delete the default location, but you can rename it to suit your needs. Satellite returns an error message if you try to delete the default location using either the web UI or the command line.

4.2.1. Creating a Location

These steps show how to create a location.

Procedure 4.5. To Create a Location:

1. Navigate to **Administer** → **Locations**.
2. Click **New Location**.
3. Insert the name of the new location in the **Name** field. If you want to create a nested location, select a **Parent** location from the drop-down menu. Optionally, specify a **Description** of the location. Click **Submit**.
4. Select the hosts to assign to the new location.
 - ✎ Click **Assign All** to assign all hosts with no location to the new location.
 - ✎ Click **Manually Assign** to manually select and assign the hosts with no location.
 - ✎ Click **Proceed to Edit** to skip assigning hosts.
5. Specify the configuration details of the location such as Capsules, subnets or compute resources. You can modify these settings later as described in [Section 4.2.2, “Editing a Location”](#).
6. Click **Submit**.

4.2.2. Editing a Location

Procedure 4.6. To Edit a Location:

1. Navigate to **Administer** → **Locations**.
2. Click the name of the location to be edited.
3. Select the resource to edit from the list on the left.
4. Click the name of the desired items to add them to the **Selected Items** list.
5. Click **Submit**.

4.2.3. Removing a Location

These steps show how to remove an existing location.

Procedure 4.7. To Remove a Location:

1. Navigate to **Administer** → **Locations**.
2. Select **Delete** from the drop-down menu to the right of the name of the location you want to remove.

An alert box appears:

Delete *Location*?

3. Click **OK**.

4.3. Life Cycle Environments

Application life cycles are divided into *life cycle environments*, which represent each stage of the application life cycle. Life cycle environments are linked to form an *environment path*. You can promote content along the environment path to the next life cycle environment when required. For example, if development ends on a particular version of an application, you can promote this version to the testing environment and start development on the next version.

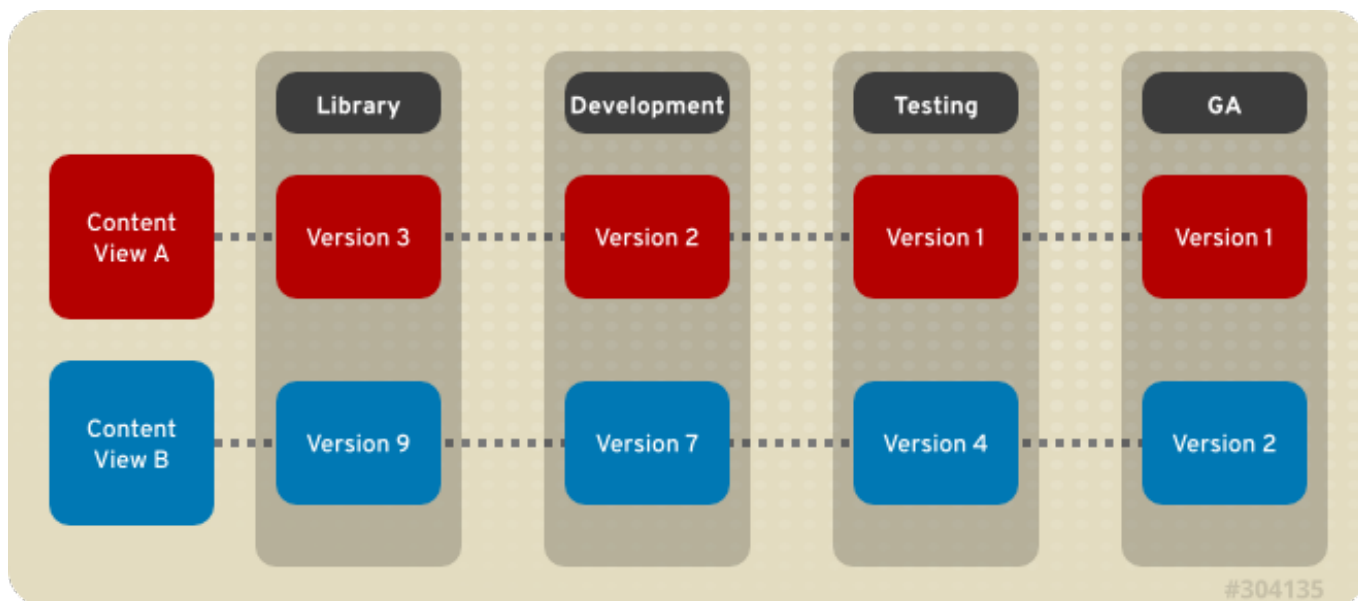


Figure 4.2. An Environment Path Containing Four Environments

4.3.1. Creating Life Cycle Environments

This procedure describes how to create a life cycle environment in Red Hat Satellite.

Procedure 4.8. To Create a Life Cycle Environment:

1. Select an organization from the menu in the top left hand corner.
2. Click **Content** → **Life Cycle Environments** and then click **New Environment Path**.
3. Insert a name and a label (automatically fills in the **Name** field input) for the life cycle environment. The **Description** field is optional.

4. Click **Save** to create the environment.

4.3.2. Promoting Content Views

After you have created a content view and an environment path consisting of two or more life cycle environments, you can promote the content view from one environment to the next as required. This means that the most recent version of the content view that exists in a specified environment will be promoted, or copied, to the next environment in the life cycle environment path.

You can promote a content view to any environment where that version does not exist. The system automatically suggests the next environment in the life cycle environment path, but you can override this and promote to a different environment if required.

Procedure 4.9. To Promote a Content View:

1. On the main menu, click **Content** → **Content Views**.
2. In the **Name** column, click the name of the content view that you want to promote.
3. On the **Versions** tab, identify the latest version, and click **Promote**.
4. Identify the promotion path where you want to promote the content view, select the appropriate life cycle environment, and click **Promote Version**.
5. After the promotion has completed, the **Versions** tab updates to display the new status of your content views.

4.3.3. Removing Life Cycle Environments

This procedure describes how to remove a life cycle environment from Red Hat Satellite.

Procedure 4.10. To Remove a Life Cycle Environment:

1. On the main menu, click **Content** → **Life Cycle Environments**.
2. Click the name of the life cycle environment that you want to remove, and then click **Remove Environment**.
3. In the confirmation dialog box, click **Remove** to remove the environment.



Note

You can only delete the latest environment in an environment path. For example, if three environments exist in the order **Library**, **Dev**, and **Prod**, you need to delete **Prod** before you can delete **Dev**. You cannot delete the **Library** environment.

4.4. Viewing Import History

These steps show how to view an import history in Red Hat Satellite.

Procedure 4.11. Viewing Import History

1. Click **Content** → **Red Hat Subscriptions**.

2. Click the **Manage Manifest** button.
3. Click the **Import History** tab.

Chapter 5. Using the Red Hat Satellite Content Dashboard




The Red Hat Satellite content dashboard provides a status overview of the subscriptions and hosts currently registered, an overview of promotions and synchronization, and a list of the latest notifications.

Navigate to **Monitor** → **Content Dashboard** to access the content dashboard. The dashboard can be rearranged by clicking on a section title and dragging the section to a different position. The following sections are available:

Content 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 5.1. Host Subscription States

State	Description	Icon
Invalid Subscriptions	Hosts that have products installed, but have not consumed a subscription. These hosts need attention immediately.	
Insufficient Subscriptions	Hosts that have consumed a subscription and have a valid entitlement, but that are not consuming their full entitlements. These hosts should be monitored to ensure they are configured as expected.	
Current Subscriptions	Hosts that have a valid entitlement and are consuming their full entitlements.	

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

Latest Notifications

A list of messages produced by hosts including administration information, product and subscription changes, and any errors. Click the gear button to change the number of notifications displayed.

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

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. Click a product name to view the synchronization status. Click the gear button to change the number of notifications displayed.

Host Collections

A list of all host collections in Satellite and their status, including the number of content hosts in each host collection. Click a host collection name to view that host collection. Click the gear button to change the number of notifications displayed.

Current Subscription Totals

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. Click the number to list subscriptions of the selected type.

Content Views Overview

A list of all Content Views in Satellite and their publish status. Click the gear button to change the number of notifications displayed.

Errata Overview

A list of all errata available for hosts registered to Satellite. Click the gear button to change the number of notifications displayed.

Chapter 6. Using Content Views

Content views are managed selections of content, which contain one or more repositories (yum, puppet, or containers) with optional filtering. These filters can be either inclusive or exclusive, and tailor a system view of content for life cycle management. They are used to customize content to be made available to client systems.

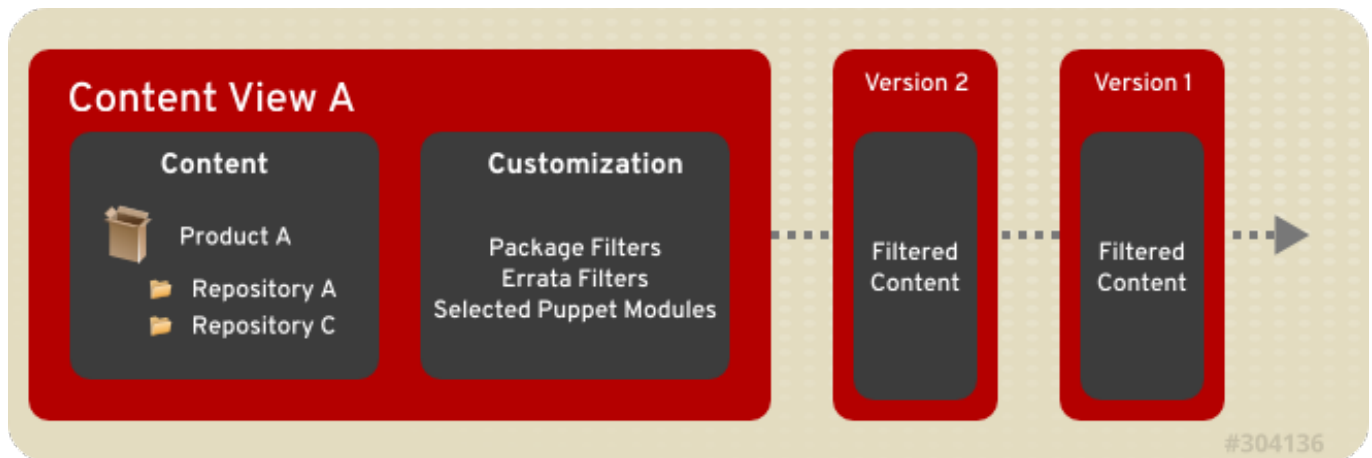


Figure 6.1. This diagram details the creation of new versions of a Content View. These content view versions are promoted along an environment path during the application life cycle.

Published content views are used with life cycle environments.

6.1. Creating a Content View

A user with administrator privileges can create content views for use within the life cycle environments.

Procedure 6.1. To Create a Content View:

1. Log in as a Satellite administrator.
2. Click **Content** → **Content Views**.
3. Click **Create New View**.
4. Specify the **Name** of the content view. The **Label** field is automatically populated when the **Name** field is filled out. Optionally, provide a description of the content view.
5. Select the **Composite View** check box to combine a series of published content views into one and choose which content view.



Note

If you select **Composite View** it will override any filtering and allow you to choose a group of published content views and bundle those views into a composite one.

6. Click **Save**.

6.2. Adding Repositories to the Content View

A repository provides storage for content. For example, a YUM repository, Puppet repository, or a Docker repository.

Procedure 6.2. To Associate a Repository with a Content View:

1. Click **Content** → **Content Views** and choose the Content View to add repositories to.
2. Depending on the type of content you want to store:
 - ✦ Click **Yum Content** and select **Repositories** from the drop-down menu. From the submenu, click **Add**.
 - ✦ Click **Puppet Modules** and click **Add New Module**.
 - ✦ Click **Docker Content** and click **Add** in the submenu.
3. Select the repositories to add and click **Add Repositories**.

6.3. Filtering Content

Filters provide a mechanism to prevent packages from being promoted to subsequent environments. You can use package names or regular expressions in the filter to create the rules to blacklist packages. Then you can associate the filter to entire products or individual repositories within any product.

6.3.1. Creating a Filter

The following procedure shows how to create a filter for packages.

Procedure 6.3. To Create a Filter:

1. Navigate to **Content** → **Content Views** and select the Content View you want to filter.
2. Click **Yum Content** → **Filters** and click **New Filter**.
3. Specify the name of the new filter in the **Name** field and choose a content type from the **Content Type** drop-down menu. Choose whether the filter includes or excludes the selected content type by selecting the **Type** drop-down menu. Optionally, insert a description in the **Description** field.
4. Click **Save** to save your new filter.

6.3.2. Adding Content to a Filter

The following procedure shows how to add content to a package filter.

Procedure 6.4. To Add Content to a Filter:

1. Navigate to **Content** → **Content Views** and select the Content View you want to filter.
2. Click **Yum Content** → **Filters** and click the name of the filter you want to edit. Depending on the type of filter selected, perform the following actions:
 - a. If the filter is made for packages, specify a package name on the **Packages**

- subtab, and select a **Detail** value from the drop-down menu. Click **Add** to add the package to the filter.
- b. If the filter is made for package groups, click the **Add** subtab, and choose the desired package group. Click **Add Package Group**.
 - c. If the filter is made for errata, click the **Add** subtab. Select the errata type (**Security**, **Enhancement**, or **Bugfix**), and specify a start date and end date. Click **Add Errata**.
 - d. If the filter is made for errata - date and type, on the **Erratum Date Range** subtab, select the errata type (**Security**, **Enhancement**, or **Bugfix**) and specify a start date and end date. Click **Save**.
3. On the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, select the desired repositories and click **Update Repositories**.
 4. Click **Publish New Version**. Insert a comment if desired, then click **Save**.

6.3.3. Removing Content from a Filter

The following procedure shows how to remove content from a package filter.

Procedure 6.5. To Remove Content from a Filter:

1. Navigate to **Content** → **Content Views** and select the Content View you want to filter.
2. Click **Yum Content** → **Filters** and click the name of the filter you want to edit. Depending on the type of filter selected, perform the following actions:
 - a. If the filter is made for packages, click the **Packages** subtab and select the **Package Name** check box next to the package to be removed. Click **Remove Packages** to remove the package from the filter.
 - b. If the filter is made for package groups, click the **List/Remove** subtab and select the **Name** check box next to the package group to be removed. Click **Remove Package Group** to remove the package group from the filter.
 - c. If the filter is made for errata, click the **List/Remove** subtab select the **Errata ID** check box next to the errata to be removed. Click **Remove Errata** to remove the errata from the filter.
 - d. If the filter is made for errata - date and type, on the **Erratum Date Range** subtab, check the errata type (**Security**, **Enhancement**, or **Bugfix**). Specify the start date and end date. Click **Save**.
3. On the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, select the desired repositories and click **Update Repositories**.
4. Click **Publish New Version**. Insert a comment if desired, and click **Save**.

6.3.4. Removing a Filter

The following procedure shows how to remove a filter.

Procedure 6.6. To Remove a Filter:

1. Navigate to **Content** → **Content Views** and select the Content View you want to filter.
2. Click **Yum Content** → **Filters** and select the check box next to the name of the package filter you want to remove.
3. Click **Remove Filters**.

6.4. Publishing a Content View

After a content view has been created, it needs to be published in order for it to be visible and usable by hosts. Before publishing the content view definition, make sure that the content view definition has the necessary products, repositories and filters.

Procedure 6.7. To Publish a Content View Definition:

1. Click **Content** → **Content Views**.
2. Click on the content view to be published.
3. Click **Publish New Version**.
4. Fill in a comment.
5. Click **Save**.

Chapter 7. Searching for Content

7.1. Basic Content Search

Content Search allows users to look for specific information about content views, products, repositories, or errata.

Procedure 7.1. To Perform a Content Search:

1. Click **Content** → **Content Search**.
2. Select either **Content Views**, **Products**, **Repositories**, **Packages**, **Errata**, or **Puppet Modules** from the **Content** drop-down menu.
3. Enter the name of the content view, product, repository, package, errata, or puppet module you are searching for in the **Products** field, and click **Search**.

7.2. Content Comparison across Environments

You can compare content across different environments using the *content search* feature.

Procedure 7.2. To Search for and Compare Content Across Different Environments:

1. Click **Content** → **Content Search**.
2. Select the entity type that you want to compare from the **Content** drop-down menu.
3. Enter the name of the entity in the **Products** field and click **Add**.
4. On the right panel, move your cursor over the "plus" (+) icon, select the environments you want to compare, and click **Search**.
5. Select either **Union**, **Intersection**, or **Difference** from the **View** drop-down menu to filter your results.

Chapter 8. Viewing and Applying Errata

Software packages in Red Hat products are subject to updates, referred to as *errata*, that are released at regular intervals as well as asynchronously. Red Hat Satellite provides tools to inspect and filter errata, allowing for precise update management. This way, you can select relevant updates and propagate them through content views to selected content hosts. See [Chapter 6, Using Content Views](#) for more information on content views.



Important

Install the *katello-agent* package on the Satellite server as described in [Section 14.3.2, “Installing the Katello Agent”](#). This package provides the necessary services for errata management.

Before applying the latest updates, make sure you have correctly synchronized the Satellite content. For more information on synchronizing content for connected or disconnected Satellite see the [Populating Red Hat Satellite with Content](#) section in the [Red Hat Satellite Installation guide](#). Navigate to **Monitor** → **Content Dashboard** to see the overview of errata synchronization.

Errata contain advisories that describe the changes introduced by the update. There are three types of advisories (in order of importance):

- **Security Advisory** describes fixed security issues found in the package. The security impact of the issue can be *Low*, *Moderate*, *Important*, or *Critical*.
- **Bug Fix Advisory** describes bug fixes for the package.
- **Product Enhancement Advisory** describes enhancements and new features added to the package.



Note

Errata are labeled according to the most important advisory type they contain. Therefore, errata labeled as *Product Enhancement Advisory* can contain only enhancement updates, while *Bug Fix Advisory* errata can contain both bug fixes and enhancements, and *Security Advisory* can contain all three types.

In Red Hat Satellite, there are two keywords that describe an erratum's relationship to the available content hosts:

- **Applicable**: erratum applies to one or more content hosts, which means it updates packages present on the content host. Applicable errata are not yet accessible by the content host.
- **Installable**: erratum applies to one or more content hosts and it has been made available to the content host. Installable errata are present in the content host's life cycle environment and content view, but are not yet installed. This way, errata can be installed by users who have permissions to manage content hosts, but are not entitled for errata management at higher levels.

8.1. Inspecting Available Errata

The following procedure describes how to view and filter the available errata and how to display metadata of the selected advisory.

Procedure 8.1. To Inspect Available Errata:

1. Navigate to **Content** → **Errata** to view the list of available errata.
2. Use the filtering tools at the top of the page to limit the number of displayed errata:
 - ✧ Select the repository to be inspected from the drop-down list. **All Repositories** is selected by default.
 - ✧ The **Applicable** check box is selected by default to view only errata applicable to the selected repository. Select the **Installable** check box to view only errata marked as installable.
 - ✧ To search the table of errata, type the query in the **Search** field in the form of:

```
parameter operator value
```

See [Table 8.1, “Parameters Available for Errata Search”](#) for the list of parameters available for search. Find the list of applicable operators in [Table 17.2, “Supported Operators for Granular Search”](#). Automatic suggestion works as you type. You can also combine queries with the use of *and* and *or* operators. For example, to display only security advisories related to the *kernel* package, type:

```
type = security and package_name = kernel
```

Press **Enter** to start the search.

3. Click the **Errata ID** of the erratum you want to inspect:
 - ✧ The **Details** tab contains the description of the updated package as well as documentation of important fixes and enhancements provided by the update.
 - ✧ On the **Content Hosts** tab, you can apply the erratum to selected content hosts as described in [Section 8.2, “Applying Errata to Content Hosts”](#).
 - ✧ The **Repositories** tab lists repositories that already contain the erratum. You can filter repositories by the environment and content view, and search for them by the repository name.

Table 8.1. Parameters Available for Errata Search

Parameter	Description	Example
bug	Search by the Bugzilla number.	bug = BZ#1172165
cve	Search by the CVE number.	cve = CVE-2015-0235
id	Search by the errata ID. The auto-suggest system displays a list of available IDs as you type.	id = RHBA-2014:2004
issued	Search by the issue date. You can specify the exact date, like "Feb16,2015", or use keywords, for example "Yesterday", or "1 hour ago". The time range can be specified with the use of the "<" and ">" operators.	issued < "Jan 12,2015"

Parameter	Description	Example
package	Search by the full package build name. The auto-suggest system displays a list of available packages as you type.	package = glib2-2.22.5-6.el6.i686
package_name	Search by the package name. The auto-suggest system displays a list of available packages as you type.	package_name = glib2
severity	Search by the severity of the issue fixed by the security update. Specify <i>Critical</i> , <i>Important</i> , or <i>Moderate</i> .	severity = Critical
title	Search by the advisory title.	title ~ openssl
type	Search by the advisory type. Specify <i>security</i> , <i>bugfix</i> , or <i>enhancement</i> .	type = bugfix
updated	Search by the date of the last update. You can use the same formats as with the <i>issued</i> parameter.	updated = "6 days ago"

8.2. Applying Errata to Content Hosts

The following procedures show how to apply one or more errata to content hosts.

Procedure 8.2. To Apply a Single Erratum to Content Hosts:

1. Navigate to **Content** → **Errata** to view the list of available errata.
2. Click the **Errata ID** of the erratum you want to apply.
3. On the **Content Hosts** tab, select one or more content hosts to be updated. You can filter the available content hosts by the environment, and search for them by name. If you select the check box at the top of the page, only the content hosts that already have the installable erratum in their life cycle environment are displayed.
4. Click **Apply to Hosts**.
 - ✦ If the erratum is *applicable*, a new minor version of the content view is created. If you select **Apply Errata to Content Hosts Immediately after publishing**, Satellite will automatically install the erratum on the content host when promoting the updated content view. Otherwise, the erratum will be made available for installation on the content host. Installable errata can be applied later using the same procedure, or manually per content host as described in [Procedure 8.4, "To Apply Installable Errata to a Content Host:"](#).
 - ✦ If the erratum is *installable*, which means it is already present in the selected content host's life cycle environment but is not installed yet, no new content view version is created.
5. Click **Confirm**.

Procedure 8.3. To Apply Multiple Errata to Content Hosts:

1. Navigate to **Content** → **Errata** to view the list of available errata.
2. Select errata you want to apply by selecting the check box to the left of the **Errata ID** field.
3. Click **Apply Errata** to apply all selected errata.

4. Select one or more content hosts to be updated. You can filter the available content hosts by the environment, and search for them by name. If you select the check box at the top of the page, only content hosts that already have the installable errata in their life cycle environment are displayed.
5. Click **Next**. If some of the selected errata are *applicable*, a new minor version of the content view is created. If you select **Apply Errata to Content Hosts Immediately after publishing**, Satellite will automatically install errata on the content host when promoting the updated content view. If only installable errata are selected, they are installed without creating a new content view version.

If the content host's life cycle environment contains installable errata, you can install them from the **Content Hosts** page as described in [Procedure 8.4, "To Apply Installable Errata to a Content Host:"](#) This way, errata can be applied by users who have permissions to manage content hosts, but are not entitled for errata management at higher levels. Similarly, you can apply installable errata to host collections as described in [Section 16.3.2, "Adding Errata to a Host Collection"](#).

Procedure 8.4. To Apply Installable Errata to a Content Host:

1. Navigate to **Hosts** → **Content Hosts**.
2. Click the name of the content host you want to manage.
3. On the **Errata** tab, select errata you want to install.
4. Click **Apply Selected** to install the selected updates.

8.3. Subscribing to Errata Notifications

You can configure email notifications for Satellite users as described in [Section 17.1.4, "Configuring Email Notifications"](#). Users can receive a summary of applicable and installable errata, notifications on content view promotion or after synchronizing a repository.

Chapter 9. Security Compliance Management with OpenSCAP

The Security Content Automation Protocol (SCAP) enables the definition of configuration and security policies, also the means of auditing for compliance with those policies. In Satellite 6, SCAP is implemented with the tools provided by the OpenSCAP project. For more information about OpenSCAP see the [Red Hat Enterprise Linux 7 Security Guide](#)

OpenSCAP provides the means of conducting compliance auditing across your managed environment. Configuration and security policies are expressed in a SCAP-compliant format and hosts are checked for compliance. The Satellite web UI provides the means of compliance auditing and tools to analyse non-compliance. Scheduled auditing against policies ensures that non-compliant hosts are identified, even if they were previously compliant.

The following specifications are supported by OpenSCAP:

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

9.1. Installation

The high-level installation steps for OpenSCAP are:

- ✧ Install the OpenSCAP packages on the Satellite server.
- ✧ Install the OpenSCAP packages on all Satellite Capsule servers.
- ✧ Import the Puppet classes and associate them with specific environments.



Note

If OpenSCAP functionality is to be enabled on a Satellite Capsule server, Puppet must already have been enabled on that server.

Procedure 9.1. Install OpenSCAP

1. On the Satellite server, install the *ruby193-rubygem-foreman_openscap* RPM package.
2. Restart the **httpd** service.

On Red Hat Enterprise Linux 7

```
# systemctl restart httpd
```

On Red Hat Enterprise Linux 6

```
# service httpd restart
```

This action adds to the Satellite web UI a **Compliance** section, under the **Hosts** menu, containing the following pages:

- » **Policies**
- » **SCAP Contents**
- » **Reports**

3. On the Satellite server and all Satellite Capsule servers, install the **puppet-foreman_scap_client** and **rubygem-smart_proxy_openscap** RPM packages.

The **puppet-foreman_scap_client** package provides the Puppet classes required to set up hosts to perform scans via OpenSCAP and creates the Cron job for periodic scanning as specified by the applicable policy.

4. On the Satellite server and all Satellite Capsule servers, restart the **foreman-proxy** service.

Red Hat Enterprise Linux 7

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

Red Hat Enterprise Linux 6

```
# service foreman-proxy restart
```

5. In the Satellite web UI, select **Configure** → **Puppet classes** → **Import from *SATELLITE_HOST***. Select the line with the new module and click **Update** to load the module.

9.2. SCAP Concepts

9.2.1. SCAP Content

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

You can either create SCAP content or obtain it from a vendor. A number of supported profiles are provided for Red Hat Enterprise Linux in the *scap-security-guide* package. The creation of SCAP content is outside the scope of this guide, but see the [Red Hat Enterprise Linux 7 Security Guide](#) or [Red Hat Enterprise Linux 6 Security Guide](#) for information on how to

download, deploy, tailor, and define your own content using the SCAP Workbench. The SCAP content provided with Red Hat Enterprise Linux is compliant with SCAP specification 1.2.

If you install the OpenSCAP components of Satellite 6 on Red Hat Enterprise Linux 6, default SCAP content will be installed for Red Hat Enterprise Linux 6. If you install the OpenSCAP components of Satellite 6 on Red Hat Enterprise Linux 7, default SCAP content will be installed for both Red Hat Enterprise Linux 6 **and** Red Hat Enterprise Linux 7.

9.2.2. XCCDF Profile

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

To list all available profiles, open the Satellite web UI, navigate to **Hosts** → **Policies**, select **Edit** from the drop-down list next to the policy of interest and select the **SCAP Content** tab. Select the **SCAP Content** of interest and browse the available profiles in the **XCCDF Profile** drop-down list.

The profiles provided with Satellite 6 are obtained from the SCAP Security Guide project, which is hosted at <https://fedorahosted.org/scap-security-guide>.

9.2.3. Compliance Policy

A compliance policy is the application of specific SCAP content and XCCDF profile to one or more host groups, on a set schedule. The schedule on which a scan is run is specified by the Satellite server but the scan itself occurs on the host. When the scan is complete, an *Asset Reporting File* (ARF) is output in XML format and uploaded to the Satellite server. You can see the results of the scan in the compliance policy dashboard.

The OpenSCAP content includes several profiles and their associated rules but no policies are included by default. For details on how to create a policy, see [Section 9.2.5, “Creating a Policy”](#).

9.2.4. Elements of a Compliance Policy

A compliance policy specifies the following:

- ✧ SCAP Content (including the XCCDF profile)
- ✧ Schedule at which the policy will be run on the target host(s)
- ✧ Locations, organizations and host groups to which it applies

The **SCAP Content** tab provides the option of selecting the SCAP content and XCCDF profile for this policy. Once you have selected these, the **SCAP Content** tab provides the name of the SCAP content file which will be distributed to the directory `/var/lib/openscap/content/` on all target hosts.

Figure 9.1. Elements of a Compliance Policy

9.2.5. Creating a Policy

Follow these steps to create a compliance policy, which specifies the SCAP content and profile to be applied to a location and either a host or host group at a specified time.

Procedure 9.2. To Create a Policy:

1. In the Satellite web UI, navigate to **Hosts** → **Policies**, click **New Compliance Policy** and follow the wizard's steps.
2. Enter a name for this policy, a description (optional), then click **Next**.
3. Select the SCAP Content and XCCDF Profile to be applied, then click **Next**.
4. Specify the scheduled time when the policy is to be applied, then click **Next**.

Select **Weekly**, **Monthly** or **Custom** from the **Period** drop-down list.

- ✦ If you select **Weekly**, also select the desired day of the week from the **Weekday** drop-down list.
- ✦ If you select **Monthly**, also specify the desired day of the month in the **Day of month** field.
- ✦ If you select **Custom**, enter a valid Cron expression in the **Cron line** field.

The **Custom** option allows for greater flexibility in the policy's schedule than either the **Weekly** or **Monthly** options.

5. Select the location(s) to which the policy is to be applied, then click **Next**.
6. Select the organizations to which the policy is to be applied, then click **Next**.
7. Select the host group(s) to which the policy is to be applied, then click **Next**.
8. Click **Submit**.

When the Puppet agent runs on the hosts which belong to the selected host group, or hosts to which the policy has been applied, the OpenSCAP client will be installed and a Cron job added with the policy's specified schedule.

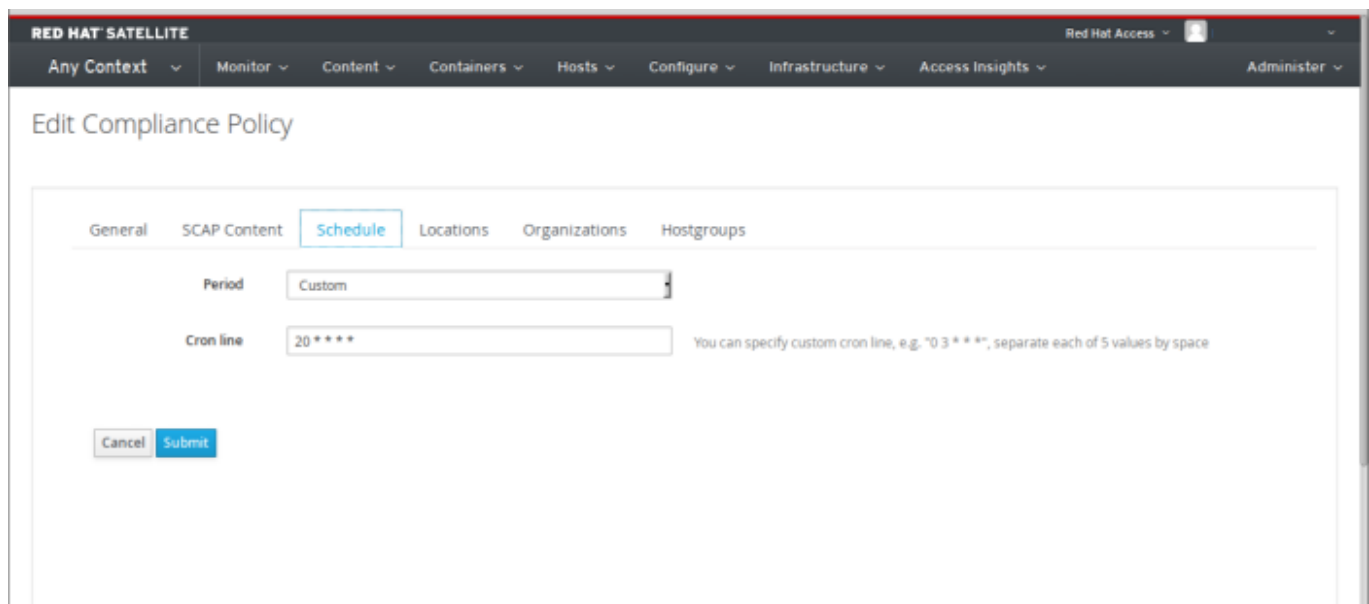


Figure 9.2. Creating a Compliance Policy

9.2.6. Viewing a Policy

Follow these steps to preview the rules which will be applied by specific OpenSCAP content and profile combination. This is useful when planning policies.

1. In the Satellite web UI, navigate to **Hosts → Policies**.
2. Click **Show Guide**.

9.2.7. Editing a Policy

Follow these steps to edit an existing policy.

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

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

9.2.8. Deleting a Policy

Follow these steps to delete an existing policy.

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

9.2.9. Compliance Policy Dashboard

The compliance policy dashboard provides an overview of hosts' compliance with a policy. To view a compliance policy's dashboard, open the Satellite web UI and navigate to **Hosts** → **Policies**, then click the policy's name. The dashboard provides the following information:

- ✧ A ring chart illustrating a high-level view of hosts' compliance with the policy.
- ✧ A statistical breakdown of hosts' compliance with the policy, in tabular format.
- ✧ Links to the policy's latest reports.

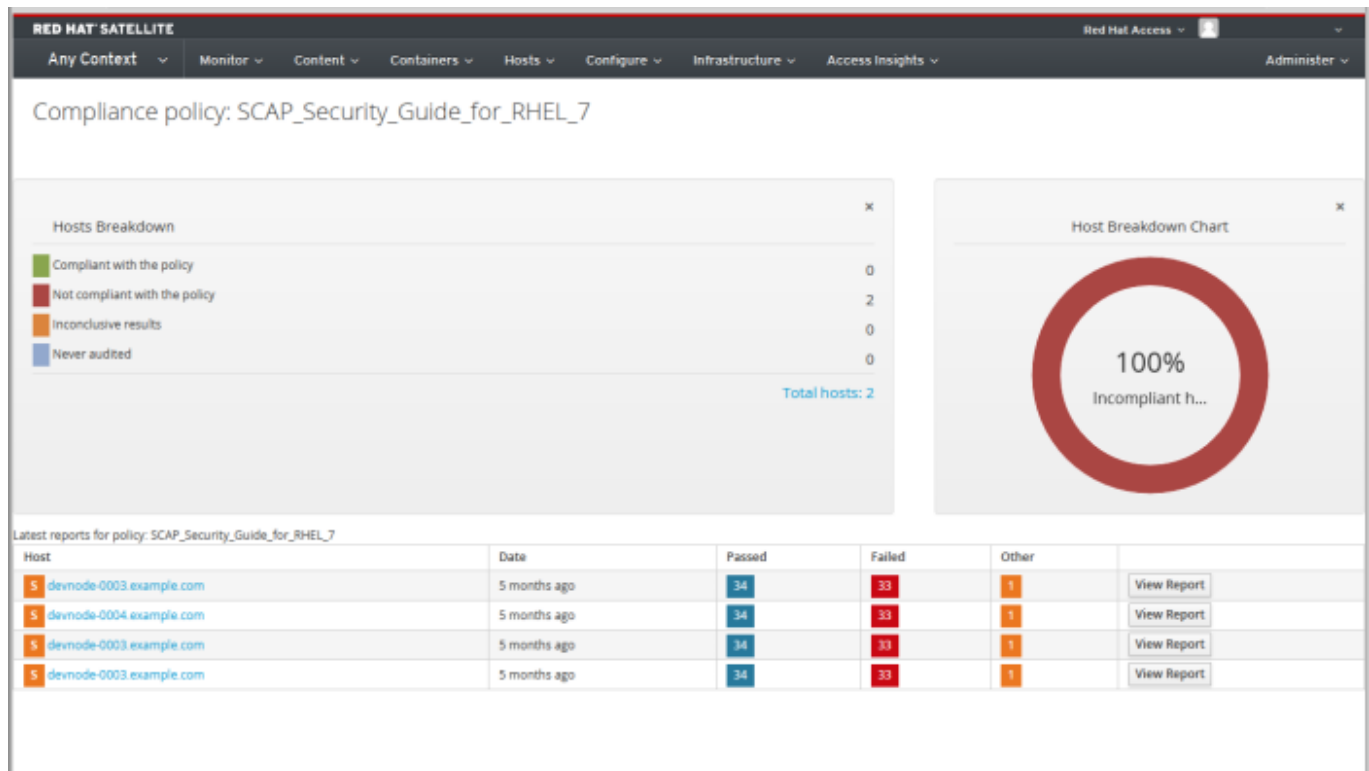


Figure 9.3. Compliance Policy Dashboard

9.3. Compliance Reports Overview

A Compliance report is the output of a policy run against a host. From the **Compliance Reports** page you can view individual reports or filter the list of available reports. All reports are listed in descending date order. For each report the total number of rules passed or failed per policy are listed. Click on each column's label to sort the list by that column, in either descending or ascending order.

All Compliance reports are available in the Satellite web UI via **Hosts** → **Reports**.

To delete a compliance report, select **Delete** from the drop-down list beside **View Report**.

Host	Date	Passed	Failed	Other	
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report

Figure 9.4. Compliance Reports Overview

9.3.1. Searching Compliance Reports

To narrow the list of compliance reports, enter search criteria in the **Search** field and either press Enter or click **Search**. The search performed is case-insensitive. Click on the empty **Search** field to see a list of available search parameters.

See [Table 17.2, “Supported Operators for Granular Search”](#) for details of all available search operators. You can create complex queries with the logical operators: **and**, **not** and **has**.

Logical Operators

- ✳ **not**: Negates an expression.
- ✳ **has**: Object must have a specified property.
- ✳ **and**: Combines search criteria.

The following search criteria finds all compliance reports for which more than five rules failed.

```
failed > 5
```

Regular expressions are not valid search criteria, however multiple fields can be searched. For example, the following query searches for OpenSCAP reports generated by the compliance_policy **rhel7_audit** from an hour ago.

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

To again list *all* available compliance reports, delete the **Search** criteria and press Enter or click **Search**.

Bookmarking Your Searches

You can bookmark a search, allowing you to easily apply the same search criteria. To create a bookmark:

Procedure 9.3. To Bookmark a Search

1. Apply your search criteria.
2. From the **Search** list select **Bookmark this search**.
3. Complete the **Name** field.

If you want the bookmark available to other users of this Satellite instance, select the **Public** check box.

4. Click **Submit**.

To use a bookmark, navigate to **Hosts** → **Reports**, click the drop-down item beside the **Search** button and click the bookmark.

9.3.2. Viewing a Compliance Report

Navigate to **Hosts** → **Reports** and click **View Report** in the row of the specific host.

A compliance report consists of the following sections:

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

9.3.2.1. Evaluation Characteristics

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

Evaluation Characteristics

Target machine

The fully-qualified domain name (FQDN) of the evaluated host. Example: **test-system.example.com**.

Benchmark URL

The URL of the SCAP content against which the host was evaluated. Example: **/var/lib/openscap/content/1fbdc87d24db51ca184419a2b6f**.

Benchmark ID

The identifier of the benchmark against which the host was evaluated. A benchmark is a set of profiles. Example: **xccdf_org.ssgproject.content_benchmark_RHEL_7**.

Profile ID

The identifier of the profile against which the host was evaluated. Example: `xccdf_org.ssgproject_content_profile_rht-ccp`.

Started at

The date and time at which the evaluation started, in ISO 8601 format. Example: `2015-09-12T14:40:02`.

Finished at

The date and time at which the evaluation finished, in ISO 8601 format. Example: `2015-09-12T14:40:05`.

Performed by

The local account name under which the evaluation was performed on the host. Example: `root`.

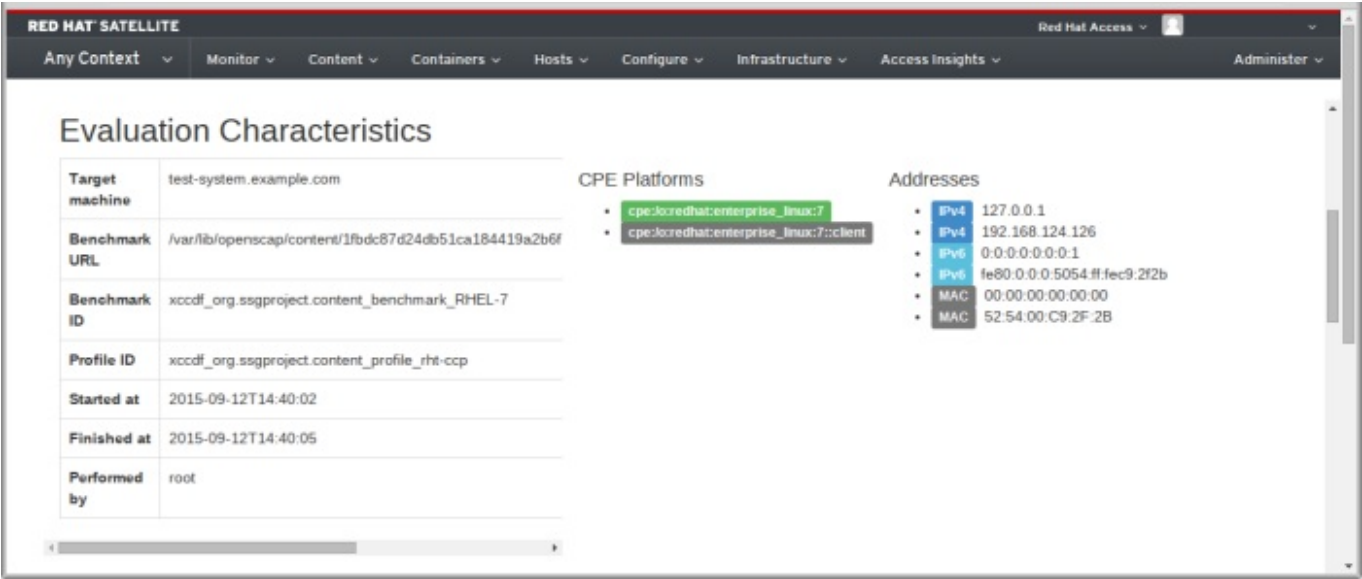


Figure 9.5. Evaluation Characteristics

9.3.2.2. Compliance and Scoring

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

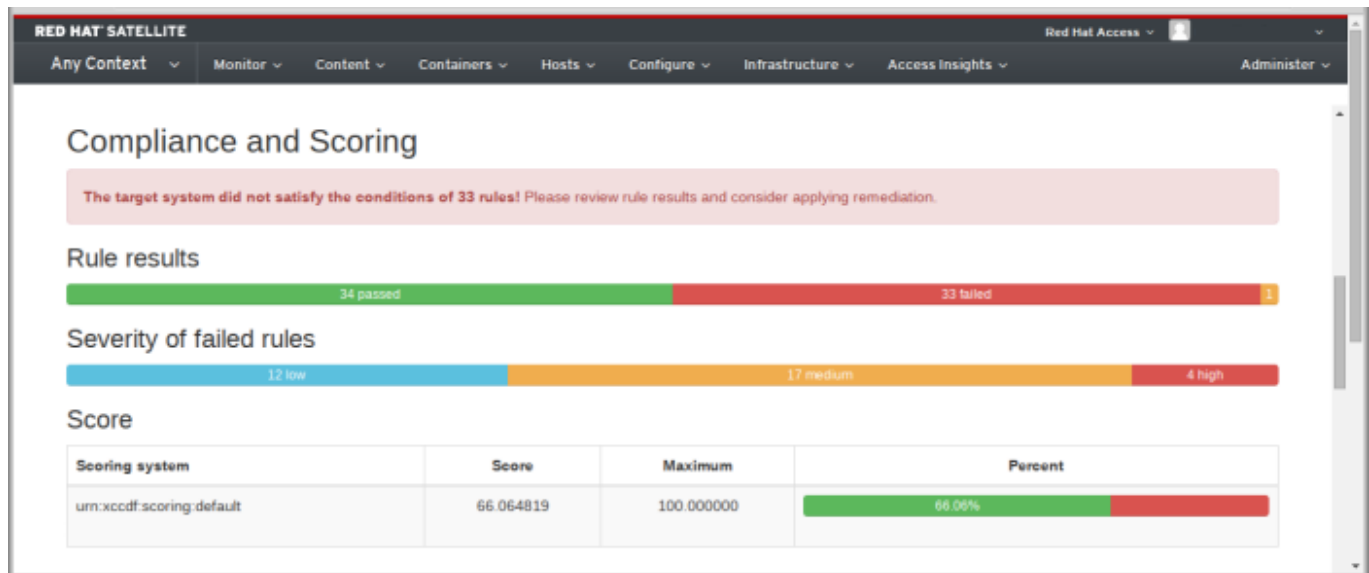


Figure 9.6. Compliance and Scoring

9.3.2.3. Rule Overview

This section provides details of every rule and the compliance result, with the rules presented in a hierarchical layout.

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

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

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

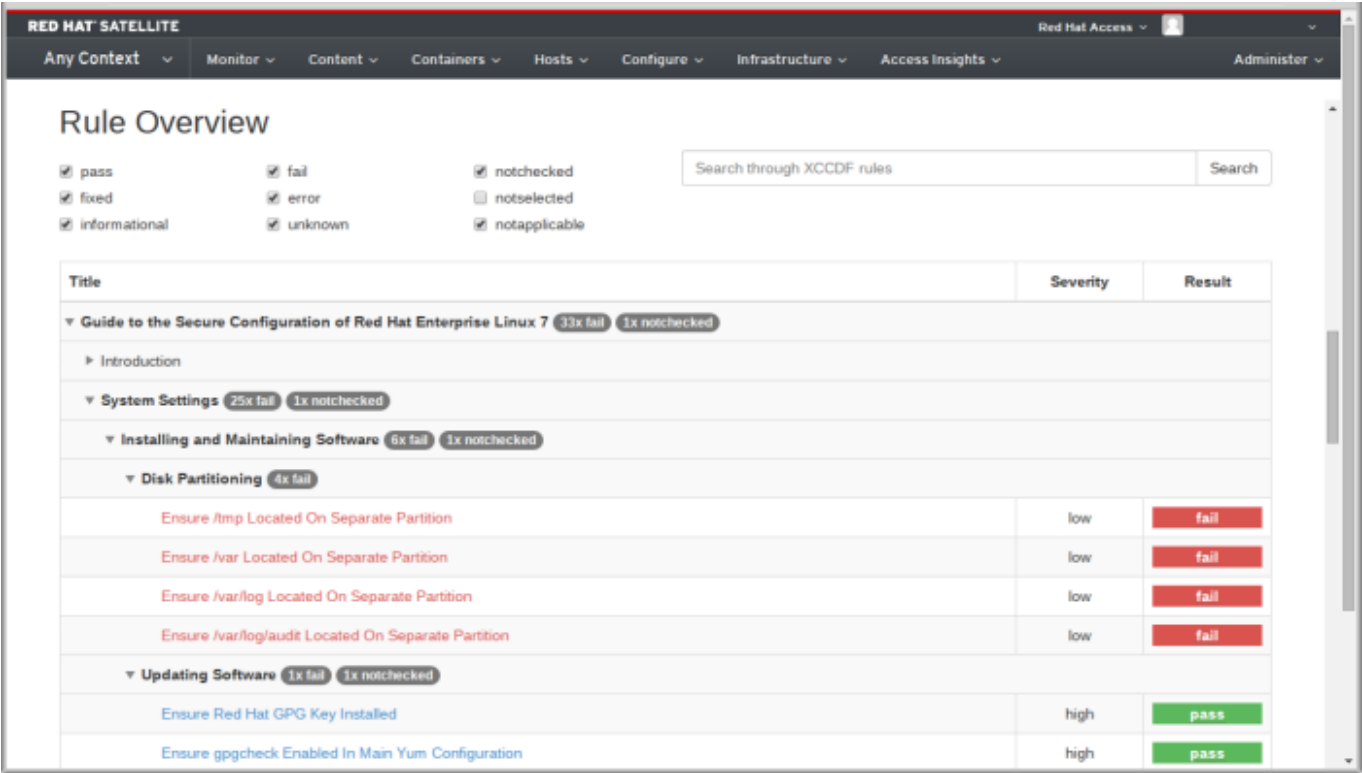


Figure 9.7. Rule Overview

9.3.2.4. Examining Rule Results

To determine why a host failed compliance on a rule, click on the rule's title. The window which then opens provides further details, including: a description of the rule (optionally instructions for bringing the host into compliance), the rationale for the rule, and optionally a remediation script.

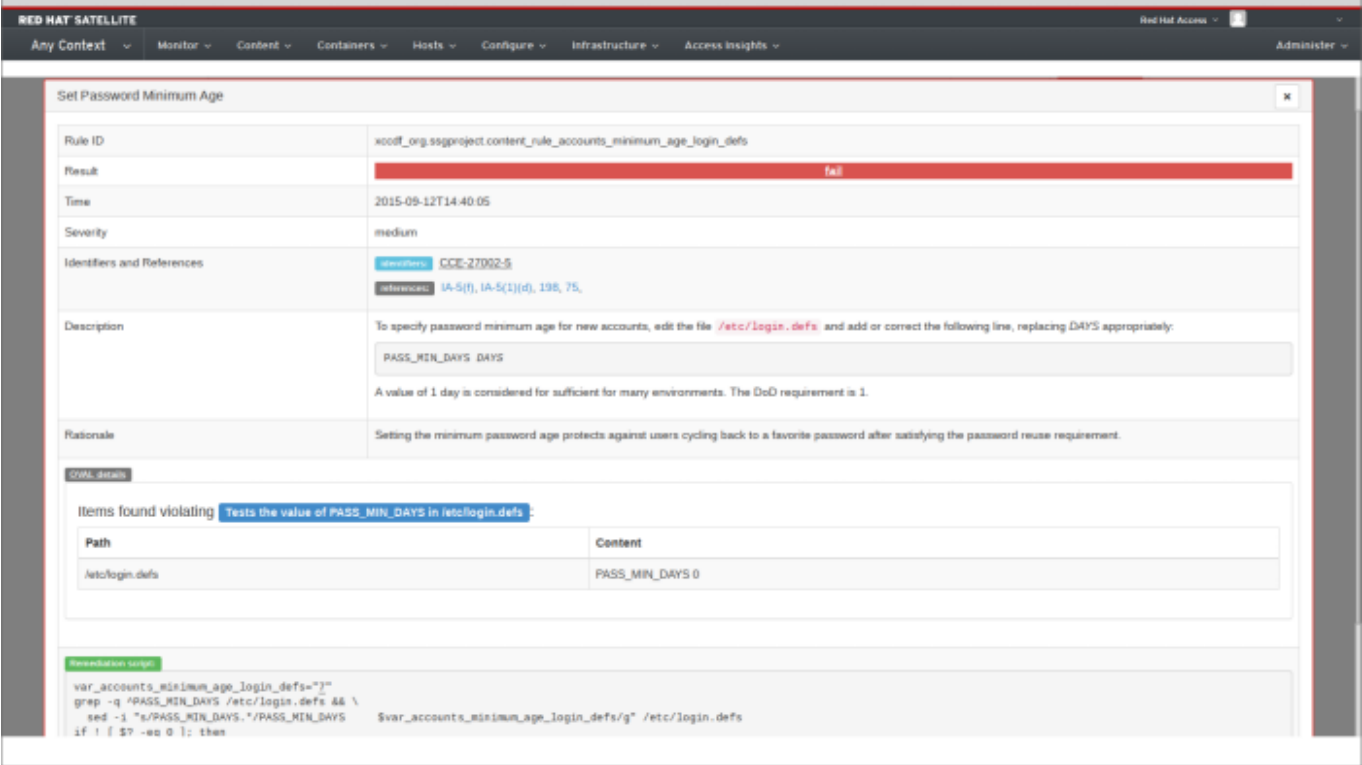


Figure 9.8. Rule Evaluation Result



Warning

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

9.4. Uploading Additional SCAP Content

You can upload additional SCAP content into the Satellite server, either content created by yourself with the SCAP Workbench or obtained elsewhere. Additional SCAP content must be imported into the Satellite server before being applied in a policy. For example, the *scap-security-guide* RPM package available in the Red Hat Enterprise Linux 7.2 repositories includes a profile for the *Payment Card Industry Data Security Standard* (PCI-DSS) version 3. You can upload this content into a Satellite server even if it is not running Red Hat Enterprise Linux 7.2 as the content is not specific to an operating system version.

Procedure 9.4. To Upload Additional SCAP Content:

1. Log in to the Satellite web UI.
2. Navigate to **Hosts** → **SCAP contents** and click **Upload New SCAP Content**.
3. Enter a title in the **Title** text box. For example: **RHEL 7.2 SCAP Content**.
4. Click **Choose file**, navigate to the location containing the SCAP content file and select **Open**.
5. Click **Submit**.

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

Chapter 10. Working with Containers

Docker is an open source project that automates the deployment of applications inside *Linux containers*, and provides the capability to package an application with its runtime dependencies into a container. Linux containers enable rapid application deployment, simpler testing, maintenance, and troubleshooting while improving security. For more information, see the [Get Started with Docker Formatted Container Images on Red Hat Systems](#) article on the Red Hat Customer Portal ^[1].

A container in the Docker format is composed of the following parts:

- **Container:** An application sandbox. Each container is based on an image that holds necessary configuration data. When you launch a container from an image, a writable layer is added on top of this image. Every time you commit a container a new image layer is added to store your changes.
- **Image:** A static snapshot of the container's configuration that is never modified. Any changes made to the container can be saved only by creating a new image layer. Each image depends on one or more parent images.
- **Platform image:** An image that has no parent. Platform images define the runtime environment, packages and utilities necessary for containerized applications to run. The platform image is not writable, so any changes are reflected in the copied images stacked on top of it. For information on how to access Red Hat Enterprise Linux platform images from Red Hat Satellite see [Example 10.1, “Creating a Red Hat Enterprise Linux Container in Satellite”](#).
- **Registry:** A public or private archive that contains images available for download. Some registries allow users to upload images to make them available to others. Red Hat Satellite allows you to import images from local and external registries. Satellite itself can act as an image registry for hosts, however, hosts cannot push changes back to the registry. For more information, see [Section 10.1.1, “Creating Containers”](#).
- **Tag:** A mark used to differentiate images in a repository, typically by the version of the application stored in the image. Repositories are used to group similar images in a container registry. Images only have unique alphanumeric identifiers, so naming in form or *repository:tag* provides a human-readable way of identifying images. For more information, see [Section 10.5, “Using Container Tags”](#) and [Section 10.2, “Managing Repositories”](#).

With Red Hat Satellite, you can create an on-premise registry, import images from various sources and distribute them to containers using content views (see [Section 6.2, “Adding Repositories to the Content View”](#) for more information on loading images to a content view). Satellite supports creating one or more Docker compute resources that act as servers for running containers. This way, you can import an image, start a container based on this image, monitor the container's activity, and commit its state to a new image layer that can be further propagated.

10.1. Managing Containers

The following sections show how to create, view, start, stop, and commit a container.

Prerequisites

In Red Hat Satellite, you can deploy containers only on a compute resource of the Docker provider type. Therefore, when you attempt to view or create containers for the first time,

Satellite prompts you to create a Docker compute resource. To do so, first create a container host as described in [Procedure 10.1, “To Prepare a Container Host:”](#), then specify this host as a compute resource as described in [Procedure 10.2, “To Create a Docker Compute Resource:”](#).

Procedure 10.1. To Prepare a Container Host:

1. Prepare a Red Hat Enterprise Linux 7 server for hosting images and enable the **docker** service on this server as described in the *Getting Docker in RHEL 7* section of the [Get Started with Docker Formatted Container Images on Red Hat Systems](#) guide on the Red Hat Customer Portal ^[2]. You can deploy the container host either on the same machine as the Satellite server or independently.



Note

Red Hat Enterprise Linux 7 is currently the only supported system for a container host. The *docker* package is available in the *rhel-7-server-extras-rpms* repository. Red Hat Enterprise Linux 6 systems are currently not supported to host containers.

2. Run the following command on the container host to install the Satellite server's CA certificate:

```
rpm -Uvh https://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

Here, *satellite.example.com* is the fully qualified domain name of your Satellite server. Skip this step if the container host is already registered as a Satellite host.

3. Depending on the location of the container host, perform the following tasks:

A. If the container host is on the same machine as the Satellite server:

- a. Create a docker user group and add the foreman user to it:

```
# groupadd docker
# usermod -aG docker foreman
```

- b. Modify the **OPTIONS** variable in the **/etc/sysconfig/docker** file as follows:

```
OPTIONS='--selinux-enabled -G docker'
```

- c. Restart the affected services to apply the changes:

```
# systemctl restart docker.service
# katello-service restart
```

B. If the container host is on a different machine than the Satellite server:

- a. Open a port on the container host to communicate with the Satellite server. To do so, modify the **OPTIONS** variable in the **/etc/sysconfig/docker** file as follows:

```
OPTIONS='--selinux-enabled -H tcp://0.0.0.0:port_number -H
unix:///var/run/docker.sock'
```

Replace *port_number* with a selected port number.

- b. Restart the docker service and verify your settings as follows:

```
# systemctl restart docker.service
# systemctl status docker.service
```

Procedure 10.2. To Create a Docker Compute Resource:

1. Make sure the port 5000 is enabled on the Satellite server. The container host uses this port to pull images from Content Views on the Satellite server.
2. Create the compute resource as described in [Procedure 13.9, “To Add a Compute Resource:”](#). Specify the resource **URL** according to the location of the container host:
 - a. If the container host is on the same machine as the Satellite server, set *unix:///var/run/docker.sock* as the resource URL.
 - b. If the container host is on a different machine than the Satellite server, specify the URL in the form of:

```
http://container_host_fqdn:port_number
```

Here, *container_host_fqdn* and *port_number* stand for the fully qualified domain name of the container host and the port number opened on the container host for communication with Satellite.

3. Click **Test Connection** to test if the container host is available.
4. Click **Submit** to create the compute resource.

10.1.1. Creating Containers

When there is at least one Docker compute resource present in your Satellite, you can create containers. To create a new container, follow the steps described in [Procedure 10.3, “To Create a Container:”](#). For instructions on how to monitor existing containers, see [Section 10.1.2, “Monitoring Containers”](#).

To create a container, you must first import an image, which can be a platform image or a previously created layered image. Satellite supports the following image sources:

- ✦ **Local content:** represented by the **Katello** option when creating a container. This option allows you to import an image from a repository that is already present on a capsule server in a certain content view and life cycle environment. For more information on how to create and populate a local registry, see [Section 10.2, “Managing Repositories”](#).
- ✦ **Docker Hub:** allows you to search the Docker Hub registry and pull images from there. Make sure that you pull only trusted images with verified content.
- ✦ **External Registry:** allows you to import images from a previously created external registry. For more information on creating registries in Red Hat Satellite, see [Section 10.3, “Importing External Registries”](#).



Note

You cannot change the configuration of an existing container. To alter the configuration, you have to create a replacement container with modified settings as described in [Procedure 10.3, “To Create a Container:”](#). Therefore, make sure that containers can be easily replaced in your workflow.

Procedure 10.3. To Create a Container:

1. Navigate to **Containers → New Container**. Alternatively, navigate to **Containers → All Containers** and click **New container**.
2. In the **PreLiminary** stage of container creation, configure the following settings:
 - ✳ On the **Compute resource** tab, select the compute resource from the **Deployed on** drop-down menu. For more information on compute resources, see [Section 13.3.4, “Compute Resources”](#).
 - ✳ On the **Locations** tab, select the locations where the new container will be available.
 - ✳ On the **Organizations** tab, select the organizations where the new container will be available.

Click **Next** to proceed.

3. In the **Image** stage of container creation, import an image that will act as a base for your container. This can be a platform image, or a previously created layered image. Select from one of the following options:
 - ✳ Select the **Katello** tab to import the image from a life cycle environment. Specify the life cycle environment, content view, repository, tag, and Capsule Server.
 - ✳ Select the **Docker hub** tab to import the image from the Docker Hub registry. After you type the image name to the **Search** field, Satellite automatically searches the compute resource. Click the looking glass icon to search the Docker Hub. Select the image from the list of search results and pick a tag from the drop-down list.
 - ✳ Select the **External registry** tab to import the image from an existing registry. Select the registry from the drop-down menu, and search it by the image name. Satellite populates the **Tag** field with tags available for the selected image name. For more information, see [Section 10.3, “Importing External Registries”](#).

Click **Next** to proceed.

4. In the **Configuration** stage of container creation, set the following parameters:
 - ✳ Provide the container name.
 - ✳ Specify a command to run inside the container.
 - ✳ Specify an entrypoint, which is a command that is executed automatically as soon as the container starts. The default entrypoint is `/bin/sh -c`.
 - ✳ Assign CPUs to the container. For example, **0-2,16** represents CPUs 0, 1, 2, and 16.
 - ✳ Define the relative share of CPU time for the container.

- Specify a memory limit for the container. For example, **512m** limits the container memory usage to 512 MB.

Click **Next** to proceed.

5. In the final stage of container creation named **Environment**, select if you want to allocate a pseudo-tty, attach STDIN, STDOUT, and STDERR to the container. Click **Add environment variable** to create a custom environment variable for the container.
6. Click **Submit** to create the container.

After creating a container, Satellite displays a summary of container metadata. By default, new containers are disabled. For instructions how to start containers see [Procedure 10.5, “To Start or Stop a Container:”](#).

Example 10.1. Creating a Red Hat Enterprise Linux Container in Satellite

To enable a Red Hat Enterprise Linux container in Red Hat Satellite, perform the following actions:

1. Create a custom registry as described in [Section 10.3, “Importing External Registries”](#). Specify `https://registry.access.redhat.com` as the registry URL.
2. Create a new container as described in [Section 10.1.1, “Creating Containers”](#). In the **Image** stage of container creation, navigate to the **External registry** tab and select the registry created in the previous step. Use the search field to find the desired version of the Red Hat Enterprise Linux image. Proceed through the **Configuration** and **Environment** stages to finalize the container.

10.1.2. Monitoring Containers

Red Hat Satellite provides the means to monitor the status of containers as well as processes running inside them. Some containers can be marked as *managed*, which means they were created and provisioned inside the Satellite environment.

The following procedure shows how to list containers of a selected organization and how to monitor the container metadata.

Procedure 10.4. To Investigate a Container:

1. Navigate to **Containers → All Containers**.
2. On the **Containers** page, every Docker compute resource has a dedicated tab. Each of these tabs contains the table of available containers together with selected parameters of each container. Select the tab of the compute resource you want to inspect.
3. To view the container metadata, click the name of the container you want to inspect. Satellite displays the table of container properties.
4. On the **Processes** tab, you can view processes that are currently running in the container. Click on the process name to view the metadata of the process.
5. If the container is running, you can view its standard output in the **Logs** tab. If you selected the **allocate a pseudo-tty** check box when creating a container, the console is interactive. Otherwise, it displays the initial standard output produced when the container started.

10.1.3. Starting, Committing, and Removing Containers

New containers are by default disabled. By enabling a container, you start the processes of the containerized application in the compute resource. Hosts are then able to communicate with the container as with a web application. The following procedure shows how to start and stop a container:

Procedure 10.5. To Start or Stop a Container:

1. Navigate to **Containers** → **All Containers** to view the list of available containers.
2. Click **Power On** next to the container you want to start. After starting the container, the button changes to **Power Off**, which allows for stopping the container. These actions are equivalent to the **docker start** and **docker stop** commands.

The following procedure shows how to commit a container to create a new image layer that stores the status of the container.

Procedure 10.6. To Commit a Container:

1. Navigate to **Containers** → **All Containers** to view the list of available containers.
2. Click the name of the container you want to commit.
3. Click **Commit**. Satellite prompts you to:
 - ✧ Specify a repository name. This can be a single name or combined with the user name, for example *user/my-rhel-image*.
 - ✧ Assign a tag to the image.
 - ✧ Provide your contact information.
 - ✧ Provide an informative comment about the image.
4. Click **Submit**.



Note

The container is committed to the repository of the original image. For example, if the container is based on an image pulled from the Docker Hub, the committed changes are pushed back to the Docker Hub.

Procedure 10.7. To Remove a Container:

1. Navigate to **Containers** → **All Containers** to view the list of available containers.
2. Click the name of the container you want to delete.
3. Click **Delete**.
4. In the alert box, click **OK** to remove the container.

10.2. Managing Repositories

This section shows how to create an internal repository for container images. You can use internal repositories to create containers as described in [Section 10.1.1, “Creating Containers”](#).

10.2.1. Creating Repositories

Repositories provide a way to synchronize the container content either from the Red Hat Content Delivery Network or from other sources. To create a repository for container images in a product, follow the steps described in the [Adding Repositories to a Product](#) section in the [Red Hat Satellite Installation Guide](#). Select **docker** as a repository type. For instructions on how to create a product, see the [Creating a Product](#) section in the same guide.

10.2.2. Uploading Images to Repositories

The following procedure shows how to import images to an existing repository.

Procedure 10.8. To Upload Images to a Repository:

1. Navigate to **Content** → **Products**.
2. Select the product that contains the repository you want to update. Navigate to the **Repositories** tab and select the docker repository you want to update.
3. Click **Browse**. Navigate to the location of the image you want to upload. Click **Open**.
4. Click **Upload** to upload the image to the repository.

10.3. Importing External Registries

The following procedure shows how to import an external registry to the Satellite server.

Procedure 10.9. To Import an External Registry:

1. Navigate to **Containers** → **Registries**. Click **New Registry**.
2. On the **Registry** tab, specify the name and URL of the registry. These settings are required. Optionally, provide a brief description of the registry. Specify a user name and password if required for accessing the registry.
3. On the **Locations** tab, select the locations where the new registry will be available.
4. On the **Organizations** tab, select the organizations where the new registry will be available.
5. Click **Submit** to create the registry.

10.4. Importing Images to Compute Resources

Importing an image is a necessary step when creating a container. You can also import images to a compute resource before creating a container as described in the following procedure.

Procedure 10.10. To Import an Image to a Compute Resource:

1. Navigate to **Infrastructure** → **Compute resources** to view a list of compute resources.
2. Select the docker compute resource you want to edit.

3. Click **New image**.
4. Specify the image details including the image name, operating system, architecture, user credentials, and a parent image. Select **User data** to enable user input for this image.
5. Click **Submit**.

10.5. Using Container Tags

Tags are convenient for organizing images, especially when you operate with several versions of a containerized application. The following procedure shows how to use tags to search for images:

Procedure 10.11. To Search Registries by Tags:

1. Navigate to **Content** → **Docker tags**.
2. Use the search field to filter tags by the image name, tag, or repository name. Automatic suggestion works as you type. For example, the following query searches for tags applied on images from the repository named `test_repo`:

```
repository = test_repo
```

3. Click the name of the tag you want to view. Satellite displays a list of images that use this tag.
4. Select an image to view its environment and content view version. The **Published At** field shows the URL that you can use to pull the image from the command line.

For the list of alternative comparison operators, see [Table 17.2, “Supported Operators for Granular Search”](#). By default, the search field recognizes the input string as a tag name. For example, type **centos** to search for all centos tags.

[1] <https://access.redhat.com/articles/881893>

[2] <https://access.redhat.com/articles/881893#get>

Chapter 11. Configuring Activation Keys

Activation Keys improve the speed, simplicity and consistency of content host registration. Activation keys can define the following properties of a content host:

- ✦ Which life cycle environment the content host should be placed in.
- ✦ Which host collection the content host should be assigned to.
- ✦ Which organization the content host should be a part of.
- ✦ Whether to use a provisioning template for the content host.
- ✦ Setting up a subscription usage limit for the content host.
- ✦ Assigning a specific subscription to the content host.

An activation key ensures that the content host is associated with the correct host group and that the content host consumes the correct subscription. Activation keys store a set of subscriptions that can be associated with content hosts without attaching these subscriptions directly. This results in several benefits for subscription management:

- ✦ Administrators have control over which subscriptions are installed to a content host without having to create and configure every content host first.
- ✦ Because activation keys are created within the Satellite Server and do not rely on content host settings or architecture, the target content host does not have to exist yet.
- ✦ Users can register their content host in a single step and automatically have all required subscriptions attached, without having to select and attach subscriptions manually and potentially miss a subscription.

The same activation key can be applied to multiple content hosts, as long as it contains enough subscriptions. However, activation keys only set the initial configuration for a content host. When it is registered to an organization, other content which that organization possesses can be attached to the content host manually.

Activation keys are defined in an organizational context meaning that an organization must be selected to access activation keys in the Satellite GUI.

11.1. Creating an Activation Key

This section describes how to create an activation key.

Procedure 11.1. To Create an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click **New Activation Key**.
3. Enter the required details for the activation key in the relevant fields.
4. Clear the **Unlimited** check box if the activation key is to be used with limitations. Type the usage limit in the **Limit** field. You can use this field to control how many times a given Activation Key is used. For example, if you associate the key with a subscription that has a limited quantity, you can set the limit on the Activation Key to eliminate exceeding that quantity.

5. Enter a suitable description in the **Description** field.
6. Select the **Environment** and **Content View** that this key should apply to.
7. Click **Save** to create the activation key.



Note

You can change the activation key details in the different tabs of the Activation Key.

11.2. Removing an Activation Key

This section describes how to remove an activation key.

Procedure 11.2. To Remove an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key name that you want to remove.
3. In the upper right of the **Activation Key** details panel, click **Remove**.
4. In the alert box, click **Remove** to confirm that you want to remove the key.

11.3. Editing Activation Keys

This section describes how to edit activation keys.

11.3.1. Using Auto-Attach with an Activation Key

The auto-attach setting of an activation key determines what subscriptions are automatically attached during registration. If auto-attach is enabled, the activation key will only consume the minimum Red Hat products required to make the subscription valid.



Note

Auto-attach is enabled by default when an activation key is created. If auto-attach is disabled, every possible subscription on the key that can be attached will be attached on registration.

Procedure 11.3. To Edit Auto-Attach on an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key name that you want to edit.
3. Click the **Subscriptions** tab.
4. Select the edit box next to **Auto-Attach**.
5. Select the check box to enable auto-attach.

6. Click **Save**.

11.3.2. Setting a Service Level for an Activation Key

An activation key can be configured to define a default service level for the new host created with the activation key. Setting a default service level will select only the matching subscriptions to be attached to the host. For example, if the default service level on an activation key is set to Premium, only subscriptions with premium service levels will be attached to the host upon registration.

Procedure 11.4. To Set the Service Level on an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key name you want to edit.
3. Click the **Details** tab.
4. Select the edit box next to **Service Level**.
5. Select the required service level from the drop-down list. The drop-down list only contains service levels available to the activation key.
6. Click **Save**.

11.3.3. Adding Subscriptions to an Activation Key

This section describes how to add subscriptions to an activation key. Subscriptions can be associated with activation keys so that hosts utilizing the activation keys will be automatically attached to the associated subscriptions upon registering with the Satellite Server.

Procedure 11.5. To Add a Subscription to an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the name of the activation key you want to edit.
3. Click **Subscriptions** → **Add**.
4. From the list of available subscriptions, select the subscriptions you want to add and then click **Add Selected**.

11.3.4. Adding Host Collections to an Activation Key

These steps show how to add host collections to an activation key. Host collections can be associated with activation keys so that hosts utilizing the activation keys will automatically be added to the associated host collections upon registering with the Satellite Server.

Procedure 11.6. To Add Host Collections to an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key that you want to add a host collection to.
3. Click **Host Collections** and then click **Add** to display the list of available host collections.
4. Select the host collections you want to add, and then click **Add Selected** to add the

host collections to the activation key.



Note

After you have added the host collections to the activation key, they no longer appear in the list of available collections. To view the host collections that have been added to an activation key, click **List/Remove**.

11.3.5. Editing Product Content on an Activation Key

The default value of products can be edited on the Product Content tab. This tab shows the repository content sets that are currently available to an activation key. The default setting for product content will determine if a product is enabled at registration. The default setting varies on a product by product basis with the exception of custom products which default to Yes.



Note

Changing default settings for content hosts that register with an activation key requires subscription-manager version 1.10 or later to be installed on that host.

Procedure 11.7. To Edit Product Content on an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key name that you want to edit.
3. Click the **Product Content** tab.
4. Click the edit box next to the required product.
5. Select either **Yes (Default)** or **Override to No**.
6. Click **Save**.

11.3.6. Removing Subscriptions from an Activation Key

These steps show how to remove subscriptions from an activation key.

Procedure 11.8. To Remove Subscriptions from an Activation Key:

1. Click **Content** → **Activation Keys**.
2. A list of activation keys is displayed. Click the activation key you want to remove subscriptions from.
3. Click the **Subscriptions** subtab.
4. A list of subscriptions is displayed. Select the subscriptions to be removed.
5. Click **Remove Selected** to remove subscriptions from the activation key.

11.3.7. Removing Host Collections from an Activation Key

These steps show how to remove host collections from an activation key.

Procedure 11.9. To Remove Host Collections from the Activation Key:

1. Click **Content** → **Activation Keys**.
2. A list of activation keys is displayed. Click the activation key you want to remove host collections from.
3. Click the **Host Collections** subtab.
4. A list of host collections attached to the Activation Key is displayed. Select the check box of the host collections you want to remove.
5. Click **Remove Selected** to remove host collections from the activation key.

11.4. Automated Host Registration with Activation Keys

The following steps show how to automatically register a host using an activation key. When the activation key has been created, you can apply it by using the *subscription-manager* utility during host registration on the Satellite Server. Prepare the host as described in [Chapter 14, Configuring Hosts](#), then follow the steps outlined in [Procedure 11.10, “To Automatically Register a Host with an Activation Key:”](#).

Procedure 11.10. To Automatically Register a Host with an Activation Key:

1. Clear any old registration data from the system.

```
[root@server]# subscription-manager clean
```

2. Register the system to the required organization on the Satellite Server. Use the *--activationkey* flag to register the system using the activation key. Enter the authentication for the admin user when prompted.

```
[root@server]# subscription-manager register --org  
'Default_Organization' --activationkey 'Test_Key'
```

3. When the system is registered, it gains access to repository content but administrators will not be able perform package and errata management until the katello agent has been installed on the client system.

```
[root@server]# yum install katello-agent
```

Chapter 12. Configuring GPG Keys

GPG keys allow you to add your existing GPG keys to Red Hat Satellite Server products and repositories to enable pairing with your repositories.

12.1. Creating a GPG Key

This section describes how to add a GPG key to Red Hat Satellite.

Procedure 12.1. To Add a GPG Key to Satellite:

1. Click **Content** → **GPG Keys** and then click **New GPG Key**.
2. Enter a name for the GPG key in the **Name** field.
3. Either upload the GPG key file or paste the GPG key contents into the text box.
4. Click **Save** to add the GPG key to Satellite.

12.2. Removing a GPG Key

This section describes how to remove a GPG from Red Hat Satellite.

Procedure 12.2. To Remove a GPG Key:

1. Click **Content** → **GPG Keys**.
2. Click the GPG key that you want to remove, and then click **Remove GPG Key**.
3. In the confirmation box, click **Remove** to confirm that you want to remove the selected key.

Chapter 13. Configuring the Provisioning Environment

13.1. Creating a Host Group

A host group defines a set of default values that hosts inherit when placed in that group. Hosts can belong to only one host group, but host groups can be nested in hierarchies. You can create a "base" or "parent" host group that represents all hosts in your organization, and then create nested or "child" host groups under that parent to provide specific settings. This section describes how to create a host group.

Procedure 13.1. To Add a Host Group to Satellite:

1. Click **Configure** → **Host Groups** and then click **New Host Group**.
2. Enter the required details for the Host Group, and then click **Submit**.

Host Group Attributes

The following table describes the attributes that apply to Satellite Host Groups.

Table 13.1. Table of Host Group Attributes

Submenu	Options	Description
Host Group	Parent	The parent Host Group for the new Host Group.
	Name	The name of the Host Group.
	Life Cycle Environment	The environment containing this Host Group.
	Puppet CA	The Red Hat Satellite Capsule Server to use for the Puppet CA server.
	Puppet Master	The Red Hat Satellite Capsule Server to use as the Puppet Master.
Puppet Classes	Included Classes	The Puppet Classes included with the Host Group.
	Available Classes	The Puppet Classes available to use with the Host Group.
Network	Domain	The domain for hosts in the Host Group.
	Subnet	The subnet for hosts in the Host Group.
Operating System	Architecture	The default architecture for systems in the Host Group.
	Operating Systems	The default operating system for systems in the Host Group.
	Media	The location of the installation media for the operating system.
	Partition Table	A file system partition layout for the operating system installation.
	Root Password	The root password for the operating system.
Parameters	Add Parameter	Provides a Name and Value pair to set parameters for the Host Group.
Organizations	Organizations	The organizations that own this host group.
Activation Keys	Content	Defines the activation keys made available in templates as <code>@host.params['kt_activation_keys']</code> .
	Environment	

13.2. Parameters

Red Hat Satellite parameters define key-value pairs to use when provisioning hosts. These are similar to Puppet's concept of a default scope parameter. You can define parameters when setting up a host with Puppet and also define a hierarchy of parameter inheritance.

The following parameter hierarchy applies:

Global Parameters

The default parameter that applies to every host in Satellite. Configured in **Configure** → **Global parameters**.

Domain Parameters

Parameters that affect all hosts in a given domain. Domain parameters override Global parameters. Configured in **Infrastructure** → **Domains**.

Host Group Parameters

Parameters that affect all hosts in the Host Group. Host Group parameters override both Global and Domain parameters. Configured in **Configure** → **Host Groups**.

Host Parameters

Parameters that affect a specific host. All previous inherited parameters are visible on the Parameters subtab and can be overridden. Configured in **Hosts** → **All hosts** → **[choose_a_host]** → **Parameters** or **Hosts** → **Content Hosts** → **[choose_a_host]** → **Parameters**.

Different types of parameters also exist:

Simple Parameters

A basic parameter that defines a relationship between a key and value pair.

Smart Parameters

A complex parameter that defines a value for a key but allows conditional arguments, validation, and overrides for specific object types.

Parameterized Classes

Parameters for classes imported from a Puppet Master.



Important

Ensure you enable parameterized class support. Navigate to **Administer** → **Settings**, select the **Puppet** tab, and ensure the **Parameterized_Classes_in_ENC** is set to **True**.

13.2.1. Creating a Global Simple Parameter

This procedure shows how to add a new global parameter to Satellite.

Procedure 13.2. To Create a Global Simple Parameter:

1. Click **Configure** → **Global Parameters**.

2. Click **New Parameter**.
3. Type a **Name** for the parameter's key.
4. Type a **Value** for the parameter.
5. Select if the value is hidden.
6. Click **Submit**.

13.2.2. Configuring Smart Parameters

The following procedure configures smart parameters in a Puppet class.

Procedure 13.3. To configure smart parameters:

1. Click **Configure → Puppet Classes**.
2. Select a class from the list.
3. Click the **Smart Variables** tab. This displays a new screen. The left section contains a list of possible parameters the class supports. The right section contains the configuration options for the parameter selected. Click the **Add Variable** to add a new parameter. Otherwise, select a parameter from the left-hand list.
4. Type a name for the **Parameter** field.
5. Edit the **Description** text box to add any plain text notes.
6. Select the **Parameter type** of data to pass. This is most commonly a string, but other data types are supported.
7. Type a **Default Value** for the parameter.
8. Use the **Optional Input Validator** section to restrict the allowed values for the parameter. Choose a **Validator type** (either a **list** of comma separated values or a regular expression, **regex**) and input the allows values or regular expression code in the **Validator rule** field.
9. The **Override Value For Specific Hosts** section at the bottom of the page provides options for overriding values based upon conditional arguments known as **Matchers**. Define the **Order** that the host values resolve, then click **Add Matcher-Value** to add your conditional argument.

For example, if desired value of the parameter is **test** for any host with a fully qualified domain name of **www.example.com**, then specify the **Match** as **fqdn=www.example.com** and the **Value** as **test**.

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

13.2.3. Importing Parameterized Classes from a Puppet Master

The following procedure imports parameterized classes from your Puppet Master.



Note

The import of parameterized classes happens automatically if your puppet modules are managed via a Product and a Content View.

Procedure 13.4. To Import Parameterized Classes:

1. Click **Configure** → **Puppet Classes**.
2. Click **Import** from *Host Name* to import parameterized classes from your Puppet Master.
3. The **Puppet Classes** page displays with the new classes listed.

13.2.4. Configuring Parameterized Classes

The following procedure configures parameterized classes.

Procedure 13.5. To Configure Parameterized Classes:

1. Click **Configure** → **Puppet Classes**.
2. Select a class from the list.
3. Click the **Smart Class Parameter** tab. This displays a new screen. The left section contains a list of possible parameters the class supports. The right section contains the configuration options for the parameter selected.
4. Select a parameter from the left-hand list.
5. Edit the **Description** text box to add any plain text notes.
6. Select **Override** to allow Satellite control over this variable. If the check box is not selected, Satellite does not pass this variable to Puppet.
7. Select the **Parameter type** of data to pass. This is most commonly a string, but other data types are supported.
8. Type a **Default Value** for the parameter.
9. The **Override Value For Specific Hosts** section at the bottom of the page provides options for overriding values based upon conditional arguments known as **Matchers**. Define the **Order** that the host values resolve, then click **Add Matcher-Value** to add your conditional argument.

For example, if desired value of the parameter is **test** for any host with a fully qualified domain name of **www.example.com**, then specify the **Match** as **fqdn=www.example.com** and the **Value** as **test**.

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

13.3. Configuring Provisioning Settings

This section shows how to create and configure elements of a provisioning environment.

13.3.1. Domains

Satellite has the ability to assign domain names with Red Hat Satellite Capsule Server DNS. This provides users with a means to group and name hosts within a particular domain.

Procedure 13.6. To Create a Domain:

1. Click **Infrastructure** → **Domains**.
2. Click **New Domain**. On the **Domain** tab, specify the following settings:
 - a. Specify a **Name** for the Domain. This is the required DNS domain name.
 - b. Type a **Description** for the Domain.
 - c. Select a DNS-enabled Capsule server.
3. On the **Parameters** tab, specify domain parameters.
4. On the **Locations** tab, select locations for the domain.
5. On the **Organizations** tab, select organizations for the domain.



Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

6. Click **Submit**.

13.3.2. Subnets

Satellite has the ability to create networks for groups of systems. Subnets use standard IP address settings to define the network and use the Red Hat Satellite Capsule Server's DHCP features to assign IP addresses to systems within the subnet.

13.3.2.1. Creating a Subnet

The following procedure shows how to create a subnet.

Procedure 13.7. To Create a Subnet:

1. Click **Infrastructure** → **Subnets**.
2. Click **New Subnet**. On the **Subnet** tab, specify the following settings:
 - a. Specify a **Name**, **Network address** (IP address), and **Network mask** for the subnet. These settings are required.
 - b. Optionally, specify the **Gateway address**, **Primary DNS server**, **Secondary DNS server**, and **VLAN ID**. You can also select the **IPAM** mode (DHCP, Internal DB, or None) and define the IP assignment range with the **Start of IP range** and **End of IP range** fields.
 - c. Select the default **Boot mode** for the subnet (DHCP or Static).

3. On the **Domains** tab, select the applicable domains for the subnet.
4. On the **Capsules** tab, select the Capsule servers to be used for hosting the **DHCP Proxy**, **TFTP Proxy**, **DNS Proxy**, and **Discovery Proxy** services.
5. On the **Locations** tab, select locations for the subnet.
6. On the **Organizations** tab, select organizations for the subnet.



Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

7. Click **Submit**.

13.3.3. Architectures

An architecture in Satellite represents a logical grouping of hosts and operating systems. Architectures are created by Satellite automatically when hosts check in with Puppet. However, none exist with a default installation and require creation.

Procedure 13.8. To Create an Architecture:

1. Click **Hosts** → **Architectures** and then click **New Architecture**.
2. Specify a **Name** for the architecture.
3. Select any **Operating Systems** that include this architecture. If none are available, you can create and assign them under **Hosts** → **Operating Systems**.
4. Click **Submit**.

13.3.4. Compute Resources

Compute resources are hardware abstractions from virtualization and cloud providers. Satellite uses compute resources to provision virtual machines and containers. Supported private providers include Red Hat Enterprise Virtualization, oVirt, OpenStack, VMware, Libvirt, and Docker. Supported public cloud providers include Amazon EC2, Google Compute Engine, and Rackspace.

Procedure 13.9. To Add a Compute Resource:

1. Navigate to **Infrastructure** → **Compute Resources**.
2. Click **New Compute Resource**. On the **Compute Resource** tab, specify the following settings:
 - a. Specify a **Name** and a **Provider** type for the Compute Resource. Optionally, insert a **Description**.
 - b. Depending on the provider type chosen, the next few fields ask for authentication and datacenter details. Refer to the following table for more information about each provider type.

Table 13.2. Provider Settings

Type	Description
RHEV	Suits Red Hat Enterprise Virtualization environments. Requires the URL of the Manager API, a valid Username and Password , and a Datacenter on the system to abstract compute resources. Click Load Datacenters to populate the drop-down menu. Optionally, you can specify a Quota ID and provide one or more certificate authorities in the X509 Certification Authorities field.
Libvirt	Suits Libvirt-based environments. Requires the URL of the virtual machine. Select the Display type . Click Test Connection to test if the virtual machine is available. Select Console passwords to set a randomly generated password on the display connection.
VMware	Suits VMware-based environments. Requires the hostname of the VCenter/Server , a valid VMware Username and Password , and a Datacenter to abstract compute resources. Click Load Datacenters to populate the drop-down menu. You can specify a certificate Fingerprint and select Console passwords to set a randomly generated password on the display connection.
RHEL OpenStack Platform	Suits OpenStack-based environments. Requires the URL of the OpenStack server, a valid OpenStack Username and Password , and a Tenant to abstract compute resources. Click Load Tenants to populate the drop-down menu.
Rackspace	Suits Rackspace public cloud accounts. Requires the URL of the Rackspace API, a valid Rackspace Username and API Key , and a Region to abstract compute resources. Click Test Connection to make sure your connection to the chosen region is valid.
EC2	Suits Amazon EC2 public cloud accounts. Requires the Access Key and Secret Key available from any valid Amazon EC2 account. Requires a Region to act as a Datacenter for resource abstraction. Click Load Regions to populate the selection drop-down menu.
Google	Suits Google Compute Engine public cloud accounts. Requires the Google Project ID , a valid Client Email and a Certificate path to the p12 file. You can also specify a zone to abstract compute resources. Click Load zones to populate the drop-down menu.
Docker	Suits container registries. Requires the URL of the internal or external compute resource. Optionally, specify a Username , Password , and a contact Email . Click Test Connection to test if the connection is available.

3. On the **Locations** tab, select desired locations to add them to the **Selected Items** list.
4. On the **Organizations** tab, select the desired organizations to add them to the **Selected Items** list.



Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

5. Click **Submit**.

13.3.5. Hardware Models

Hardware models help run unattended Solaris installations. For Solaris SPARC-based machines, users define the CPU and Vendor information, while other architectures do not need to do so.

Procedure 13.10. To Create a Hardware Model:

1. Click **Hosts** → **Hardware Models**.
2. Click **New Model**.
3. Specify a **Name** for the Hardware Model.
4. For Sparc Solaris builds, insert the CPU **Hardware model** and **Vendor class**. Other architectures do not require values in these fields.
5. Type a description of the Hardware Model in the **Information** field.
6. Click **Submit**.

13.3.6. Installation Media

Red Hat Satellite uses installation media (ISO images) as content for kickstart trees and new host installations.

Procedure 13.11. To Add an Installation Medium:

1. Click **Hosts** → **Installation Media**.
2. Click **New Medium**. On the **Medium** tab, specify the following settings:
 - a. Type a **Name** for the Installation Media. This setting is required.
 - b. Type a **Path** to the Installation Medium. Options include either a URL or a valid NFS server. This setting is required.
 - c. Select an **Operating System Family** to define the type of the Installation Medium.
3. On the **Locations** tab, select the desired locations to add them to the **Selected Items** list.
4. On the **Organizations** tab, select the desired organizations to add them to the **Selected Items** list.



Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

5. Click **Submit**.

13.3.7. Partition Tables

Partition tables define the partitions and file system layout for new installations when provisioning systems. Satellite users specify the host's disk layout as an explicit sequence of partitions or use a dynamic disk layout script.

Procedure 13.12. To Create a Partition Table:

1. Click **Hosts** → **Partition Tables**.
2. Click **New Partition Table**.
3. Type a **Name** for the partition table.
4. Specify the **Layout** of the partition table. The **Layout** field also accepts dynamic disk partitioning scripts.
5. Select the operating system from the **OS Family** drop-down list.
6. Click **Submit**.

New partition table has to be associated with an operating system as described in [Section 13.3.10, "Operating Systems"](#)

13.3.8. Provisioning Templates

Provisioning templates provide the systematic means to run unattended installations. Provisioning templates can be executed via several methods including bash scripts, kickstart scripts, and PXE-based installations.

Procedure 13.13. To Create a Provisioning Template:

1. Click **Hosts** → **Provisioning Templates**.
2. Click **New Template**. On the **Provisioning Template** tab, specify the following settings:
 - a. Specify a **Name** for the template.
 - b. Insert your template in the **Template editor** field. Alternatively, click **Browse** to upload the template. This replaces the content in the **Template editor** field with the content of your chosen file.
 - c. Optionally, type a comment in the **Audit Comment** field. Satellite adds the comment to the template history to track changes. View the template history under the **History** tab.
3. On the **Type** tab, select **Snippet** to store the template code without defining it as particular script or template type, or select the type from the **Type** drop-down menu.
4. On the **Association** tab, select host groups, environments and operating systems to be associated with the template. Select the operating systems from the **Applicable Operating Systems** list. Click **Add Combination** and select a **Hostgroup** and **Environment** to limit the template's use. Note that associations are not available for templates of type snippet.
5. On the **Association** tab, you can view the history of existing templates. No history is available when creating a new template.
6. On the **Locations** tab, select locations for the template.
7. On the **Organizations** tab, select organizations for the template.

**Important**

Ensure that the Locations and Organizations are configured as they will help with future debugging.

8. Click **Submit**.

13.3.9. Configuring gPXE to Reduce Provisioning Times

To reduce provisioning time when downloading PXE boot files, pPXE enables the use of additional protocols such as **HTTP** to reduce download time. To make use of gPXE, proceed as follows:

- On systems configured to be a **TFTP** server, copy `/usr/share/syslinux/gpxelinuxk.0` to `/var/lib/tftpboot`.
- In the **PXE Handoff** section of `/etc/dhcp/dhcpd.conf`, change the **DHCP filename** option from `pxelinux.0` to `gpxelinuxk.0`.
- Create provisioning templates as follows and then assign them, together with the default template, to the operating systems.

Procedure 13.14. Configure a gPXE Provisioning Template

1. Click **Hosts** → **Provisioning templates**.
2. Find the template **Kickstart default PXELinux** and select **Clone**.
3. Enter a name, for example, **Kickstart default gPXELinux**.
4. In the Template editor, search and replace `@initrd` with `@host.url_for_boot(:initrd)`
5. In the Template editor, search and replace `@kernel` with `@host.url_for_boot(:kernel)`
6. Select the **Type** tab. From the **Type** drop-down menu, select **PXELinux**.
7. On the **Association** tab, select host groups, environments and operating systems to be associated with the template. Select the operating systems from the **Applicable Operating Systems** list. Click **Add Combination** and select a **Hostgroup** and **Environment** to limit the template's use.
8. Click **Submit**.

13.3.10. Operating Systems

Operating Systems define combinations of installation methods and media and are grouped within families. As a default, Red Hat Satellite uses a **RedHat** family. Families allow Satellite to change certain behaviors when provisioning hosts.

Procedure 13.15. To Add an Operating System:

1. Click **Hosts** → **Operating Systems**.

2. Click **New Operating system**. On the **Operating System** tab, specify the following settings:
 - a. Type the **Name** of the Operating System and its **Major Version**. These settings are required.
 - b. Optionally, define the **Minor Version**, select the **OS Family**, and add a **Description** of the operating system.
 - c. Select a **Root password hash** (MD5, SHA256, or SHA512).
 - d. Select the **Architectures** from the list of available Architectures. If none are available, create and assign them under **Hosts** → **Architectures** as described in [Section 13.3.3, “Architectures”](#).
3. On the **Partition tables** tab, select the applicable file system layouts from the list. For more information on creating partition tables, see [Section 13.3.7, “Partition Tables”](#).
4. On the **Installation Media** tab, select the applicable installation media from the list. For more information on adding installation media, see [Section 13.3.6, “Installation Media”](#).
5. On the **Templates** tab, you can assign provisioning templates when editing an existing operating system. This option is not available when creating a new operating system. For more information on creating provisioning templates, see [Section 13.3.8, “Provisioning Templates”](#).
6. On the **Parameters** tab, you can add parameters for the operating system.
7. Click **Submit**.

13.4. Storing and Maintaining Host Information

Red Hat Satellite 6 uses a combination of applications to gather information about managed hosts and to ensure that those hosts are maintained in the desired state. These applications include:

- ✦ **Foreman**: Provides for the provisioning and life cycle management of physical and virtual systems. Foreman automatically configures these systems using various methods, including kickstart and Puppet modules.
- ✦ **Puppet**: A client/server architecture for configuring hosts, consisting of the Puppet Master (server) and the Puppet Agent (client).
- ✦ **Facter**: Puppet's system inventory tool. Facter gathers basic information (facts) about hosts such as hardware details, network settings, OS type and version, IP addresses, MAC addresses, SSH keys, and more. These facts are then made available in Puppet manifests as variables.

The use of Puppet, Facter, and facts is discussed in more detail below.

13.4.1. The Puppet Architecture

Puppet usually runs in an agent/master (also known as a client/server) architecture, where a Puppet server controls important configuration information, and managed hosts (clients) request only their own configuration catalogs. Puppet configures hosts in two steps:

- ✦ It compiles a catalog

- » It applies that catalog to the appropriate host

In the agent/master setup, the Puppet client sends facts gathered by **Facter** and other information to the Puppet Master. The Puppet Master compiles a catalog based on these facts, and then sends this catalog to the client. The client sends a report of all the changes it made, or would have made if the **--noop** parameter had been used, to the Puppet Master, which in turn sends the results to Foreman. This catalog describes the desired state for one specific host. It lists the resources to manage on that host, including any dependencies between those resources. The agent applies the catalog to the host.

This communication between master and agent occurs every 30 minutes by default. You can specify a different value in the `/etc/puppet/puppet.conf` file using the **`runinterval`** parameter. You can also run **`puppet agent apply`** to initiate communication manually.

13.4.2. Using Facter and Facts

Facter is Puppet's system inventory tool, and includes a large number of built-in facts. You can run **Facter** at the command line on a local host to display fact names and values. You can extend **Facter** with custom facts, and then use these to expose site-specific details of your hosts to your Puppet manifests. You can also use the facts provided by **Facter** to inform conditional expressions in Puppet.

Puppet determines a system state based on resources; for example, you can tell Puppet that the **httpd** service should always be running and Puppet knows how to handle that. If you are managing different operating systems, you can use the **`osfamily`** fact to create conditional expressions to tell Puppet which service to watch or which package to install. You can use the **`operatingsystemmajrelease`** and **`versioncmp`** parameters to create conditional expressions based on different versions of the same operating system. See [Example 13.1, “Using Conditional Expressions with Facts”](#) for an example of using conditional expressions.

Example 13.1. Using Conditional Expressions with Facts

```
if $::osfamily == 'RedHat' {
  if $::operatingsystemmajrelease == '6' {
    $ntp_service_name = 'ntpd'
  }

  elseif versioncmp($::operatingsystemmajrelease, '7') >= 0 {
    $ntp_service_name = 'chrony'
  }
}
```



Note

This example uses the expression `"versioncmp($::operatingsystemmajrelease, '7') >= 0"` to test for version 7 or later of Red Hat Enterprise Linux. Do not use the expression `"$::operatingsystemmajrelease >= '7'"` to perform this test. See <https://docs.puppetlabs.com/references/latest/function.html#versioncmp> for more information about this and other Puppet functions.

Puppet also sets other special variables that behave a lot like facts. See [Special Variables Added by Puppet](#) [3] and [Core Facts](#) [4] for more information.

13.4.2.1. Displaying Facts for a Particular Host

Puppet can access Facter's built-in core facts as well as any custom or external facts present in your Puppet modules. You can view available facts from the command line (**facter -p**) and also from the web UI (**Monitor → Facts**). You can browse the list of facts or use the **Search** box to search for specific facts. For example, type "**facts.**" to display a list of available facts.



Note

The list of available facts is very long. The UI only displays 20 facts at a time. The list of facts gradually filters as you enter more details. For example, type "facts.e" to display all facts that begin with the letter "e."

Procedure 13.16. To View Facts for a Particular Host:

1. On the main menu, click **Hosts** → **All Hosts** and then click the name of the host that you want to inspect.
2. In the **Details** pane, click **Facts** to display all known facts about the host.



Note

- ✧ For any fact listed on this page, you can click **Chart** to display a chart of the distribution of this fact name over all managed hosts.
- ✧ You can bookmark a search to make it easier to use in the future. When you have refined your search, click the drop-down arrow next to the **Search** button, and click **Bookmark this search**. Bookmarked searches appear in the **Search** drop-down list, and also under **Administer → Bookmarks** on the main menu.

13.4.2.2. Searching for Hosts based on Facts

You can use Facter information to search for specific hosts. This means that you can search for all hosts that match specific fact criteria, such as **facts.architecture = x86_64**.

Procedure 13.17. To Search for Hosts Based on Facts:

1. On the main menu, click **Monitor** → **Facts** to display the **Fact Values** page.
2. In the **Search** field, start typing the name of the fact that you want to filter by. You can search by specific name, name/value pairs, and so on.
3. Click **Search** to retrieve the list of matching hosts.

13.4.2.3. Custom Fact Reporting

Obtaining custom information from managed hosts is fully supported with Red Hat Satellite 6. This section illustrates using a Puppet module obtained from Puppet forge, but the principle applies equally for other sources of Puppet modules.

The number of facts reported via the standard `Facter` interface can be extended. For example, to gather a fact for use as a variable in modules. If a fact that describes the packages installed was available, you could search this data and make informed configuration management decisions based on the information.

To obtain a report on the packages installed on a host the process is as follows:

- The manifest **pkginventory** is obtained from Puppet Forge and saved to the base system.
- The Puppet module is added to a content view and then this is promoted to a system and deployed to that system.
- The facts for the system are then queried using a package name. In this example, for a host called *hostname* and using a Satellite user with credentials *username* and *password*, the following API query would return the facts that matched the search string "bash":

```
curl -u username:password -X GET
http://localhost/api/hosts/:hostname/facts?search=bash
{"hostname":{"pkg_bash":"4.2.45-5.el7_0.4"}}
```

The search returns the package version. This could then be used to populate an external database.

Adding the **pkginventory** Puppet Module

To add the **pkginventory** Puppet module to the Red Hat Satellite Server application, download the module from <https://forge.puppetlabs.com/ody/pkginventory> to the base system where the Satellite Server application is installed and then follow the procedures below.

Procedure 13.18. Uploading a Puppet Module to a Repository

Puppet modules are usually stored in a custom repository named Puppet Modules. This procedure assumes you have made a custom repository with that name. If you have not yet made a custom repository for Puppet Modules, see [Creating Custom Products and Repositories](#) in the [Red Hat Satellite 6.1 Provisioning Guide](#).

1. Download the Puppet module to the base system. Modules that are downloaded will have a **.tar.gz** extension.
2. Click **Content** → **Products** and then click the product name in the **Name** field associated with the Puppet module repository. For example, **Custom Products**.
3. On the **Repositories** tab, select the Puppet Modules repository you want to modify. For example, **Puppet Modules**.
4. In the **Upload Puppet Module** section, click **Browse**, and navigate to the module that you downloaded.
5. Click **Upload**.

Procedure 13.19. Adding a Module to a Content View

To distribute a Puppet module to clients, content hosts, the module must be applied to a Content View and published. Follow this procedure to add a module to a Content View.

1. Click **Content** → **Content Views** and then select a content view from the **Name** menu.
2. On the **Puppet Modules** tab, click **Add New Module**. A list of installed modules appears.

3. From the **Actions** column, click **Select a Version** to select the module you want to add. A table of available versions appears.
4. Click **Select Version** next to the version of the module that you want to add.
5. Click **Publish New Version** to create the new Content View.
6. Optionally add a description and click **Save**.

[3] https://docs.puppetlabs.com/puppet/3.7/reference/lang_facts_and_builtin_vars.html#special-variables-added-by-puppet

[4] https://docs.puppetlabs.com/facter/latest/core_facts.html

Chapter 14. Configuring Hosts

In Red Hat Satellite, hosts are client systems which have Red Hat Subscription Manager installed. Red Hat Subscription Manager sends updates to Red Hat Satellite and Red Hat Satellite provides updates to these client systems.

Hosts must be registered in order to be managed. After a host has been registered, it can be viewed and edited in the **Hosts** tab. This enables a user to add and manage subscriptions, add and remove software packages, and apply updates.

14.1. Creating a Host

The following procedure describes how to create a host in Red Hat Satellite.

Procedure 14.1. To Create a Host:

1. Click **Hosts** → **New Host**.
2. On the **Host** tab, enter the required details.
3. On the **Puppet Classes** tab, select the puppet classes you want to include.
4. On the **Network** tab, perform the following actions:
 - a. Enter the **Domain** and **Realm** details. It is required to specify a domain to make the host provisioning possible. This automatically updates the **Subnet** list with a selection of suitable subnets.
 - b. Enter the **Primary Interface** details. If there is a DHCP-enabled Capsule Server on the selected subnet, the IP address is automatically suggested. Click **Suggest new** to generate a different address.
 - c. Optionally, click **Add Interface** to include an additional network interface. See [Section 14.4, “Configuring an Additional Network Interface”](#) for details.
5. On the **Operating System** tab, enter the required details. You can select a partition table from the drop-down list or enter a custom partition table in the **Custom partition table** field. You cannot specify both.
6. On the **Parameters** tab, click **Add Parameter** to add any required parameters. This includes all Puppet Class Parameters and Host Parameters associated with the host.
7. On the **Additional Information** tab, enter additional information about the host.
8. Click **Submit** to complete your provisioning request.

14.2. Configuring a Host for Registration

Red Hat Enterprise Linux hosts register to the Red Hat Network (RHN) by default. You must update each host configuration so that they register to and update from the correct Red Hat Satellite Server.

Prerequisites

Address the following requirements before proceeding:

- ✧ Hosts must be using the following Red Hat Enterprise Linux Version:
 - 5.8 or later (5.7 or later on s390x)
 - 6.4 or later
 - 7.0 or later
- ✧ All architectures of Red Hat Enterprise Linux are supported (i386, x86_64, s390x, ppc_64)
- ✧ On the Red Hat Satellite Server, ensure that the date and time are correct and synchronized with the client.
- ✧ On each client system, address the following requirements:
 - Ensure that the date and time are correct and synchronized with the server.
 - Enable **ntpd** or a similar time synchronization tool in all virtual environments:

For Red Hat Enterprise Linux 6:

```
# chkconfig ntpd on; service ntpd start
```

For Red Hat Enterprise Linux 7:

```
# systemctl start chronyd; systemctl enable chronyd
```

The following procedure shows how to automatically configure your client system to register to Red Hat Satellite.

Procedure 14.2. To Automatically Configure a Host:

1. Take note of the Red Hat Satellite's fully qualified domain name (FQDN), for example *satellite.example.com*.
2. Open a terminal console and log in as root.
3. Download and install a copy of the CA Certificate for the host from the Red Hat Satellite FQDN:

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



Note

katello-ca-consumer-hostname-1.0-1.noarch.rpm is an additional *katello-ca-consumer* rpm available that contains the server's hostname. The *katello-ca-consumer-latest.noarch.rpm* rpm will always reflect the most updated version. Both serve the same purpose.

14.3. Registration

14.3.1. Registering a Host

These steps show how to register hosts in Red Hat Satellite Server. Hosts provisioned by Satellite Server appear on the **Hosts** page accessible through **Hosts** → **All hosts**. Hosts registered to the Satellite Server via Red Hat Subscription Manager, which can occur either during the post phase of a kickstart or through the terminal, will appear on the **Content Hosts** page accessible through **Hosts** → **Content Hosts**.

Prerequisites

- ✦ Ensure that all steps in [Section 14.2, “Configuring a Host for Registration”](#) have been completed.
- ✦ Make sure there is a pre-existing activation key for the system or create an activation key for the system. See [Section 11.1, “Creating an Activation Key”](#) for instructions on creating an activation key.

Procedure 14.3. To Register Systems:

1. Open a terminal console and login as the **root** user on the command line.
2. Clear old system data in preparation for registering. This makes sure that your updated system data is uploaded correctly.

```
# subscription-manager clean
```

3. Register the system using the Red Hat Subscription Manager (RHSM):

```
# subscription-manager register --org your_org_name --activationkey  
your_activation_key
```



Note

Activation keys will allow you to add environments, provisioning templates and dictate what subscriptions are available and should be applied to the registering system.

There are various options that may be added. For more information, use the command **man subscription-manager**.

The command line output after the registration should look like:

```
# subscription-manager register --org MyOrg --activationkey TestKey-1  
The system has been registered with id: 62edc0f8-855b-4184-b1b8-72a9dc793b96
```

The system should now appear in the Red Hat Satellite Server.



Note

For systems with Red Hat Enterprise Linux 6.3, the release version defaults to Red Hat Enterprise Linux 6 Server. To ensure that it is pointing to the 6.3 repository, follow these steps:

1. On Red Hat Satellite, select **Hosts** → **Content Hosts**.
2. Click on the **Details** tab.
3. Click the name of the host that needs to be changed.
4. In the **Content Host Content** section click the edit icon to the right of **Release Version**.
5. Select '6.3' from the **Release Version** drop-down menu.
6. Click **Save**.

14.3.2. Installing the Katello Agent

The following procedure shows how to install the Katello agent on a Satellite 6 host. The *katello-agent* package depends on the *gofer* package that provides the *goferd* service. This service must be enabled so that the Red Hat Satellite Server or Capsule Server can provide information about errata that are applicable for content hosts.

Prerequisites

Satellite version 6.1 and later require that you enable the **Satellite Tools** repository. The **Red Hat Common** repositories are no longer used and are not compatible with Satellite version 6.1 and later.

Procedure 14.4. Verify the Satellite Tools Repository is Enabled

The **Satellite Tools** repository must be enabled, synchronized to the Red Hat Satellite server and made available to your hosts as it provides the required packages.

1. Open the Satellite web UI, navigate to **Content** → **Red Hat Repositories** and click on the **RPMs** tab.
2. Find and expand the **Red Hat Enterprise Linux Server** item.
3. Find and expand the **Red Hat Satellite Tools 6.1 (for RHEL VERSION Server) (RPMs)** item.

If the **Red Hat Satellite Tools 6.1** items are not visible, it may be because they are not included in the subscription manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the subscription manifest and import it into Satellite.

4. Ensure the **Enabled** check box beside the repository's name is selected. If not, select it.

Enable the **Satellite Tools** repository for every supported major version of Red Hat Enterprise Linux running on your hosts.

Procedure 14.5. To Install Katello Agent:

1. On the host, open a terminal console and log in as the **root** user.
2. Verify that the **satellite-tools** repository is enabled, using the following command:

```
# yum repolist enabled | grep -i satellite-tools
```

If the **satellite-tools** is not enabled, enable it using the following command:

```
# subscription-manager repos --enable satellite-tools
```

3. Install the **katello-agent** RPM package using the following command:

```
# yum install katello-agent
```

The **goferd** service is started and enabled automatically after successful installation of **katello-agent**.

14.3.3. Installing and Configuring the Puppet Agent

This section describes how to install and configure the Puppet agent on a Satellite 6 host. When you have correctly installed and configured the Puppet agent, you can navigate to **Hosts** → **All hosts** to list all hosts visible to Red Hat Satellite Server.

Prerequisites

Satellite version 6.1 and later require that you enable the **Satellite Tools** repository. The **Red Hat Common** repositories are no longer used and are not compatible with Satellite version 6.1 and later.

Procedure 14.6. Verify the Satellite Tools Repository is Enabled

The **Satellite Tools** repository must be enabled, synchronized to the Red Hat Satellite server and made available to your hosts as it provides the required packages.

1. Open the Satellite web UI, navigate to **Content** → **Red Hat Repositories** and click on the **RPMs** tab.
2. Find and expand the **Red Hat Enterprise Linux Server** item.
3. Find and expand the **Red Hat Satellite Tools 6.1 (for RHEL VERSION Server) (RPMs)** item.

If the **Red Hat Satellite Tools 6.1** items are not visible, it may be because they are not included in the subscription manifest obtained from the Customer Portal. To correct that, log in to the Customer Portal, add these repositories, download the subscription manifest and import it into Satellite.

4. Ensure the **Enabled** check box beside the repository's name is selected. If not, select it.

Procedure 14.7. To Install and Enable the Puppet Agent:

1. On the host, open a terminal console and log in as the **root** user.
2. Verify that the **satellite-tools** repository is enabled, using the following command:

```
# yum repolist enabled | grep -i satellite-tools
```

If the **satellite-tools** is not enabled, enable it using the following command:

```
# subscription-manager repos --enable satellite-tools
```

3. Install the Puppet agent RPM package using the following command:

```
# yum install puppet
```

4. Configure the puppet agent to start at boot:

- A. On Red Hat Enterprise Linux 6:

```
# chkconfig puppet on
```

- B. On Red Hat Enterprise Linux 7:

```
# systemctl enable puppet
```

Procedure 14.8. Configuring the Puppet Agent

Prerequisites

You must meet the following conditions before continuing with this task:

- ✱ The host must be registered to the Red Hat Satellite Server.
- ✱ The Satellite Tools repository must be enabled.
- ✱ Puppet packages must be installed on the host.

1. Configure the Puppet agent by changing the `/etc/puppet/puppet.conf` file:

```
# vi /etc/puppet/puppet.conf
```

```
[main]
# The Puppet log directory.
# The default value is '$vardir/log'.
logdir = /var/log/puppet

# Where Puppet PID files are kept.
# The default value is '$vardir/run'.
rundir = /var/run/puppet

# Where SSL certificates are kept.
# The default value is '$confdir/ssl'.
ssldir = $vardir/ssl

[agent]
# The file in which puppetd stores a list of the classes
# associated with the retrieved configuration. Can be loaded in
# the separate ``puppet`` executable using the ``--loadclasses``
# option.
# The default value is '$confdir/classes.txt'.
classfile = $vardir/classes.txt
pluginsync = true
report = true
```

```

ignoreschedules = true
daemon = false
ca_server = satellite.example.com
server = satellite.example.com
environment = KT_Example_Org_Library_RHEL6Server_3

# Where puppetd caches the local configuration. An
# extension indicating the cache format is added automatically.
# The default value is '$confdir/localconfig'.
localconfig = $vardir/localconfig

```



Important

Set the **environment** parameter to the host's Puppet environment from the Satellite server. The Puppet environment label contains the organization label, lifecycle environment, content view name, and the content view ID. To see a list of Puppet environments in the Satellite 6 web UI, navigate to **Configure** → **Environments**.

2. Run the Puppet agent on the host:

```
# puppet agent -t --server satellite.example.com
```

3. Sign the SSL certificate for the puppet client through the Satellite Server web interface:

- a. Log in to the Satellite Server through the web interface.
- b. Select **Infrastructure** → **Capsules**.
- c. Click **Certificates** to the right of the required host.
- d. Click **Sign**.
- e. Rerun the **puppet agent** command:

```
# puppet agent -t --server satellite.example.com
```



Note

When the Puppet agent is configured on the host it will be listed under **All Hosts** but only when **Any Organization** is selected as the host will not be assigned to an organization or location. To assign the host to an organization see [Section 4.1.3, “Editing an Organization”](#) and to assign the host to a location see [Section 4.2.2, “Editing a Location”](#).

14.4. Configuring an Additional Network Interface

Red Hat Satellite supports specifying multiple network interfaces for a single host. You can configure these interfaces when creating a new host as described in [Section 14.1, “Creating a Host”](#) or when editing an existing host.

There are several types of network interfaces that you can attach to a host. When adding a new interface, select one of:

- ✱ **Interface:** Allows you to specify an additional physical or virtual interface. There are two types of virtual interfaces you can create. Use *VLAN* when the host needs to communicate with several (virtual) networks using a single interface, while these networks are not accessible to each other. Another type of virtual interface is *alias*, which is an additional IP address attached to an existing interface. See [Section 14.4.2, “Adding a Virtual Interface”](#), or [Section 14.4.1, “Adding a Physical Interface”](#) for details.
- ✱ **Bond:** Creates a bonded interface. NIC bonding is a way to bind multiple network interfaces together into a single interface that appears as a single device and has a single MAC address. This enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy. See [Section 14.4.3, “Adding a Bonded Interface”](#) for details.
- ✱ **BMC:** *Baseboard Management Controller* (BMC) allows you to remotely monitor and manage physical state of machines. See "Using Power Management Features on Managed Hosts" in the [Red Hat Satellite Installation Guide](#) for more information on BMC, and [Section 14.4.4, “Adding a Baseboard Management Controller \(BMC\) Interface”](#) for details on configuring a BMC interface.



Note

Additional interfaces have by default the **Managed** flag enabled, which means the new interface is configured automatically during provisioning by the DNS and DHCP Capsule Servers associated with the selected subnet. This requires a subnet with correctly configured DNS and DHCP Capsule Servers. If you use a kickstart method for host provisioning, configuration files are automatically created for managed interfaces in the post-installation phase at `/etc/sysconfig/network-scripts/ifcfg-interface_id`.



Note

Virtual and bonded interfaces currently require a MAC address of a physical device. Therefore, the configuration of these interfaces works only on bare-metal hosts.

14.4.1. Adding a Physical Interface

The following steps show how to add an additional physical interface to a host.

Procedure 14.9. To Add a Physical Interface:

1. Navigate to **Hosts** → **All hosts** to view available hosts.
2. Click **Edit** next to the host you want to edit.
3. On the **Network** tab, click **Add Interface**.
4. Keep the *Interface* option selected in the **Type** menu.
5. Specify a **MAC address** of the additional interface. This setting is required.

6. Specify the device **Identifier**, for example `eth0` or `eth1.1`. Identifier is used for bonded interfaces (in the **Attached devices** field, see [Procedure 14.11, “To Add a Bonded Interface:”](#)), VLANs and aliases (in the **Attached to** field, see [Procedure 14.10, “To Add a Virtual Interface:”](#)).
7. Specify the **DNS name** associated with the host's IP address. Satellite saves this name in the Capsule Server associated with the selected domain (the "DNS A" field) and the Capsule Server associated with the selected subnet (the "DNS PTR" field). A single host can therefore have several DNS entries.
8. Select a domain from the **Domain** drop-down menu. To create and manage domains, navigate to **Infrastructure → Domains**.
9. Select a subnet from the **Subnet** drop-down menu. To create and manage subnets, navigate to **Infrastructure → Subnets**.
10. Specify the interface **IP address**. Managed interfaces with assigned DHCP Capsule Server require this setting for creating a DHCP lease. DHCP-enabled managed interfaces provide an automatic suggestion of IP address.
11. Decide if the interface will be managed. If the **Managed** check box is selected, the interface configuration is pulled from the associated Capsule Server during provisioning, and DNS and DHCP entries are created. If using kickstart provisioning, a configuration file is automatically created for the interface.
12. Select the **Virtual NIC** check box to create a virtual interface. See [Section 14.4.2, “Adding a Virtual Interface”](#) for details.
13. Click **OK** to save the interface configuration, and then click **Submit** to apply the changes to the host.

14.4.2. Adding a Virtual Interface

The following steps show how to configure an additional virtual interface for a host.

Procedure 14.10. To Add a Virtual Interface:

1. Navigate to **Hosts → All hosts** to view available hosts.
2. Click **Edit** next to the host you want to edit.
3. On the **Network** tab, click **Add Interface**.
4. Keep the *Interface* option selected in the **Type** menu.
5. Specify the general interface settings. The applicable configuration options are the same as for the physical interfaces described in [Section 14.4.1, “Adding a Physical Interface”](#). Specify **MAC address** for managed virtual interfaces so that the configuration files for provisioning are generated correctly. However, **MAC address** is not required for virtual interfaces that are not managed. If creating a VLAN, specify ID in the form of `eth1.10` in the **Identifier** field. If creating an alias, use ID in the form of `eth1:10`.
6. Select the **Virtual NIC** check box. Additional configuration options specific to virtual interfaces are appended to the form:
 - ✱ **Tag:** You can specify tags per interface to provide a higher-level segmentation of the network. If left blank, managed interfaces inherit the tag from the VLAN ID of the associated subnet, given that this subnet has the VLAN ID specified. User-specified entries from this field are not applied on alias interfaces.

- » **Attached to:** Specify the identifier of the physical interface to which the virtual interface belongs, for example eth1. This setting is required.
7. Click **OK** to save the interface configuration. Then click **Submit** to apply the changes to the host.

14.4.3. Adding a Bonded Interface

The following steps show how to configure a bonded interface for a host.

Procedure 14.11. To Add a Bonded Interface:

1. Navigate to **Hosts** → **All hosts** to view available hosts.
2. Click **Edit** next to the host you want to edit.
3. On the **Network** tab, click **Add Interface**.
4. Select *Bond* from the **Type** menu. Additional type-specific configuration options are appended to the form.
5. Specify the general interface settings. The applicable configuration options are the same as for the physical interfaces described in [Section 14.4.1, “Adding a Physical Interface”](#). Bonded interfaces use IDs in the form of *bond0* in the **Identifier** field. It is sufficient if you specify a single MAC address in the **MAC address** field.
6. Specify the configuration options specific to bonded interfaces:
 - » **Mode:** Select the *bonding mode* that defines a policy for fault tolerance and load balancing. See [Table 14.1, “Bonding Modes Available in Red Hat Satellite”](#) for a brief description of individual bonding modes.
 - » **Attached devices:** Specify a comma separated list of identifiers of attached devices. These can be physical interfaces or VLANs.
 - » **Bond options:** Specify a space separated list of configuration options, for example *miimon=100*. There are several configuration options you can specify for the bonded interface, see [Red Hat Enterprise Linux 7 Networking Guide](#) for details.
7. Click **OK** to save the interface configuration. Then click **Submit** to apply the changes to the host.

Table 14.1. Bonding Modes Available in Red Hat Satellite

Bonding Mode	Description
balance-rr	Transmissions are received and sent out sequentially on each bonded interface.
active-backup	Transmissions are received and sent out via the first available bonded interface. Another bonded interface is only used if the active bonded interface fails.
balance-xor	Transmissions are based on the selected hash policy. In this mode, traffic destined for specific peers will always be sent over the same interface.
broadcast	All transmissions are sent on all bonded interfaces.
802.a3	Creates aggregation groups that share the same settings. Transmits and receives on all interfaces in the active group.

Bonding Mode	Description
balance-tlb	The outgoing traffic is distributed according to the current load on each bonded interface.
balance-alb	Receive load balancing is achieved through Address Resolution Protocol (ARP) negotiation.

14.4.4. Adding a Baseboard Management Controller (BMC) Interface

This section describes how to configure a *baseboard management controller* (BMC) interface for a host that supports this feature.

Prerequisites

Ensure the following prerequisites are satisfied before proceeding:

- ✧ BMC is enabled on the Capsule. If required, see [Procedure 14.12, “To Enable BMC Power Management on an Existing Capsule:”](#).
- ✧ The *ipmitool* package is installed.
- ✧ You know the MAC address, IP address, and other details of the BMC interface on the host, and the appropriate credentials for that interface.



Note

You only need the MAC address for the BMC interface if the BMC interface is managed. This is so that it can create a DHCP reservation.

Procedure 14.12. To Enable BMC Power Management on an Existing Capsule:

1. Ensure the following lines exist in the `/etc/foreman-proxy/settings.d/bmc.yml` file. Create the file if necessary.

```
:enabled: true
:bmc_default_provider: your_bmc_provider
```

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

```
# service foreman-proxy restart
```

3. Refresh the features for the Capsule.
 - a. Log in to the Satellite web UI, and navigate to **Infrastructure → Capsules**.
 - b. Identify the Capsule whose features you need to refresh. In the drop-down list on the right, click **Refresh features**. The list of features in the **Features** column should now include BMC.

Procedure 14.13. To Add a BMC Interface:

1. Navigate to **Hosts → All hosts** to view available hosts.

2. Click **Edit** next to the host you want to edit.
3. On the **Network** tab, click **Add Interface**.
4. Select *BMC* from the **Type** menu. Type-specific configuration options are appended to the form.
5. Specify the general interface settings. The applicable configuration options are the same as for the physical interfaces described in [Section 14.4.1, “Adding a Physical Interface”](#).
6. Specify the configuration options specific to BMC interfaces:
 - ✳ **Username, Password:** Specify any authentication credentials required by BMC.
 - ✳ **Provider:** Specify the BMC provider.
7. Click **OK** to save the interface configuration, and then click **Submit** to apply the changes to the host.

14.5. Removing a Host

The following procedure shows how to remove a host from Red Hat Satellite.

Procedure 14.14. To Remove a Host:

1. Click **Hosts** → **All hosts** or **Hosts** → **Content Hosts**.
2. Choose the hosts to be removed.
3. Click **Select Action** and choose **Delete Hosts** from the drop-down menu.
4. A confirmation pop-up box will appear. Select **Yes** to remove the host from Red Hat Satellite permanently.

Chapter 15. Discovering Bare-metal Hosts on Satellite

Red Hat Satellite 6.1 ships with the Discovery plug-in already installed. The Discovery plug-in enables automatic bare-metal discovery of unknown hosts on the provisioning network. These new hosts are registered to the Satellite Server and the Puppet agent on the client uploads system facts collected by Facter, such as serial ID, network interface, memory, and disk information. After registration, the hosts are displayed on the **Discovered Hosts** page in the Satellite web UI. You can then initiate provisioning either manually (using the web UI, CLI, or API) or automatically, using predefined discovery rules.

The Discovery plug-in communicates through the Satellite Capsule, which has direct access both to the provisioning network and the Satellite Server instance. It is possible to discover hosts directly from the Satellite Server, but Red Hat recommends the following scheme be used:

```
Satellite Server (Satellite Server Discovery plug-in) <--> Satellite Capsule
(Satellite Capsule Discovery plug-in) <--> Discovered Host (Satellite
Discovery image)
```

The Satellite Discovery plug-in consists of three different components:

The Satellite Server Discovery plug-in

This runs on the Satellite Server and provides API and UI functionality for working with discovered hosts. The *ruby193-rubygem-foreman_discovery* package contains this plug-in.

The Satellite Capsule Discovery plug-in

This is a communication proxy between discovered hosts on a provisioning network and the Satellite Server. The *rubygem-smart_proxy_discovery* package contains this plug-in.

The Satellite Discovery image

This is the minimal operating system based on Red Hat Enterprise Linux that is PXE-booted on hosts to acquire initial hardware information and to check in to the Satellite Server. Discovered hosts keep running the Satellite Discovery image until they are rebooted into Anaconda, which then initiates the provisioning process. The *foreman-discovery-image* package contains this image. It must be installed on the Satellite Capsule that provides TFTP services.

15.1. Configuring the Satellite Discovery Plug-in

The following sections describe how to configure the Satellite Discovery plug-in and how to prepare the PXE-boot template on the Satellite Server.

15.1.1. Deploying the Satellite Discovery Image

Install the package containing the Satellite Discovery image on the Satellite Capsule that provides TFTP services (not on the Satellite Server itself):

```
# yum install foreman-discovery-image
```

This package contains the Linux kernel and initial RAM disk image as a bootable ISO file which is used for PXE-booting discovered hosts. You can run the following command to investigate the contents of the package. This produces output similar to the following:

```
$ rpm -ql foreman-discovery-image
/usr/share/foreman-discovery-image
/usr/share/foreman-discovery-image/fdi-image-rhel_7-2.1.0-20150212.1.iso
```

When you install this package, it extracts the kernel and image from the ISO file into the TFTP directory and creates symbolic links to the latest versions of the image and kernel. Use the symbolic links in the PXE-boot provisioning template to make sure that you do not need to change the version in the template every time the *foreman-discovery-image* package is upgraded. For example:

```
$ find /var/lib/tftpboot/boot
/var/lib/tftpboot/boot
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-img
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-vmlinuz
/var/lib/tftpboot/boot/fdi-image-rhel_7-img
/var/lib/tftpboot/boot/fdi-image-rhel_7-vmlinuz
```

The last two lines in the above output contain symbolic links to be used in the PXE-boot provisioning template. For more information see [Section 15.1.2, “Configuring PXE-booting”](#).



Note

Currently, only Red Hat Enterprise Linux 7 Discovery images are provided, even for Satellite 6 installations on Red Hat Enterprise Linux 6. If there are discovered hosts running during the upgrade of the *foreman-discovery-image* package, reboot them all to load the updated version of the image as soon as possible. This can be done through the Satellite 6 web UI, CLI, or API.

15.1.2. Configuring PXE-booting

When an unknown host is booted on the provisioning network, the Satellite Server provides a PXELinux boot menu with a single option; to boot from the local hard drive. The following procedure describes how to change this behavior in order to enable hardware discovery. This requires changing several variables in the PXE Linux global default template. These variables are described below:

- The **KERNEL** and **APPEND** lines in the template use symbolic links, created when installing the *foreman-discovery-image* package (see [Section 15.1.1, “Deploying the Satellite Discovery Image”](#)). The URLs are relative to the `/var/lib/tftpboot/` directory. Ensure the **APPEND** parameters are specified on a single line.
- The **proxy.type** variable can be set to either **proxy** (recommended) or **foreman**. When the variable is set to **proxy**, all communication goes through the Satellite Capsule. When the variable is set to **foreman**, the communication goes directly to Satellite Server. Examples in this chapter assume **proxy.type** is set to **proxy**.

- The **proxy.url** variable specifies the URL of the Satellite Capsule or Satellite Server, depending on the **proxy.type** setting. Both **HTTP** and **HTTPS** schemes are supported. The default port is 9090 for accessing the Satellite Capsule (**proxy.type=proxy**), and 80 for direct communication with the Satellite Server (**proxy.type=foreman**).
- The **IPAPPEND 2** setting detects interfaces connected to the provisioning network. The image will not boot correctly if this option is removed or modified.

Procedure 15.1. To Configure PXE-booting:

1. In the Satellite web UI, navigate to **Hosts** → **Provisioning Templates**.
2. Edit the *PXELinux global default* template. Add the following menu entry to the template:

```

LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live://fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset
proxy.url=https://SATELLITE_CAPSULE_URL:9090 proxy.type=proxy
IPAPPEND 2

```

3. Set the new menu entry to be the default by modifying the **ONTIMEOUT** variable:

```
ONTIMEOUT discovery
```

4. Click **Build PXE Default** at the top of the **Provisioning Templates** page. This instructs the TFTP proxy to rewrite the **pxelinux.cfg/default** file. Repeat this step every time a change is made to the default template to ensure that the changes are deployed on the TFTP Satellite Capsule.

As an alternative to the above procedure, you can omit the **proxy.url** variable from the PXE-boot template. In this case, the Discovery image searches the DNS configuration file for an SRV record named **x-foreman.tcp**. The **proxy.url** variable must be set to **proxy** in this case. The DNS server must also be suitably configured. For example, the following configuration statement specifies the Capsule to be used with HTTPS:

```
_x-foreman._tcp SRV 0 5 9090 capsule
```

Here, *capsule* is the name of the Capsule that is included in the DNS configuration.



Note

Satellite 6.1 only allows you to specify only one Capsule URL for all subnets where hosts can be discovered. Because templates cannot be used per subnet, use a DNS alias name on all networks. Alternatively, use an SRV record.



Important

The DNS servers from the DHCP settings are taken into account only for the interface that is specified via the **BOOTIF** variable. **BOOTIF** is set automatically by the **IPAPPEND** variable in the PXE template. This means that when a system has multiple NICs, DNS will only work for the interface that it was booted from.

15.1.3. Reviewing Global Discovery Settings

You can review global settings related to the Discovery plug-in in the Satellite web UI. Navigate to **Administer** → **Settings** and open the **Discovered** tab. Notable settings are:

discovery_organization, discovery_location

These variables specify where to place the discovered hosts. By default, the discovered hosts are automatically placed under the first organization and location created.

discovery_fact

This variable specifies which incoming fact to use to determine the MAC address of the discovered host. By default, the PXELinux BOOTIF kernel command line option is used.

discovery_auto

This variable enables automatic provisioning according to specified rules. Set to false by default. Red Hat recommends that you test the configuration with manual provisioning before enabling **discovery_auto**. See [Section 15.3, “Provisioning Discovered Hosts”](#) for more information.

discovery_fact_column

This variable allows you to add any fact reported by Facter as an additional column in the list of discovered hosts.

15.2. Configuring the Satellite Capsule Discovery Plug-in

Ensure the **foreman_url** setting exists in the Satellite Capsule configuration file. The setting can appear as follows:

```
# grep foreman_url /etc/foreman-proxy/settings.yml
:foreman_url: https://satellite.example.com
```

The **capsule-installer** command configures this variable automatically, but Red Hat recommends that you check that the host responds correctly and there are no firewall rules blocking communication.

15.2.1. Configuring Discovery Subnets

You need to configure all subnets with discovered hosts to communicate through the Satellite Capsule. In the Satellite web UI, navigate to **Infrastructure** → **Subnets** and select the required Capsule for each subnet that needs to perform host discovery and ensure it is connected to the Discovery Capsule.

To verify that a Capsule has the Discovery plug-in enabled, navigate to **Infrastructure** → **Capsules**. The Discovery plug-in should appear in the list of features associated with the Capsule. Click **Refresh features** to ensure that the list is up-to-date.

15.2.2. Using Hammer with the Discovery Plug-in

To use the **hammer** command with the Discovery plug-in, you need to enable the Discovery plug-in in `/etc/hammer/cli.modules.d/foreman_discovery.yml` as follows:

```
:foreman_discovery:
  :enable_module: true
```

See [hammer configuration directories](#)^[5] for more information about the files and directories that **hammer** uses.

15.2.3. Reviewing User Permissions

When it first starts, the Satellite Capsule Discovery plug-in creates a role called **Discovery**. You can assign this role to non-administrative users to allow them to use the Discovery plug-in. Alternatively, assign the **perform_discovery** permission to an existing role. For more information on roles and permissions, see [Section 17.3, “Creating and Managing Roles”](#).

15.3. Provisioning Discovered Hosts

After you have correctly configured Discovery plug-ins on both the Satellite Server and the Capsule, you can automatically detect bare-metal hosts. To do so, boot a machine in any provisioning network that was configured with the PXE configuration template described in [Section 15.1.2, “Configuring PXE-booting”](#). The machine is automatically registered with the Satellite Server and appears in the **Hosts** → **Discovered Hosts** list in the Satellite web UI.

You can either provision the discovered host manually, or you can configure automatic provisioning.

15.3.1. Manually Provisioning Hosts

The following procedure describes how to manually provision discovered hosts from the Satellite web UI.

Procedure 15.2. To Manually Provision a Discovered Host:

1. Navigate to **Hosts** → **Discovered Hosts**.
2. Select the host you want to provision and click **Provision**.
3. On the host's **Edit** page, complete the necessary details, and then click **Save**.

When the host configuration is saved, Satellite modifies the host's PXELinux file on the TFTP server and reboots the discovered host. It then boots into an installer for the chosen operating system, and finally into the installed operating system.

If you decide to re-provision an existing discovered host, delete the operating system from the machine and reboot it. The host then reappears on the **Discovered Hosts** page.

15.3.2. Decommissioning Discovered Hosts

If a host is no longer required to be managed by Red Hat Satellite, the host must be decommissioned according to the following procedure to prevent it from being discovered again.

1. Shutdown the host.
2. Navigate to **Hosts** → **Discovered Hosts**.
3. In the **Name** column find the host you want to decommission and then select **Delete** from the corresponding **Edit** drop-down menu.

15.3.3. Automatically Provisioning Hosts

With Satellite 6.1, it is possible to define provisioning rules that will assign a host group to provisioned hosts and trigger provisioning automatically.

Procedure 15.3. To Create a Provisioning Rule:

1. Navigate to **Configure** → **Discovery rules**.
2. Click **New Rule**. Specify the following parameters of the provisioning rule:
 - ✳ **Name** is the name of the rule displayed in the list of rules. This name must not contain spaces or non-alphanumeric characters.
 - ✳ **Search** is the search statement used to match discovered hosts for the particular rule. You can use scoped search syntax to define it. See [Section 15.3.4, “Scoped Search Syntax”](#) for examples of using scoped search.
 - ✳ **Host Group** is the host group to be assigned to a matching host before starting the provisioning process. Make sure that the selected host group has all the required parameters set; required parameters are marked with an asterisk (*).
 - ✳ **Hostname** defines a pattern for assigning human-readable host names to the matching hosts. When left blank, the host name is assigned in the format "macMACADDRESS" by default. The same syntax used for provisioning templates is used in this instance. See [Section 15.3.5, “Host Name Patterns”](#) for more information and examples.
 - ✳ **Hosts limit** is the the maximum number of provisioned hosts per rule. If the limit is reached, the rule will not take effect until one or more hosts are deleted. Typical use cases are rules per server rack or row when it is necessary to change provisioning parameters such as host name or host group per entry. You can set this value to zero (0) to specify no limit.
 - ✳ **Priority** specifies the order of execution of rules. The value must be greater than or equal to zero. A lower value indicates a higher priority. If two rules have the same priority, the first rule encountered is applied.
 - ✳ **Enabled** provides the option to temporarily enable or disable rules.
3. Click **Submit** to save the rule.

By default, Satellite does not enable automatic discovery of hosts. The following procedure describes how to enable the **discovery_auto** variable to provide automatic provisioning according to specified rules.

Procedure 15.4. To Enable Automatic Provisioning:

1. Navigate to **Administer** → **Settings** → **Discovered** in the Satellite web UI.
2. Locate **discovery_auto** in the **Name** column, and set its value to **true**.
3. Click **Save**.

After you have defined some rules, Red Hat recommends that you discover a host and apply the rules using the **Auto discover** button on the host. This triggers auto-provisioning without the need to enable the global option.

15.3.4. Scoped Search Syntax

This section shows how to use scoped search syntax to filter the discovered hosts according to selected parameters. This is useful when creating a rule for automatic provisioning (see [Section 15.3.3, “Automatically Provisioning Hosts”](#)).

The search fields in the Satellite web UI support automatic completion to make building search strings easier. For example, you can test search patterns on the **Hosts** → **Discovered Hosts** page. The following are examples of typical search queries:

- ✱ facts.architecture = x86_64
- ✱ facts.bios_vendor ~ 'Dell*'
- ✱ facts.macaddress = "aa:bb:cc:dd:ee:ff"
- ✱ facts.macaddress_eth0 = "aa:bb:cc:dd:ee:ff"
- ✱ facts.ipaddress_eth1 ~ "192.168.*"
- ✱ facts.architecture ^ (x86_64,i386)



Note

The caret symbol (^) in scoped searches means "in" (the same usage as in SQL) and not "starts with" as it is used in regular expressions. You can review the full list of scoped search operators at https://github.com/wvanbergen/scoped_search/blob/master/lib/scoped_search/query_language/tokenizer.rb

In Satellite 6.1, all facts are strings, so it is not possible to do numeric comparisons. However, three important facts are extracted and converted to numbers. These are described in [Table 15.1, “Facts that Allow Numerical Comparison”](#).

Table 15.1. Facts that Allow Numerical Comparison

Search Parameter	Description	Example Usage
cpu_count	The number of CPUs	cpu_count >= 8
disk_count	The number of disks attached	disk_count < 10
disks_size	The total amount of disk space (in MiB)	disks_size > 1000000

15.3.5. Host Name Patterns

This section lists the host name patterns that you can use when creating a rule for automatic

provisioning (see [Section 15.3.3, “Automatically Provisioning Hosts”](#)).

The target host name template pattern has the same syntax as the provisioning templates (ERB). The domain is appended automatically. In addition to the **@host** attribute, the **rand()** function for random integers is available. For example:

- ✧ application-server-<%= rand(99999) %>
- ✧ load-balancer-<%= @host.facts['bios_vendor'] + '-' + rand(99999) %>
- ✧ wwwsrv-<%= @host.hostgroup.name %>
- ✧ minion-<%= @host.discovery_rule.name %>
- ✧ db-server-<%= @host.ip.gsub('.', '-') + '-' + @host.hostgroup.subnet.name %>>



Important

When creating host name patterns, ensure the resulting host names are unique. Host names must not start with numbers. A good approach is to use unique information provided by Facter (for example, the MAC address, BIOS or serial ID) or to otherwise randomize the host name.

15.3.6. Using the Discovery Plug-in on the Command Line

You can use the **hammer** command to perform certain tasks related to discovery. Run the **hammer -h** command to verify your configuration:

```
$ hammer -h | grep discovery
discovery           Manipulate discovered hosts.
discovery_rule      Manipulate discovered rules.
```

Use the **hammer discovery -h** command to view the available options. For example, you can use the following command to reboot a discovered host (assuming its ID is 130):

```
$ hammer discovery reboot -id 130
Host reboot started
```

15.4. Extending the Discovery Image

It is possible to extend the Satellite Discovery image with custom facts, software, or device drivers. You can also provide a compressed archive file containing extra code for the image to use.

First, create the following directory structure:

```
.
├── autostart.d
│   └── 01_zip.sh
├── bin
│   └── ntpdate
├── facts
│   └── test.rb
```

```
└─ lib
   └─ libcrypto.so.1.0.0
      └─ ruby
         └─ test.rb
```

Where:

- ✦ The **autostart.d** directory contains scripts that are executed in POSIX order by the image when it starts, but before the host is registered to Satellite.
- ✦ The **bin** directory is added to the `$PATH` variable; you can place binary files here and use them in the autostart scripts.
- ✦ The **facts** directory is added to the `FACTERLIB` variable so that custom facts can be configured and sent to Satellite.
- ✦ The **lib** directory is added to the `LD_LIBRARY_PATH` variable and **lib/ruby** is added to the `RUBYLIB` variable, so that binary files in **/bin** can be executed correctly.

New directives and options are appended to the existing environment variables (`PATH`, `LD_LIBRARY_PATH`, `RUBYLIB` and `FACTERLIB`). If you need to specify the path to something explicitly in your scripts, the zip contents are extracted to the **/opt/extension** directory on the image.

After creating the above directory structure, package it into a zip archive with the following command:

```
zip -r my_extension.zip .
```

You can create multiple zip files but be aware they will be extracted to the same place on the Discovery image, so files in later zips will overwrite earlier ones if they have the same file name.

To inform the Discovery image of the extensions it should use, place your zip files on your TFTP server with the Discovery image, and then update the `APPEND` line of the PXELinux template with the **fdi.zips** option where the paths are relative to the TFTP root. For example, if you have two archives at **\$TFTP/zip1.zip** and **\$TFTP/boot/zip2.zip**, use the following syntax:

```
fdi.zips=zip1.zip,boot/zip2.zip
```

See [Section 15.1.2, “Configuring PXE-booting”](#) for more information on updating the PXE template.

15.5. Troubleshooting Satellite Discovery

If a machine does not show up correctly in the Satellite web UI under **Hosts** → **Discovered Hosts**, inspect the following configuration areas to help isolate the error:

- ✦ Try redeploying the default PXE Linux template.
- ✦ Verify the **pxelinux.cfg/default** configuration file on the TFTP Capsule.
- ✦ Ensure adequate network connectivity between hosts, the Capsule, and the Satellite Server.
- ✦ Verify the **proxy.url** and **proxy.type** options in the default PXE Linux template.

- ✦ Ensure that the DNS is working correctly for that image, or use an IP address in the **proxy.url** option in the default PXE Linux template.
- ✦ Ensure that the DHCP server is delivering IP addresses to the booted image correctly.
- ✦ Ensure the discovered host (or virtual machine) has at least 500 MB of memory. Less memory can lead to various random kernel panic errors as the image needs to be extracted in-memory.

For gathering important system facts, use the **discovery-debug** command. It prints out system logs, network configuration, list of facts, and other information on the standard output. The typical use case is to redirect this output and copy it with the **scp** command for further investigation.

The first virtual console on the discovered host is reserved for systemd logs. Particularly useful system logs are tagged as follows:

- ✦ discover-host - initial facts upload
- ✦ foreman-discovery - facts refresh, reboot remote commands
- ✦ nm-prepare - boot script which pre-configures NetworkManager
- ✦ NetworkManager - networking information

Use TTY2 or higher to log in to a discovered host. The root account and SSH access are disabled by default, but you can enable SSH and set the root password using the following kernel command-line options in the Default PXELinux template on the APPEND line:

```
fdi.ssh=1 fdi.rootpw=redhat
```

[5] <https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md#locations>

Chapter 16. Configuring Host Collections

The *Host Collections* application tab is a system management tool that allows the administrator to:

- ✦ Add hosts to a collection.
- ✦ Apply a mass installation of packages, errata, or package groups to all host members of a host collection.
- ✦ Update specific packages, errata, or specific package groups to all host members.

16.1. Creating a Host Collection

The following procedure shows how to create host collections.

Procedure 16.1. To Create a Host Collection:

1. Click **Hosts** → **Host Collections**.
2. Click **New Host Collection**.
3. Add the Name and Description of the host collection.
4. Deselect **Unlimited Content Hosts** to specify the maximum number of hosts that will be allowed to the group. Otherwise, leave it checked to allow unlimited hosts to join the host collection.
5. Click **Save**.

16.2. Adding Hosts to a Host Collection

The following procedure shows how to add hosts to host collections.

Prerequisites

A host must be registered to Red Hat Satellite in order to add it to a Host Collection. Refer to [Section 14.3.1, “Registering a Host”](#) for information on how to register a host.

Procedure 16.2. To Add Hosts to a Host Collection:

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the host should be added.
3. On the **Content Hosts** tab, select the **Add** subtab.
4. Select the hosts to be added from the table and click **Add Selected**.

16.3. Adding Content to Host Collections

These steps show how to add content to host collections in Red Hat Satellite.

16.3.1. Adding Packages to a Host Collection

The following procedure shows how to add packages to host collections.

Prerequisites

- ✦ The content to be added should be available in one of the existing repositories or added prior to this procedure.
- ✦ Content should be promoted to the environment where the hosts are assigned.

Procedure 16.3. To Add Packages to Host Collections:

1. Click **Hosts → Host Collections**.
2. Click the host collection where the package should be added.
3. On the **Collection Actions** tab, click **Package Installation, Removal, and Updates**.
4. In the field provided, specify the package or package group name. Then click:
 - ✦ **Install** - if you want to install a new package
 - ✦ **Update** - if you want to update an existing package in the host collection

16.3.2. Adding Errata to a Host Collection

The following procedure shows how to add errata to host collections.

Prerequisites

- ✦ The errata to be added should be available in one of the existing repositories or added prior to this procedure.
- ✦ Errata should be promoted to the environment where the hosts are assigned.

Procedure 16.4. To Add Errata to a Host Collection:

1. Click **Hosts → Host Collections**.
2. Select the host collection where the errata should be added.
3. On the **Collection Actions** tab, click **Errata Installation**.
4. Select the errata you want to add to the host collection and click **Install Selected**.

16.4. Removing Content from a Host Collection

The following procedure shows how to remove packages from host collections.

Procedure 16.5. To Remove Content from a Host Collection:

1. Click **Hosts → Host Collections**.
2. Click the host collection where the package should be removed.
3. On the **Collection Actions** tab, click **Package Installation, Removal, and Updates**.

4. In the field provided, specify the package or package group name. Then click **Remove**.

16.5. Changing the Life Cycle Environment or Content View of a Host Collection

The following procedure shows how to change the assigned life cycle environment or content view of host collections.

Procedure 16.6. To Change the Life Cycle Environment or Content View of a Host Collection:

1. Click **Hosts** → **Host Collection**.
2. Selection the host collection where the life cycle environment or content view should be changed.
3. On the **Collection Actions** tab, click **Change assigned Life Cycle Environment or Content View**.
4. Select the life cycle environment to be assigned to the host collection.
5. Select the required content view from the drop-down list.
6. Click **Assign**.

16.6. Removing a Host from a Host Collection

The following procedure shows how to remove hosts from host collections.

Procedure 16.7. To Remove Hosts from a Host Collection:

1. Click **Hosts** → **Host Collections**.
2. Choose the desired host collection.
3. On the **Content Hosts** subtab, select the hosts you want to remove from the host collection.
4. Click **Remove Selected**.

16.7. Removing a Host Collection

The following procedure shows how to remove a host collection.

Procedure 16.8. To Remove a Host Collection:

1. Click **Hosts** → **Host Collections**.
2. Choose the host collection to be removed.
3. Click **Remove**. An alert box appears:

Are you sure you want to remove host collection *Host Collection Name*?

4. Click **Remove**.

16.8. Cloning a Host Collection

The following procedure shows how to clone a host collection.

Procedure 16.9. To Clone a Host Collection:

1. Click **Hosts** → **Host Collections**.
2. On the left hand panel, click the host collection you want to clone.
3. Click **Copy Collection**.
4. Specify a name for the cloned collection.
5. Click **Create**.

16.9. Reviewing Host Collection Details

The following procedure shows how to review host collection details.

Procedure 16.10. To Reviewing Host Collection Details:

1. Click **Hosts** → **Host Collections**.
2. Select the host collection you want to review and navigate to the **Details** tab.

Chapter 17. 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 17.1, “Creating and 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. See [Section 17.2, “Creating User Groups”](#) for more information on creating user groups.

Roles define a set of permissions and access levels. Each role contains one or more *permission filters* that specify the *actions* allowed for the role. Actions are grouped according to the *Resource type*. Once a role has been created, users and user groups can be associated with that role. This way, you can assign the same set of permissions to large groups of users. Red Hat Satellite provides a set of predefined roles and also enables creating custom roles and permission filters as described in [Section 17.3, “Creating and Managing Roles”](#).

17.1. Creating and Managing Users

For the administrator, Red Hat Satellite provides the ability to create, modify, and remove users. Also, it is possible to configure access permissions through assigning roles to users.

17.1.1. Creating a User

The following steps show how to create a user:

Procedure 17.1. To Create a User:

1. Navigate to **Administer** → **Users** and then click **New User**.
2. Enter the required details on the **User** tab.
3. On the **Locations** tab, select the required locations for this user.
4. On the **Organizations** tab, select the required organizations for this user.
5. On the **Roles** tab, select the required roles for this user. Active roles are displayed in the right panel.
6. Click **Submit** to create the user.

17.1.2. Editing a User

The following steps show how to edit details of an existing user:

Procedure 17.2. To Edit an Existing User:

1. Navigate to **Administer** → **Users**.
2. Click the user name of the user to be altered. General information about the user will appear on the right.
3. You can modify the user's username, first name, surname, email address, default location, default organization, language, and password in the **User** tab.

4. You can modify the user's assigned locations in the **Locations** tab.
5. You can modify the user's assigned organizations in the **Organizations** tab. If no organization is selected, the user can access all available organizations.
6. You can modify the user's assigned roles in the **Roles** tab.
7. Click **Save** to save your changes.

17.1.3. Assigning Roles to a User

By default, a new user has no roles assigned. The following procedure describes how to assign one or more roles to a user. You can select from predefined roles, or define a custom role as described in [Section 17.3.1, “Creating a Role”](#). You can apply a similar procedure to user groups.

Procedure 17.3. To Assign a Role to a User:

1. Navigate to **Administer** → **Users**.
2. Click the user name of the user that you want to modify. General information about the user appears on the right.
3. Click the **Roles** tab to display the list of available role assignments.
4. Select role you want to assign to the user in the **Roles** list. The list contains the predefined roles, as well as any custom roles, see [Table 17.1, “Predefined Roles Available in Red Hat Satellite”](#). Alternatively, select the **Administrator** check box to assign all available permissions to the selected user.
5. Click **Save**.

To view the roles assigned to any user, access the **Roles** tab as described in the first three steps of the above procedure. To remove a role, click the role name in the **Selected items** list in the **Roles** tab.

17.1.4. Configuring Email Notifications

The following procedure shows how to configure email notifications.

Procedure 17.4. To Configure Email Notifications:

1. Navigate to **Administer** → **Users**.
2. Click the user name of the user you want to edit.
3. On the **Mail Preferences** tab, select **Mail enabled** to enable updates.
4. Select the type of notifications the user will receive. The following notification types are available:
 - ✳ **Puppet error state** is a notification sent after a host reports an error related to Puppet. To enable these notifications, select **Subscribe** from the drop-down menu.
 - ✳ **Puppet summary** is a summary of Puppet reports. Choose the frequency of emails from the drop-down list that offers **Daily**, **Weekly**, or **Monthly** updates.
 - ✳ **Satellite Host Advisory** is a summary of applicable and installable errata for hosts managed by the user. Choose the frequency of emails from the drop-down list that

offers **Daily**, **Weekly**, or **Monthly** updates.

- » **Satellite Promote Errata** is 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 you to monitor what updates have been applied to which hosts. To enable these notifications, select **Subscribe** from the drop-down menu.
- » **Satellite Sync Errata** is a notification sent only after synchronizing a repository. It contains a summary of new errata introduced by the synchronization. To enable these notifications, select **Subscribe** from the drop-down menu.

5. Click **Submit**.

The configuration of outgoing emails from the Satellite server is stored in **/etc/foreman/email.yaml**. You can select to deliver messages through an **SMTP** server or using the **sendmail** command. For example, the following configuration uses SMTP as a delivery method:

```
production:
  email_delivery:
    delivery_method: :smtp
    smtp_settings:
      address: smtp.example.com
      port: 25
      domain: example.com
      authentication: :login
      user_name: satellite@example.com
      password: satellite
```

The **user_name** and **password** directives specify the login credentials for the SMTP server. The default **/etc/foreman/email.yaml** contains **authentication: :none**.

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

```
production:
  email_delivery:
    delivery_method: :smtp
    smtp_settings:
      enable_starttls_auto: true
      address: "smtp.gmail.com"
      port: '587'
      domain: "smtp.gmail.com"
      authentication: :plain
      user_name: "user@gmail.com"
      password: "password"
```

**Note**

If your SMTP server uses TLS authentication, perform one of the following steps:

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

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

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

- ✳ Alternatively, add the following directive to **/etc/foreman/email.yaml** under `smtp_settings`:

```
enable_starttls_auto: :false
```

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

```
production:
  email_delivery:
    delivery_method: :sendmail
    sendmail_settings:
      arguments: "-i -t -G"
```

You can use the **arguments** directive to pass command-line options to **sendmail**, default value of **arguments** is "-i -t". For more information see the `sendmail(1)` man page.

**Important**

After updating the **/etc/foreman/email.yaml** file, run the following command to apply the changes:

```
# katello-service restart
```

You can set the additional email settings, such as the reply address or subject prefix, in Satellite GUI at **Administer** → **Settings** under the **General** tab.

17.1.5. Removing a User

The following procedure describes how to remove an existing user.

Procedure 17.5. To Remove a User:

1. On the main menu, click **Administer** → **Users** to open the **Users** page.
2. Click the **Delete** link to the right of the username you want to delete.

3. In the alert box, click **OK** to delete the user.

17.2. Creating User Groups

With Red Hat Satellite, you can assign permissions to groups of users. You can also create user groups as collections of other user groups. If using an external authentication source, you can map Satellite user groups to external user groups as described in [Section 17.2.1, “Configuring External User Groups”](#).

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

Procedure 17.6. To Create a User Group:

1. Navigate to **Administer** → **User groups** to view the user groups on your Satellite.
2. Click **New User Group**.
3. On the **User group** tab, specify the name of the new user group and select group members from the list of users. To include a previously-created user group, select the check box next to the name of the group to be added.
4. On the **Roles** tab, select the roles you want to assign to the user group. Alternatively, select the **Administrator** check box to assign all available permissions.
5. Click **Submit** to create the user group.

17.2.1. Configuring External User Groups

Users authenticated through external sources are automatically created on the Satellite server the first time they log in. This does not apply to external user groups that must be mapped to user groups created manually in the Satellite GUI. Members of the external user group then automatically become members of the Satellite user group and receive the associated permissions.

Prerequisites

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

- ✦ If using an LDAP source, make sure the LDAP authentication is correctly configured. Navigate to **Administer** → **LDAP Authentication** to view and modify the existing sources. For instructions on how to create an LDAP source, see [Section 20.1, “Using LDAP”](#). Take note of the LDAP group names you want to use.
- ✦ If your Satellite is enrolled with the IdM/IPA or AD server as described in [Chapter 20, Configuring External Authentication](#), take note of the external group names you want to use. To find the group membership of external users, execute the **id** command on Satellite:

```
# id username
```

Here, *username* is the name of the external group member. Note that Satellite allows you to configure external groups only after at least one external user authenticates for the first time. Also, at least one user must exist in the external authentication source.

Procedure 17.7. To Configure an External User Group:

1. Navigate to **Administer** → **User Groups**. Click **New User Group**.
2. On the **User group** tab, specify the name of the new user group. Do not select any users as they will be added automatically when refreshing the external user group.
3. On the **Roles** tab, select the roles you want to assign to the user group. Alternatively, select the **Administrator** check box to assign all available permissions.
4. On the **External groups** tab, click **Add external user group** and select the authentication source:
 - A. If using an LDAP source, select its name from the **Auth source** drop-down menu.
 - B. If using IdM/IPA or AD, select EXTERNAL from the **Auth source** drop-down menu.

Specify the exact name of the LDAP or external group in the **Name** field.
5. Click **Submit**.



Important

LDAP user groups are refreshed automatically through a scheduled task (cron job) synchronizing the LDAP Authentication source (every 30 minutes by default). If the user groups in the LDAP Authentication source change in the lapse of time between scheduled tasks, the user can be assigned to incorrect external user groups. This is corrected automatically when the scheduled task runs. You can also refresh the LDAP source manually by executing **foreman-rake ldap:refresh_usergroups** or by refreshing the external user groups through the web interface.

External user groups based on IdM/IPA or AD are refreshed only when a group member logs in to Satellite. It is not possible to alter user membership of external user groups in the Satellite GUI, such changes are overwritten on the next group refresh. To assign additional permissions to an external user, add this user to an internal user group that has no external mapping specified. Then assign the required roles to this group.

17.3. Creating and Managing Roles

Red Hat Satellite provides a set of predefined roles with permissions sufficient for standard tasks, as listed in [Table 17.1, “Predefined Roles Available in Red Hat 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.

Table 17.1. Predefined Roles Available in Red Hat Satellite

Role	Permissions Provided by Role [a]
Anonymous	The set of permissions that every user is granted, irrespective of any other roles.
Discovery manager	View, provision, edit, and destroy discovered hosts and manage discovery rules.
Discovery reader	View hosts and discovery rules.
Boot disk access	Download the boot disk.

Role	Permissions Provided by Role
Red Hat Access Logs	View the log viewer and the logs.
Manager	A most extensive set of permissions, the majority of actions from each resource type is enabled.
Edit partition tables	View, create, edit and destroy partition tables.
View hosts	View hosts.
Edit hosts	View, create, edit, destroy, and build hosts.
Viewer	A passive role that provides the ability to view the configuration of every element of the Satellite structure, logs, and statistics.
Site manager	A restrained version of the Manager role.
Tasks manager	View and edit Satellite tasks.
Tasks reader	View Satellite tasks.

[a] The exact set of allowed actions associated with predefined roles can be viewed by the privileged user as described in [Section 17.3.4, “Viewing Permissions of a Role”](#).

17.3.1. Creating a Role

The following steps show how to create a role.

Procedure 17.8. To Create a Role:

1. Navigate to **Administer** → **Roles**.
2. Click **New Role**.
3. Provide a **Name** for the role.
4. Click **Submit** to save your new role.

To serve its purpose, a role must contain permissions. After creating a role, proceed to [Section 17.3.3, “Adding Permissions to a Role”](#).

17.3.2. Cloning a Role

Cloning an existing role is a time-saving method of role creation, especially if you want to create a new role that is a variation of an existing permission set. The following procedure explains how to clone a role.

1. Navigate to **Administer** → **Roles**
2. Select **Clone** from the drop-down list to the right of the role to be copied.
3. Select the name for the new role and alter the permissions as needed.

17.3.3. Adding Permissions to a Role

The following steps show how to add permissions to a role.

Procedure 17.9. To Add Permissions to a Role:

1. Navigate to **Administer** → **Roles**.
2. Select **Add Permission** from the drop-down list to the right of the required role.

3. Select the **Resource type** from the drop-down list.



Note

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. Select whether the permission is **Unlimited**.
6. To search for a particular role, use the **Search** field at the top of the list of roles. See [Section 17.4, “Granular Permission Filtering”](#) for the details of using filtering during these searches.
7. Click **Next**.
8. Click **Submit** to save changes.

17.3.4. Viewing Permissions of a Role

The following procedure shows how to view permissions assigned to an existing role.

Procedure 17.10. To View Permissions Associated with a Role:

1. 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. See [Procedure 17.11, “To Create a Complete Permission Table:”](#) for instructions.

17.3.5. Creating a Complete Permission Table

The following procedure explains how to generate a table of all the permissions available in your installation of Satellite. This procedure produces an exhaustive list of the permissions in the Satellite system, and is the best method of generating a reference of procedures for your installation of Satellite.

Procedure 17.11. To Create a Complete Permission Table:

1. Ensure that the required packages are installed:

```
# yum install ruby193-rubygem-foreman*
```

2. Start the Satellite console with the following command:

```
# foreman-rake console
```

3. Insert the following code into the console:

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

```

result = Foreman::AccessControl.permissions.sort {|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.

4. 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>

```

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

```

</table>

```

6. Open `/tmp/table.html` in a web browser to view the table.

17.3.6. Removing a Role

The following steps show how to remove an existing role.

Procedure 17.12. To Remove a Role:

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

17.4. Granular Permission Filtering

As mentioned in [Section 17.3.3, “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.

To create a granular filter, specify a query in the **Search** field on the **Edit Filter** page. Deselect the **Unlimited** check box for the field to be active. Queries have the following form:

```

field_name operator value

```

Where:

- » *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 [Table 17.2, “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, it is also possible to combine the queries by using the *and* and *or* operators.

Table 17.2. Supported Operators for Granular Search

Operator	Description
=	<i>Is equal to.</i> An equality comparison that is case-sensitive for text fields.
!=	<i>Is not equal to.</i> An inversion of the = operator.
~	<i>Like.</i> A case-insensitive occurrence search for text fields.
!~	<i>Not like.</i> An inversion of the ~ operator.
^	<i>Starts with.</i> A case-insensitive search for text fields starting with a certain string.
!^	<i>Does not start with.</i> An inversion of the ^ operator.
>, >=	<i>Greater than, greater than or equal to</i> Supported for numerical fields only.
<, <=	<i>Less than, less than or equal to</i> Supported for numerical fields only.

For example, the following query applies any permissions specified for the Host/managed resource type only to hosts in the group named host-editors.

```
hostgroup = host-editors
```

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

```
Dev
```

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



Note

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 will not prevent the user from assigning a custom name to the newly created location.

You can limit user permissions to a certain organization or location with use of the permission filter. However, resource types provide a GUI alternative in form of **Locations** and **Organizations** tabs. On these tabs, you can select from the list of available organizations and locations. See [Example 17.1, “Creating an Organization-specific Manager Role”](#).

Example 17.1. Creating an Organization-specific Manager Role

This example shows how to create a manager role restricted to a single organization named *org-1*.

1. Navigate to **Administer** → **Roles**.
2. Clone the existing *Manager* 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 Manager*.
3. Click **Filters** next to *org-1 Manager* to view the filters associated with the role. All filters are marked as unlimited.
4. For each filter, click **Edit**.
5. If the filter contains the **Organizations** tab, navigate to it. Otherwise it is a global setting that cannot be limited.
6. On the **Organizations** tab, select **org-1**. Click **Submit**.
7. The restricted filters are no longer marked as unlimited. Users assigned with the *org-1 Manager* role can now perform management tasks only in the selected organization.

Chapter 18. Backup and Disaster Recovery

This chapter describes the minimum and typical backup and restore procedures required to ensure continuity of your Red Hat Satellite deployment and associated data in the event of a disaster. If your deployment uses custom configurations you should take these into account when planning your backup and disaster recovery policy.

18.1. Backing up Red Hat Satellite Server

This section describes the process required to create a complete backup of your Satellite Server and all associated data.

Procedure 18.1. To Back up Your Red Hat Satellite Server:

1. Ensure your backup location has enough disk space to contain a copy of all of the following directories:

- ✧ `/etc/`
- ✧ `/var/lib/pulp`
- ✧ `/var/lib/mongodb`
- ✧ `/var/lib/pgsql/`

This can be a considerable amount of space so plan accordingly.

2. Run the backup script:

```
# /usr/bin/katello-backup backup_directory
```

The **katello-backup** script stops all services which could impact the backup, performs the backup, then restarts the required services.

This process can take a long time to complete, due to the amount of data to copy.

18.2. Restoring Red Hat Satellite Server from a Backup

This section describes how to fully restore a Red Hat Satellite Server from the backup data created as a result of following the steps in [Section 18.1, “Backing up Red Hat Satellite Server”](#). This process restores the backup on the same server that generated the backup. If the original system is unavailable, provision the same configuration with the same settings (host name, IP address, and so on).



Important

The following process describes a full Red Hat Satellite restoration. This process deletes all data from the target Satellite instance. Ensure that you address the following conditions:

- ❖ You are restoring to the correct instance. The Red Hat Satellite instance must have the same configuration, package versions and errata as the original system.
- ❖ All commands are executed as **root** in the directory where the archives were created during the backup process.
- ❖ All SELinux contexts are correct. Run the following command to restore the correct SELinux contexts:

```
# restorecon -Rnv /
```

Procedure 18.2. To Restore Red Hat Satellite from Backup:

1. Install Satellite 6 using the procedures in the [Red Hat Satellite 6 Installation Guide](#)^[6].
2. Copy the backup data to the Satellite's local file system, for example, **/var/tmp/satellite-backup/**. Ensure you have enough space to store this data on the Satellite server as well as enough space after the restoration to contain all the data in the **/etc/** and **/var/** directories contained within the backup.
3. Run the restoration script:

```
# /usr/bin/katello-restore backup_directory
```

This process can take a long time to complete, due to the amount of data to copy.

When this process completes, all services should be running and the Satellite Server should be available for use.

[6] https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.1/html-single/Installation_Guide/index.html

Chapter 19. Maintaining a Red Hat Satellite Server

This chapter provides information on how to maintain a Red Hat Satellite Server, including information on relevant log files, and how to view the import history.

19.1. Logging and Reporting

Red Hat Satellite provides system information in the form of notifications and log files. Examples of log files for troubleshooting are listed in [Table 19.1, “Log Files for Reporting and Troubleshooting”](#):

Table 19.1. Log Files for Reporting and Troubleshooting

Log File	Description of Log File Content
<code>/var/log/elasticsearch</code>	Web UI search index display
<code>/var/log/candlepin</code>	Subscription management
<code>/var/log/foreman</code>	Foreman
<code>/var/log/foreman-proxy</code>	Foreman proxy
<code>/var/log/httpd</code>	Apache HTTP server
<code>/var/log/katello-installer</code>	Satellite installer
<code>/var/log/capsule-installer</code>	Capsule installer
<code>/var/log/libvirt</code>	Virtualization API
<code>/var/log/mongodb</code>	Satellite database
<code>/var/log/pulp</code>	Celerybeat and Celery startup request messages. After startup is complete, messages are logged to <code>/var/log/messages</code> .
<code>/var/log/puppet</code>	Configuration management
<code>/var/log/rhsm</code>	Subscription management
<code>/var/log/tomcat6</code> and <code>/var/log/tomcat</code>	Apache web server messages for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7, respectively.
<code>/var/log/messages</code>	Various other log messages related to pulp, rhsm, and goferd.

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

On Red Hat Enterprise Linux 7, you can use the *journal* for more extensive logging information. See [Using the Journal](#) ^[7] for more information.

19.2. Collecting Information from Log Files

You can use the **foreman-debug** command to collect configuration and log file data for Red Hat Satellite, its back-end services, and system information. This information is collected and written to a tar file. You can also generate reports to view and monitor information about the hosts being maintained.



Important

The **foreman-debug** command removes all security information such as passwords, tokens, and keys while collecting information. However, the tar file can still contain sensitive information about the Red Hat Satellite Server. Red Hat recommends that you send this information directly to the intended recipient and not to a public target.

19.3. Enabling Debug Logging

This section describes how to enable *debug logging* to provide detailed debugging information for the major components of Satellite 6. Debug logging provides the most detailed log information and can help with troubleshooting issues that may arise with Satellite 6 and its components. Other logging levels include **WARN**, **INFO**, and **Error**. Different components provide varying levels of logging.

Foreman and Katello

To enable debug logging for these components, modify the `/usr/share/foreman/config/environments/production.rb` file to ensure the following line exists:

```
config.log_level = :debug
```

Restart the required services:

```
# service foreman restart
# service foreman-tasks restart
```

You should now see more verbose messages in the `/var/log/foreman/production.log` file.

Puppet

See <https://docs.puppetlabs.com/references/latest/configuration.html#loglevel> for information on how to enable debug logging for Puppet. The Puppet log files are saved to the `/var/log/puppet/logs` directory.

Pulp

See <https://pulp.readthedocs.org/en/latest/user-guide/troubleshooting.html> for initial information on how to enable logging for Pulp.

Procedure 19.1. To Ensure Successful Debug Logging for Pulp:

1. Ensure that **rsyslog** allows debug log files to be written to `/var/log/messages`, or redirect the log files to another directory.
2. In the `/etc/pulp/server.conf` file, change the following line:

```
# log_level: INFO

to:

log_level: DEBUG
```

3. Restart the required services:

```
# for i in pulp_resource_manager pulp_workers pulp_celerybeat; do
service $i restart; done
```

**Note**

If you set the Pulp logging level to **Debug** and you are using **rsyslog**, you might encounter a situation where many log entries are discarded and missed. If this occurs, create a new log configuration file as follows:

```
# vi /etc/rsyslog.d/pulp.conf
:programname, startswith, "pulp" -/var/log/pulp.log
& ~
```

Save the file and then restart the required services:

```
# service rsyslog restart
# for i in pulp_resource_manager pulp_workers pulp_celerybeat; do
service $i restart; done
```

Inspect the contents of the **/var/log/pulp.log** file for debug output.

Candlepin

To enable debug logging for Candlepin, add the following line to the **/etc/candlepin/candlepin.conf** file:

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

Restart the required services: on Red Hat Enterprise Linux 6, the service is called **tomcat6**; on Red Hat Enterprise Linux 7, the service is called **tomcat**.

```
# service tomcat6 restart
```

You should now see more verbose messages in the **/var/log/candlepin/candlepin.log** file.

Capsule

To enable debug logging for Capsule, uncomment the **DEBUG** line in the **/etc/foreman-proxy/settings.yml** file:

```
# WARN, DEBUG, Error, Fatal, INFO, UNKNOWN
:log_level: DEBUG
```

Restart the **foreman-proxy** service:

```
# service foreman-proxy restart
```

The log files are saved to the `/var/log/foreman-proxy/proxy.log` file.

Hammer

To enable debug logging for **hammer**, comment out the `log_level` entry in the `/etc/hammer/cli_config.yml` file, as follows:

```
# :log_level: 'error' `
```

The log files are saved to the `~/.foreman/log/hammer.log` file. You can configure the log file directory in the `cli_config.yml` file.

19.4. Using Log Files in Support Cases

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

The **sosreport** command is a tool that collects configuration and diagnostic information from a Red Hat Enterprise Linux system, such as the running kernel version, loaded modules, and system and service configuration files. The command also runs external programs to collect further information, and stores this output in the resulting archive.

See the following articles for more information about using **sosreport** and raising support cases:

- <https://access.redhat.com/solutions/3592>: *What is a sosreport and how can I create one?*
- <https://access.redhat.com/articles/38363>: *How to open and manage a support case on the Customer Portal*
- <https://access.redhat.com/articles/445443>: *Red Hat Access: the Red Hat Support Tool*

For more information about Customer Portal services, see [Section 21.1, “Accessing Customer Portal Services from Red Hat Satellite”](#).

[7] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/s1-Using_the_Journal.html

Chapter 20. Configuring External Authentication

By using external authentication you can derive user and user group permissions from user group membership in an external identity provider. Therefore, you do not have to create these users and maintain their group membership manually on the Satellite server. Red Hat Satellite supports four general scenarios for configuring external authentication:

1. Using *Lightweight Directory Access Protocol* (LDAP) server as an external identity provider. LDAP is a set of open protocols used to access centrally stored information over a network. For more information see [Section 20.1, “Using LDAP”](#).
2. Using *Identity Management* (IdM) or *Identity, Policy, Audit* (IPA) server as an external identity provider. IdM and IPA deal with the management of individual identities, their credentials and privileges used in a networking environment. For more information see [Section 20.2, “Using Identity Management”](#).
3. Using *Active Directory* (AD) integrated with IdM or IPA through cross-forest Kerberos trust as an external identity provider. For more information see [Section 20.3, “Using Active Directory with Cross-Forest Trust”](#).
4. Using direct AD as an external identity provider. For more information see [Section 20.4, “Using Active Directory Directly”](#).

The above scenarios are about providing access to the Satellite server. In addition, hosts provisioned with Satellite can also be integrated with IdM/IPA realms. Red Hat Satellite has a realm feature that will automatically manage the life cycle of any system registered to a realm or domain provider. See [Section 20.5, “External Authentication for Provisioned Hosts”](#) for more information.

20.1. Using LDAP

Red Hat Satellite includes the option to use a Lightweight Directory Access Protocol (LDAP) service for user information and authentication, using one or more LDAP directories. See [Red Hat Enterprise Linux System Administrator's Guide](#) ^[8] for more information on LDAP.

You can also use LDAP to connect to an IdM/IPA or AD server, however this is not recommended as LDAP does not support server discovery or cross-forest trusts.

20.1.1. Configure TLS for Secure LDAP (LDAPS)

If you require Red Hat Satellite to use **TLS** to establish a secure LDAP connection (LDAPS), first obtain certificates used by the LDAP server you are connecting to and mark them as trusted on the base operating system of your Satellite server as described below. If your LDAP server uses a certificate chain with intermediate certificate authorities, all of the root and intermediate certificates in the chain must be trusted, so ensure all certificates are obtained. If you do not require secure LDAP at this time, proceed to [Procedure 20.1, “To Configure LDAP Authentication”](#).

Obtain the Certificate from the LDAP Server

If you use Active Directory Certificate Services, export the Enterprise PKI CA Certificate using the Base-64 encoded X.509 format. See [How to configure Active Directory authentication with TLS on Satellite 6.1](#) for information on creating and exporting a CA certificate from an Active Directory server.

Download the LDAP server certificate to a temporary location on the Red Hat Enterprise Linux system where the Satellite server is installed and remove it when finished. For example, `/tmp/example.crt`. The filename extensions `.cer` and `.crt` are only conventions and can refer to DER binary or PEM ASCII format certificates.

Trust the Certificate from the LDAP Server

Red Hat Satellite Server requires the CA certificates for LDAP authentication to be individual files in `/etc/pki/tls/certs/` directory.

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

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

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

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

Restart the **httpd** service:

- » On Red Hat Enterprise Linux 6:

```
# service httpd restart
```

- » On Red Hat Enterprise Linux 7:

```
# systemctl restart httpd
```

20.1.2. Configuring Red Hat Satellite to Use LDAP



Note

SELinux can prevent outgoing LDAP connections. Execute the following command to allow them:

```
# setsebool authlogin_nsswitch_use_ldap=1
```

Procedure 20.1. To Configure LDAP Authentication

Follow this procedure to configure LDAP authentication using the web UI. See the tables below this procedure for examples of the parameter format relevant to the LDAP server being used.

1. Navigate to **Administer** → **LDAP Authentication**.
2. Click **New authentication source**.
3. On the **LDAP server** tab, enter the LDAP server's name, host name, port, and server type. The default port is 389, the default server type is POSIX (alternatively you can

select FreeIPA or Active Directory depending on the type of authentication server). For **TLS** encrypted connections, select the **LDAPS** check box to enable encryption. The port should change to 636, which is the default for LDAPS.

4. On the **Account** tab, enter the following information:

- ✳ **Account username:** an LDAP user who has read access to the LDAP server. User name is not required if the server allows anonymous reading, otherwise use the full path to the user's object. For example:

```
uid=$login,cn=users,cn=accounts,dc=example,dc=com
```

- ✳ **Account password:** the LDAP password for the user defined in the **Account username** field. This field can remain blank if the **Account username** is using the "\$login" variable.

- ✳ **Base DN:** the top level domain name of your LDAP directory. For example:

```
dc=redhat,dc=com
```

- ✳ **Groups base DN:** the top level domain name of your LDAP directory tree that contains groups.
- ✳ **LDAP filter:** a filter to restrict your LDAP queries.
- ✳ **Automatically create accounts in Foreman:** creates Satellite accounts automatically for LDAP users who log in for the first time in Satellite.

5. On the **Attribute mappings** tab, map LDAP attributes to Satellite attributes. You can map Login name, First name, Surname, Email address, and Photo attributes.

6. Click **Submit**.

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

Table 20.1. Example Settings for Active Directory LDAP Connection

Setting	Example value
Account username	DOMAIN\redhat
Base DN	DC=example,DC=COM
Groups Base DN	CN=Users,DC=example,DC=com
Login name attribute	sAMAccountName

Table 20.2. Example settings for FreeIPA LDAP Connection

Setting	Example value
Account username	uid=redhat,cn=users,cn=accounts,dc=example,dc=com
Base DN	dc=example,dc=com
Groups Base DN	cn=groups,cn=accounts,dc=example,dc=com
Login name attribute	uid

Table 20.3. Example Settings for POSIX (OpenLDAP) LDAP Connection

Setting	Example value
Account username	uid=redhat,dc=example,dc=com
Base DN	dc=example,dc=com
Groups Base DN	dc=example,dc=com
Login name attribute	uid

20.2. Using Identity Management

This section shows how to integrate Red Hat Satellite server with an IdM or IPA server and how to enable host-based access control.

Prerequisites

- ✧ The Satellite server has to run on Red Hat Enterprise Linux 7.1 or Red Hat Enterprise Linux 6.6 or later.
- ✧ The base operating system of the Satellite server has to be IPA-enrolled. Ask the IdM/IPA administrator of your organization to perform the following steps on the IdM/IPA server:
 - ✧ Create a host entry for the Satellite server with the **ipa host-add** command. Generate a one-time password with the **--random** option. This password will be used on the client to complete IPA-enrollment. For more information on host configuration properties, see [Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide](#) [9].
 - ✧ Create an HTTP service for the Satellite server with the **ipa service-add HTTP/satellite_fqdn** command. For more information on managing services, see [Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide](#) [10].

The examples in this chapter assume separation between IdM/IPA and Satellite configuration. However, if you have administrator privileges for both servers, you can configure IPA-enrollment as described in [Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide](#) [11].

Procedure 20.2. To Configure IdM/IPA Authentication:

1. Execute the following command as root to configure IPA-enrollment on the client:

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

Replace *OTP* with the one-time password provided by the IdM/IPA administrator.

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

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

The installer is dependent on packages which, on Red Hat Enterprise Linux 7, are in the optional repository **rhel-7-server-optional-rpms**. On Red Hat Enterprise Linux 6 all necessary packages are in the **base** repository.

3. Execute the following command:

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

This command is not limited to a fresh Satellite installation; you can use it to modify an existing Satellite installation.

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

20.2.1. Host Based Access Control Configuration

Host-based access control (HBAC) rules define which machine within the domain an IPA user is allowed to access. You can configure HBAC on the IPA server to prevent selected users from accessing the Satellite server. With this approach, you can prevent Satellite from creating database entries for users that are not allowed to log in. For more information on HBAC, see the [Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide](#)^[12]

Procedure 20.3. To Configure HBAC:

1. Create HBAC service and rule on the IdM/IPA server and link them together. The following examples use the PAM service name *satellite-prod*. Execute the following commands on the IdM/IPA server:

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

2. Add the user who is to have access to the service *satellite-prod*, and the hostname of the Satellite server:

```
$ ipa hbacrule-add-user allow_satellite_prod --user=username
$ ipa hbacrule-add-host allow_satellite_prod --hosts=the-satellite-fqdn
```

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

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

```
$ ipa hbacrule-find satellite-prod
$ ipa hbactest --user=username --host=the-satellite-fqdn --service=satellite-prod
```

4. Ensure the `allow_all` rule is disabled on the IdM/IPA server. For instructions on how to do so without disrupting other services see the [How to configure HBAC rules in IPA](#) article on the Red Hat Customer Portal ^[13].
5. Configure the IdM/IPA integration with the Satellite server as described in [Procedure 20.2, “To Configure IdM/IPA Authentication:”](#). On the Satellite server, define the PAM service as root:

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

20.3. Using Active Directory with Cross-Forest Trust

Kerberos can create *cross-forest trust* that defines a relationship between two otherwise separate domain forests. A domain forest is a hierarchical structure of domains; both AD and IdM/IPA constitute a forest. With a trust relationship enabled between AD and IdM/IPA, users of AD can access Linux hosts and services using a single set of credentials. For more information on cross-forest trusts, see [Red Hat Enterprise Linux Windows Integration Guide](#) ^[14].

From the Satellite point of view, the configuration process is the same as integration with IdM/IPA server without cross-forest trust configured. The Satellite server has to be IPA-enrolled and integrated as described in [Section 20.2, “Using Identity Management”](#). On the IdM/IPA server, the following additional steps are required:

1. To enable the HBAC feature, create an external group and add the AD group to it. Add the new external group to a POSIX group. Use this POSIX group in a HBAC rule.
2. Configure `sssd` to transfer additional attributes of AD users. Add these attributes to the `nss` and `domain` sections in `/etc/sss/sss.conf`. For example:

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

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

20.4. Using Active Directory Directly

This section shows how to use direct Active Directory (AD) as an external authentication source for the Satellite server. Direct AD integration means that the Satellite server is joined directly to the AD domain where the identity is stored. The recommended setup consists of two steps: first enroll Satellite with AD as described in [Procedure 20.4, “To Enroll the Satellite Server with the AD Server:”](#), then finalize the AD integration with use of GSS-proxy as described in [Procedure 20.5, “To Configure Direct AD Integration with GSS-proxy:”](#)

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



Note

The AD integration requires the Red Hat Satellite server to be deployed on Red Hat Enterprise Linux 7.1.

Perform the following procedures on Red Hat Enterprise Linux that acts as a base operating system for your Satellite server. For the examples in this section *EXAMPLE.ORG* is the Kerberos realm for the AD domain.

Prerequisites

Ensure GSS-proxy is installed:

```
# yum install gssproxy
```

Procedure 20.4. To Enroll the Satellite Server with the AD Server:

1. Install the required packages:

```
# yum install sssd adcli realmd ipa-python
```

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

```
# realm join -v EXAMPLE.ORG
```

After enrolling Satellite with the AD server, you can configure the direct AD integration with GSS-proxy using the **katello-installer** command. This can be done for already installed Satellite or during the Satellite installation. Note that the Apache user must not have access to the keytab file. Also take note of the effective user ID of the Apache user (that can be found by executing **id apache**). The following procedure uses the example UID48.

Procedure 20.5. To Configure Direct AD Integration with GSS-proxy:

1. The **katello-installer** command is by default set for the IdM/IPA integration. Change this setting by creating the **/etc/ipa/default.conf** file with the following content:

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

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

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

3. Create a keytab file for HTTP using the following command:

```
# KRB5_KTNAME=FILE:/etc/gssproxy/http.keytab net ads keytab add HTTP -
U administrator -d3 -s /etc/net-keytab.conf
```

This command fetches the HTTP service keytab file from the AD server and stores it at **/etc/gssproxy/http.keytab**. Make sure this file is owned by the root user and group:

```
# chown root:root /etc/gssproxy/http.keytab
```

4. Insert the following line at the beginning of the **/etc/krb5.conf** file:

```
includedir /var/lib/sss/pubconf/krb5.include.d/
```

5. Create an empty keytab file at **/etc/httpd/conf/http.keytab**:

```
# touch /etc/httpd/conf/http.keytab
```

6. Execute the following command:

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

7. Place the following text at the beginning of the **/etc/gssproxy/gssproxy.conf** configuration file:

```
[service/HTTP]
  mechs = krb5
  cred_store = keytab:/etc/gssproxy/http.keytab
  cred_store = ccache:/var/lib/gssproxy/clients/krb5cc_%U
  euid = 48
```

Here, 48 is the effective UID of the Apache user. This text must precede any section containing the **allow_any_uid=yes** directive, therefore place it before the **[service/nfs-client]** section in the file.

8. Create a separate cache directory for Kerberos to avoid AVC denials:

```
# mkdir /var/lib/gssproxy/rcache
```

To configure the **gssproxy** service to use the cache, create the **/etc/systemd/system/gssproxy.service** file and insert the following text:

```
.include /usr/lib/systemd/system/gssproxy.service
[Service]
Environment=KRB5RCACHEDIR=/var/lib/gssproxy/rcache
```

Apply changes to the service:

```
# systemctl daemon-reload
```

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

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

10. Configure the Apache server to use GSS-proxy by creating the `/etc/systemd/system/httpd.service` file with the following content:

```
.include /lib/systemd/system/httpd.service
[Service]
Environment=GSS_USE_PROXY=1
```

Apply changes to the service:

```
# systemctl daemon-reload
```

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

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

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

By completing the above procedure you allow users that belong to the EXAMPLE.ORG realm to log in to the Satellite server. Users can configure Kerberos SSO in their browsers to be able to log in without filling in access credentials in the Satellite GUI. For more information on configuring the Firefox browser see the [Red Hat Enterprise Linux System-Level Authentication Guide](#). Users of the Internet Explorer browser have to add the Satellite server to the list of Local Intranet or Trusted sites, and turn on the *Enable Integrated Windows Authentication* setting. See the Internet Explorer documentation for details.



Note

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

```
access_provider = ad
ad_gpo_access_control = enforcing
ad_gpo_map_service = +satellite-prod
```

Here, *satellite-prod* is the PAM service name. For more information on GPOs, please refer to the [Red Hat Enterprise Linux Windows Integration Guide](#)^[15].

20.5. External Authentication for Provisioned Hosts

This section shows how to configure IdM integration to authenticate provisioned hosts. First configure the Satellite or Capsule server for IdM realm support, then add hosts to the IdM realm group.

20.5.1. Configuring a Red Hat Satellite Server or Capsule Server for IdM Realm Support

To use IdM for provisioned hosts, first configure the Red Hat Satellite Server or Red Hat Satellite Capsule Server.

Prerequisites

1. A Satellite Server is registered to the content delivery network, an independent Capsule Server is registered to the Satellite Server.
2. A realm or domain provider such as Red Hat Identity Management is configured.

Procedure 20.6. To configure the Satellite Server or Capsule Server for IdM Realm Support:

1. On the Satellite Server or Capsule Server, install the following packages:

```
# yum install ipa-client foreman-proxy ipa-admintools
```

2. Configure the Satellite Server (or Capsule Server) as an IPA client:

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

3. Create a realm-capsule user and the relevant roles in Red Hat Identity Management on the Satellite Server or Capsule Server:

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

Running `foreman-prepare-realm` will prepare an IPA or IdM server for use with the Capsule. It creates a dedicated role with the permissions needed for Satellite, creates a user with that role and retrieves the keytab file. You will need your Identity Management server configuration details on this step.

If the command successfully executes, you should be able to see the following command output:

```
Keytab successfully retrieved and stored in: freeipa.keytab
Realm Proxy User:    realm-capsule
Realm Proxy Keytab:  /root/freeipa.keytab
```

4. Move the `/root/freeipa.keytab` to the `/etc/foreman-proxy` directory and set the ownership settings to the user `foreman-proxy`:

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

5. Configure the realm based on whether you are using Satellite Server or Capsule Server:
 - A. If you are using the integrated capsule in the Satellite Server, use **katello-installer** to configure the realm:

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



Note

You may also run these options when you first configure the Red Hat Satellite Server.

- B. If you are using an independent Capsule Server, use **capsule-installer** to configure the realm:

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

6. Make sure that the most updated versions of the ca-certificates package is installed and trust the IPA Certificate Authority:

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

7. (Optional) If you are configuring IdM on an already existing Satellite Server or Capsule Server, the following steps should also be taken to make sure that the configuration changes take effect:

- a. Restart the foreman-proxy service:

```
# service foreman-proxy restart
```

- b. Log in to the Satellite Server and click **Infrastructure → Capsules**.
- c. Click on the drop-down menu on the right-hand side of the Capsule Server you have configured for IdM and choose **Refresh Features**.

8. Finally, create a new realm entry in the Satellite Server user interface:

- a. Click **Infrastructure → Realms** and on the right-hand corner of the main page, click **New Realm**.
- b. Fill in the fields in the following subtabs:
- On the **Realm** subtab, provide the realm name, the type of realm to use and the realm proxy.
 - On the **Locations** subtab, choose the locations where the new realm is intended for use.
 - On the **Organizations** subtab, choose the organizations where the new realm is intended for use.
- c. Click **Submit**.

The Satellite Server or Capsule Server is now ready to provision hosts that automatically register to IdM. The next section will detail the steps on how to automatically add hosts to an IdM host group.

20.5.2. Adding Hosts to an IdM Host Group

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

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



Note

The Satellite Server or Capsule Server sends updates to the IdM server, however automembership rules are only applied at initial registration.

Procedure 20.7. To Add Hosts to an IdM Host Group:

1. On the IdM server, create a host group:

```
# ipa hostgroup-add hostgroup_name
Description: hostgroup_description
-----
Added hostgroup "hostgroup_name"
-----
Host-group: hostgroup_name
Description: hostgroup_description
```

Where:

- a. *hostgroup_name* is the host group's name.
- b. *hostgroup_description* is the host group's description.

2. Create an automembership rule:

```
# ipa automember-add --type=hostgroup automember_rule
-----
Added automember rule "automember_rule"
-----
Automember Rule: automember_rule
```

Where:

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

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

```
# ipa automember-add-condition --key=userclass --type=hostgroup --
inclusive-regex=^webserver hostgroup_name
-----
```

```
Added condition(s) to "hostgroup_name"
-----
Automember Rule: automember_rule
Inclusive Regex: userclass=^webserver
-----
Number of conditions added 1
-----
```

Where:

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

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

[8] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Directory_Servers.html

[9] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Linux_Domain_Identity_Authentication_and_Policy_Guide/host-attr.html

[10] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Linux_Domain_Identity_Authentication_and_Policy_Guide/services.html

[11] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Linux_Domain_Identity_Authentication_and_Policy_Guide/linux-manual.html

[12] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Linux_Domain_Identity_Authentication_and_Policy_Guide/configuring-host-access.html

[13] <https://access.redhat.com/solutions/67895>

[14] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Windows_Integration_Guide/active-directory-trust.html

[15] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Windows_Integration_Guide/sssd-gpo.html

Chapter 21. Red Hat Satellite User Interface Plug-ins

21.1. Accessing Customer Portal Services from Red Hat Satellite

The Red Hat Access pre-installed plug-in lets you access several Red Hat Customer Portal services from within the Red Hat Satellite web interface.

The Red Hat Access plug-in provides the following services:

- » **Search:** Search solutions in the Customer Portal from within the Red Hat Satellite interface.
- » **Logs:** Send specific parts (snippets) of the log files to assist in problem solving. Send these log snippets to the Red Hat Customer Portal diagnostic tool chain.
- » **Support:** Access your open support cases, modify an open support case and open a new support case from within the Red Hat Satellite interface.



Note

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

21.1.1. Searching for Solutions in the Red Hat Access Plug-in

The Red Hat Access plug-in provides search capabilities that look through the solutions database available in the Red Hat Customer Portal without needing to log in to the Customer Portal interface.

Procedure 21.1. To Search for Solutions from the Red Hat Satellite Server:

1. In the upper right, click **Red Hat Access** → **Search**.
2. If necessary, log in to the Red Hat Customer Portal. In the main panel on the upper right, click Log In.



Note

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. In the **Red Hat Search** field, enter your search query. Search results display in the left-hand **Recommendations** list.
4. In the **Recommendations** list, click a solution. The solution article displays in the main panel.

21.1.2. Using Logs in the Red Hat Access Plug-in

The log file viewer lets you view the log files and isolate log snippets. You can also send the log snippets through the Customer Portal diagnostic tool chain to assist with problem solving.

Procedure 21.2. To Use the Logs Diagnostic Tool from the Red Hat Satellite Server:

1. In the upper right, click **Red Hat Access → Logs**.
2. If necessary, log in to the Red Hat Customer Portal. In the main panel on the upper right, click **Log In**.



Note

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. In the left file tree, select a log file and click the file name.
4. Click **Select File**. A pop-up window displays the log file contents.
5. In the log file, highlight any text sections you want diagnosed. The **Red Hat Diagnose** button displays.
6. Click **Red Hat Diagnose**. The system sends the highlighted information to the Red Hat Customer Portal, and provides solutions that closely match the provided log information.
7. If a solution does the following:
 - ✦ Matches the problem, click the solution and follow the required steps to troubleshoot the issue.
 - ✦ Does not match the problem, click **Open a New Support Case**. The support case is populated with the highlighted text from the log file. See [Section 21.1.5, “Creating Support Cases Using the Red Hat Access Plug-in”](#).

21.1.3. Viewing Existing Support Cases Using the Red Hat Access Plug-in

You can view your existing support case from your Red Hat Satellite Server using the Red Hat Access Plug-in.

Procedure 21.3. To View Existing Support Cases from the Red Hat Satellite Server:

1. In the upper right, click **Red Hat Access → Support → My Cases**.
2. If necessary, log in to the Red Hat Customer Portal. In the main panel on the upper right, click **Log In**.



Note

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. To search for a specific support case from existing cases, do any of the following:

- a. In the **Search** field, provide a key word or phrase.
 - b. From the drop-down list, choose a specific **Case Group**. Your organization has defined **Case Groups** inside the Red Hat Customer Portal.
 - c. Choose a Case Status.
4. From the results, choose a specific support case and click the **Case ID**. The support case is ready to view.

21.1.4. Modifying Support Cases Using the Red Hat Access Plug-in

You can update your existing support cases from your Red Hat Satellite Server using the Red Hat Access Plug-in.

Procedure 21.4. To Update Support Cases from the Red Hat Satellite Server Web Interface:

1. Complete the instructions from [Section 21.1.3, “Viewing Existing Support Cases Using the Red Hat Access Plug-in”](#)
2. In the support case, scroll down to the marked sections to do the following:
 - ✳ **Attachments:** - Attach a local file from the system. Add a file name to make it easier to identify.



Note

File names must be less than 80 characters and the maximum file size for attachments uploaded using the web interface is 250 MB. Use FTP for larger files.

- ✳ **Case Discussion:** - Add any updated information about the case you wish to discuss with Global Support Services. After adding information, click **Add Comment**.

21.1.5. Creating Support Cases Using the Red Hat Access Plug-in

You can create a new support case from your Red Hat Satellite Server using the Red Hat Access Plug-in.

Procedure 21.5. To Create a New Support Case using the Red Hat Satellite Server:

1. In the upper right, click **Red Hat Access → Support → New Case**.
2. If necessary, log in to the Red Hat Customer Portal. In the main panel on the upper right, click Log In.



Note

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. The **Product** and **Product Version** fields are automatically populated. Complete the

other relevant fields, as follows:

- ✳ **Summary:** - Provide a brief summary of the issue.
- ✳ **Description:** - Write a detailed description of the issue.

Based on the summary provided, recommendations for possible solutions display in the main panel.

4. Click **Next**.
5. Choose the appropriate options, as follows:
 - ✳ **Severity:** Select the ticket urgency as 4 (low), 3 (normal), 2 (high) or 1 (urgent).
 - ✳ **Case Group:** Based on who needs to be notified, create case groups associated with the support case. Select Case Groups in Red Hat Satellite. Create Case Groups within the Customer Portal.
6. Attach any required files. Add a file description and click **Attach**.

To ensure you provide relevant information, it is recommended that you attach the output of the following commands:

```
# sosreport  
# foreman-debug
```



Important

foreman-debug removes all security information such as password, tokens and keys while collecting information. However, the tarball can still contain sensitive information about the Red Hat Satellite Server. It is recommended to send this information directly to the intended recipient and not publicly.



Note

File names must be less than 80 characters and the maximum file size for attachments uploaded using the web interface is 250 MB. Use FTP for larger files.

7. Click **Submit**. The system uploads the case to the Customer Portal, and provides a case number for your reference.

Chapter 22. Command Line Reference

hammer is the CLI management tool for Red Hat Satellite functionality. It can:

- ✱ Provision hosts.
- ✱ Edit the attributes of a resource or group.
- ✱ Interact and manipulate hosts, capsules and domains.

hammer can be executed on the command line through its parameters and options or through the interactive shell. To invoke the shell:

Example 22.1. Invoking the hammer Shell

```
[root@satellite.example.com ~]# hammer shell
Welcome to the hammer interactive shell
Type 'help' for usage information
Command completion is disabled on ruby < 1.9 due to compatibility
problems.
hammer> organization list
---|-----|-----|-----
--
ID | NAME           | LABEL           | DESCRIPTION
---|-----|-----|-----
--
1  | ACME_Corporation | ACME_Corporation | ACME_Corporation Organization
3  | Test Corp       | Test_Corp       |
---|-----|-----|-----
--
hammer>
```

The full list of options and subcommands are available in the help file:

```
# hammer -h
```

22.1. Configuring hammer

By default, if you run **hammer** from the command line, you need to enter your credentials for each operation. To avoid this, you can either use **hammer shell** as described in [Example 22.1, “Invoking the hammer Shell”](#) or you can create a configuration file in your home directory with your login credentials.



Important

Saving credentials in plain text files is a potential security risk. Take the necessary precautions to ensure your passwords and other sensitive information are protected.

Procedure 22.1. To Configure hammer to Use Saved Credentials:

1. Create a `~/.hammer/cli_config.yml` file if it does not already exist.
2. Add the following contents to the file. Ensure you replace the example values with your own details.

```
:foreman:  
  :host: 'https://satellite.example.com/'  
  :username: 'admin'  
  :password: 'changeme'
```

3. Save and close the file. Now when you run **hammer** it should not prompt you for your credentials.

Appendix A. Glossary of Terms

The following terms are used throughout this document. Familiarize yourself with these terms to help your understanding of Red Hat Satellite 6.

Activation Key

A registration token used in a Kickstart file to control actions at registration. These are similar to Activation Keys in Red Hat Satellite 5, but provide a subset of features because Puppet controls package and configuration management after registration.

Application Life Cycle Environment

An *Application Life Cycle Environment* represents a step, or stage, in a promotion path through the Software Development Life Cycle (SDLC). Promotion paths are also known as development paths. Content such as packages and Puppet modules move through life cycle environments by publishing and promoting Content Views. All Content Views have versions, which means you can promote a specific version through a typical promotion path; for example, from development to test to production. Channel cloning implements this concept in Red Hat Satellite 5.

Attach

The process of associating a Subscription to a Host that provides access to RPM content.

Capsule

A *Capsule* is an additional server that can be used in a Red Hat Satellite 6 deployment to facilitate content federation and distribution in addition to other localized services (Puppet Master, **DHCP**, **DNS**, **TFTP**, and more).

Catalog

A *Catalog* is a document that describes the desired system state for one specific computer. It lists all of the resources that need to be managed, as well as any dependencies between those resources.

Compute Profile

Compute Profiles specify default attributes for new virtual machines on a compute resource.

Compute Resource

A *Compute Resource* is virtual or cloud infrastructure, which Red Hat Satellite 6 uses for deployment of hosts and systems. Examples include Red Hat Enterprise Virtualization Manager, OpenStack, Amazon EC2, and VMware vSphere.

Content

Content includes software packages (RPM files) and Puppet modules. These are synchronized into the Library and then promoted into Life Cycle Environments using Content Views so that they can be consumed by Hosts.

Content Delivery Network (CDN)

The *Content Delivery Network (CDN)* is the mechanism used to deliver Red Hat content in a geographically co-located fashion. For example, content that is synchronized by a Satellite in Europe pulls content from a source in Europe.

Content Host

A *Content Host* is the part of a host that manages tasks related to content and subscriptions.

Content View

A *Content View* is a definition of content that combines products, packages, and Puppet modules with capabilities for intelligent filtering and creating snapshots. Content Views are a refinement of the combination of channels and cloning from Red Hat Satellite 5.

External Node Classifier

An *External Node Classifier* is a Puppet construct that provides additional data for a Puppet Master to use when configuring Hosts. Red Hat Satellite 6 acts as an External Node Classifier to Puppet Masters in a Satellite deployment.

Facter

Facter is a program that provides information (facts) about the system on which it is run; for example, Facter can report total memory, operating system version, architecture, and more. Puppet modules enable specific configurations based on host data gathered by Facter.

Hammer

Hammer is a command line tool for Red Hat Satellite 6. Use Hammer to manage Red Hat Satellite 6 as a standard CLI, for scripts, and also through an interactive shell.

Hiera

Hiera is a key/value look-up tool for configuration data which allows keeping site-specific data out of puppet manifests.

Host

A *Host* refers to any system, either physical or virtual, that Red Hat Satellite 6 manages.

Host Collection

A *Host Collection* is equivalent to a Satellite 5 *System Group*, that is, a user defined group of one or more Hosts.

Host Group

A *Host Group* is a template for building a Host. This includes the content view (which defines the available RPM files and Puppet modules) and the Puppet classes to apply (which ultimately determines the software and configuration).

Location

A *Location* is collection of default settings that represent a physical place. These can be nested so that you can set up an hierarchical collection of locations. For example, you can set up defaults for "Middle East", which are refined by "Tel Aviv", which are further refined by "Data Center East", and then finally by "Rack 22".

Library

The *Library* contains every version, including the latest synchronized version, of the software that the user will ever deploy. For an Information Technology Infrastructure Library (ITIL) ^[16] organization or department, this is the Definitive Media Library ^[17] (previously named the Definitive Software Library).

Manifest

A *Manifest* transfers subscriptions from the Customer Portal to Red Hat Satellite 6. This is similar in function to certificates used with Red Hat Satellite 5.

For more information about certificates and subscription types, see:

- [RHN Classic, Red Hat Satellite, and Channel Entitlements](#) ^[18]
- [The Structure of Satellite Certificates \(Classic Style of Certificates\)](#) ^[19]

Organization

An *Organization* is an isolated collection of systems, content, and other functionality within a Satellite 6 deployment.

Product

A collection of content repositories. Products can be Red Hat products or newly-created products made up of software and configuration content.

Promote

The act of moving a content view comprised of software and configuration content from one Application Life Cycle Environment to another, such as moving from development to QA to production.

Provisioning Template

A *Provisioning Template* is a user-defined template for Kickstart files, snippets, and other provisioning actions. In Satellite 6 they provide similar functionality to Kickstart Profiles and cobbler Snippets in Red Hat Satellite 5.

Pulp Node

A *Pulp Node* is a Capsule Server component that mirrors content. This is similar to the Red Hat Satellite 5 Proxy. The main difference is that content can be staged on the Pulp Node before it is used by a Host.

Puppet Agent

The *Puppet Agent* is an agent that runs on a Host and applies configuration changes to that Host.

Puppet Master

A *Puppet Master* is a Capsule Server component that provides Puppet manifests to Hosts for execution by the Puppet Agent.

Puppet Module

A *Puppet Module* is a self-contained bundle of code and data that you can use to manage resources such as users, files, and services.

Repository

A *Repository* provides storage for a collection of content. For example, a YUM repository or a Puppet repository.

Role

A *Role* specifies a collection of permissions that are applied to a set of resources, such as Hosts.

Smart Proxy

A *Smart Proxy* is a Capsule Server component that can integrate with external services, such as **DNS** or **DHCP**.

Smart Variable

A *Smart Variable* is a configuration value that controls how a Puppet Class behaves. This can be set on a Host, a Host Group, an Organization, or a Location.

Standard Operating Environment (SOE)

A *Standard Operating Environment (SOE)* is a controlled version of the operating system on which applications are deployed.

Subscription

Subscriptions are the means by which you receive content and service from Red Hat.

Synchronizing

Synchronizing refers to mirroring content from external resources into the Red Hat Satellite 6 Library.

Synchronization Plans

Synchronization Plans provide scheduled execution of content synchronization.

Unattended Mode

In the context of PXE-less discovery, refers to the ability of Red Hat Satellite to initiate the provisioning process with no interaction from the user. For more information, see [Unattended and semi-automatic mode](#) in the [Foreman Discovery Manual](#) [20].

User Group

A *User Group* is a collection of roles which can be assigned to a collection of users. This is similar to a Role in Red Hat Satellite 5.

User

A user is anyone registered to use Red Hat Satellite. Authentication and authorization is possible through built-in logic, through external LDAP resources, or with Kerberos.

[16] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[17] http://en.wikipedia.org/wiki/Definitive_Media_Library

[18] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/MigratingRHN/sat-certs.html

- [19] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/Subscription_Concepts_and_Workflows/index.html#subscription-legacy
- [20] http://theforeman.org/plugins/foreman_discovery/4.1/index.html#1.ForemanDiscovery4.1Manual

Appendix B. Revision History

Revision 2-6	Wed Aug 24 2016	Stephen Wadeley
BZ 1343567: Ad Integration Install packages step requires installing 'gssproxy' package.		
Revision 2-5	Tue Dec 15 2015	Stephen Wadeley
BZ 1275755: Removed duplicate content on Red Hat Satellite Capsule Servers.		
Revision 2-4	Mon Nov 16 2015	Hayley Hudgeons
Building for async 2.		
Revision 2-3	Tue Nov 13 2015	David O'Brien
BZ 1275127: Added a description of 'unattended mode' to the glossary.		
Revision 2-2	Tue Nov 03 2015	David O'Brien
BZ 1277304: Fix incorrect package names in Capsule installation instructions.		
BZ 1271079: Install Guide bug also appeared in this guide.		
Revision 2-1	Mon Oct 12 2015	David O'Brien
BZ 1195777: Update chapter on Discovery with changes and new features.		
BZ 1145773: Added chapter on debugging.		
Revision 2-0	Mon Aug 31 2015	David O'Brien
Relocate CLI reference closer to end of other reference material at end of book.		
BZ 1172415: Add path arguments to backup and restore script examples.		
Revision 1-55	Thurs July 23 2015	Megan Lewis
BZ#1206788 Integrating peer review feedback.		
BZ#1213399 Updated instructions on how to create a manifest.		
Revision 1-54	Wed July 22 2015	Jo Somers
BZ#1245391 Corrections to Step 2 in 13.2 Configuring the Foreman Discovery Plug-in.		
BZ#1236535 Deleted 11.2.2 Creating a Puppet Class.		
BZ#1244503 Fixed typos in Chapter 8 Working with Containers.		
BZ#1158752 Added Step 1 and added info to new Step 3 in section 3.3.1 Creating Lifecycle Environments.		
Revision 1-53	Fri July 17 2015	Megan Lewis
BZ#1216135 Corrections to note in 12.3.1. Registering a Host.		
Revision 1-52	Tues July 14 2015	David O'Brien
Remove draft and beta status.		
Publish for technical review.		
Revision 1-51	Tues July 14 2015	Megan Lewis
BZ#1206788 Further corrections to the Disconnected Satellite section.		
Revision 1-50	Mon July 13 2015	Megan Lewis

BZ#1107485 Made corrections based on peer review feedback.
 BZ#1206788 Restructured and corrected Disconnected Satellite section.
 BZ#1206788 Changed reference in Chapter 7. Viewing and Applying Errata to new Disconnected Satellite section.

Revision 1-49	Thu July 2 2015	Jo Somers
BZ#1171611 Changed section Registering Host Systems to a Red Hat Satellite Capsule Server to match Installation Guide.		
Revision 1-48	Wed July 1 2015	Megan Lewis
BZ#1206788 Corrected error in Step 2 of 4.3.4. Importing Content to a Disconnected Satellite Server.		
Revision 1-47	Tue Jun 30 2015	Megan Lewis
BZ#1107485 Further changes made based on peer review to Chapter 21 Red Hat Satellite User Interface Plug-ins.		
Revision 1-46	Mon Jun 29 2015	Megan Lewis
BZ#1175938 Added requested changes to the Activation Key section.		
BZ#1107485 Changes made based on peer review to Chapter 19 Maintenance.		
BZ#1107485 Partial changes made based on peer review to Chapter 21 Red Hat Satellite User Interface Plug-ins.		
Revision 1-45	Thu Jun 25 2015	Jo Somers
BZ 1234705 Changed channel to repository in sections Synchronization Status and Red Hat Satellite Capsule Server Prerequisites		
BZ 1234705 Changed channel to interface in section Configuring an Additional Network Interface		
Revision 1-44	Mon Jun 22 2015	Megan Lewis
BZ#1132527 Added draft of Using Identity Management for Authentication.		
Revision 1-43	Mon Jun 15 2015	David O'Brien
6.1 Public Beta release.		
Remove 'report a bug' links and related files.		
Revision 1-42	Mon June 8 2015	Jo Somers
BZ#1222882 Changed 12.2.1. subsection into a procedure under 12.2; last step in prerequisites added for RHEL7: # systemctl start chronyd; systemctl enable chronyd		
Revision 1-41	Thu June 4 2015	Jo Somers
BZ#1180277 15.2. Red Hat Satellite Capsule Server Prerequisites change firewall-cmd --reloac		
BZ#1222882 Deleted Section 12.2.2 Manual Configuration.		
Revision 1-40	Mon May 11 2015	David O'Brien
Build for technical review.		
Revision 1-39	Fri May 8 2015	Megan Lewis

BZ#1153595 Added conceptual information to the Activation Keys chapter.
 Minor edits to the Creating an Activation Key section.
 Minor edits to the Removing an Activation Key section.
 Restructured the Activation Keys chapter.
 BZ#1175938 Added section on auto-attach to the Activation Keys chapter.
 BZ#1175938 Added section on setting a service level to the Activation Keys chapter.
 Minor edits to the Adding Subscriptions to an Activation Key section.
 Minor edits to the Adding Host Collections to an Activation Key section.
 BZ#1175938 Added section on editing product content to the Activation Keys chapter.
 Minor edits to the Removing Subscriptions to an Activation Key section.
 Minor edits to the Removing Host Collections to an Activation Key section.
 BZ#1175938 Added section on automated host registration to the Activation Keys chapter.
 Added section on Configuring External User Groups.

Revision 1-38	Thu April 30 2015	Megan Lewis
BZ#1175924 Updated note in 4.3.4. Importing Content to a Disconnected Satellite Server.		
Revision 1-37	Wed April 29 2015	Megan Lewis
BZ#1175835 Updated example in 4.3.2 Synchronizing Content.		
BZ#1175924 Updated Step 7 of 4.3.4. Importing Content to a Disconnected Satellite Server.		
Fixed typos in 4.3.4. Importing Content to a Disconnected Satellite Server.		
Revision 1-36	Fri April 24 2015	Megan Lewis
BZ#1177770 Corrected errors in commands in Installing and Configuring the Foreman Discovery Plugin.		
Revision 1-35	Thu April 23 2015	Megan Lewis
BZ#1175431 Corrected mismatch between instruction and location of extra information in Workflow section.		
Revision 1-34	Wed April 22 2015	David O'Brien
New procedure for backup and recovery. Replaces all previous procedures.		
BZ 1213912: Update procedure for changing FQDN.		
Revision 1-33	Fri April 17 2015	Megan Lewis
Updated procedure 12.3.1. Automated Configuration to reflect changes requested on the test day.		
Revision 1-32	Thu April 16 2015	Jo Somers
Section 4.3.4 Importing Content to a Disconnected Satellite Server: Updated with improved procedure from Satellite 6.1 Installation Guide, Section 4.2.4 Importing Content to a Disconnected Satellite Server		
Revision 1-31	Wed April 8 2015	Megan Lewis
Updated the brand.		
Revision 1-30	Thu April 2 2015	Athene Chan
Changed the sort order number for the splash page.		
Revision 1-29	Thu April 2 2015	Athene Chan
Added a sort order number for the splash page.		

Revision 1-28	Mon Mar 30 2015	David O'Brien
BZ 1203878: Change repository name from RH Common to Satellite Tools.		
Revision 1-27	Tue Mar 17 2015	Athene Chan
BZ#1200016 Changed "DNS Proxy" to "DNS Capsule".		
Revision 1-26	Tue Mar 17 2015	David O'Brien
BZ 1172727 Update section on installing puppet to include enabling agent at boot. BZ# 1198724 Add section how to configure BMC interfaces.		
Revision 1-25	Mon Mar 02 2015	Jo Somers
Fixed BZ#1153608 rewrote configure host for registration; edited User Interface Plug-Ins for clarity		
Revision 1-24	Wed Feb 25 2015	Athene Chan
BZ#1180191 Corrected the required RPMs to install for synchronizing hosts in a disconnected Satellite Server.		
Revision 1-23	Tue Feb 24 2015	David O'Brien
BZ 1195128 Remove extra trailing slash from command. BZ 1179535 Update the Disaster Recovery Section of the User Guide.		
Revision 1-22	Mon Feb 9 2015	Megan Lewis
BZ#1178176 Further corrections in 4.3.4. Importing Content to a Disconnected Satellite Serve		
Revision 1-21	Mon Feb 9 2015	David O'Brien
BZ 1175084 Section on Puppet Facts. BZ 1153584 Update section on promoting content views through life cycle environments.		
Revision 1-20	Fri Jan 23 2015	Megan Lewis
BZ#1179022 Corrected errors in examples in 5.4. Configuring a Red Hat Satellite Capsule Server. BZ#1178176 Corrected 40G to 40GB in 4.3.4. Importing Content to a Disconnected Satellite Server.		
Revision 1-19	Fri Jan 23 2015	David O'Brien
Add initial section on Puppet, Facter, and facts. Add Puppet Module and Catalog to glossary. Updates to comply with style guide.		
Revision 1-18	Fri Dec 19 2014	David O'Brien
Remove requirement for yum-rhn-plugin from chapter "Configuring Hosts". Update some command layouts to comply with standards.		
Revision 1-17	Tues Dec 9 2014	Megan Lewis
BZ#1168273 Corrected the package name for Installing the Puppet Agent.		
Revision 1-16.1	Wed Nov 26 2014	Athene Chan
BZ#1139329 Added introductory text into "Using the Foreman Discovery Plug-in". BZ#1167966 Satellite Server backup script has changed, removed grinder from the command list.		

Revision 1-16	Mon Nov 24 2014	Athene Chan
BZ#1166660 Missing step in the configuring IDM chapter added. BZ#1166656 Changed "realm-proxy@example.com" to "realm-capsule@example.com" for consistency. BZ#1139329 Revised the Troubleshooting for the Foreman Discovery Plug-in" section.		
Revision 1-15.2	Fri Nov 21 2014	Athene Chan
Removed the Foreman Discovery chapter for further review.		
Revision 1-15	Thurs Nov 20 2014	Megan Lewis
Minor corrections. Added "Enabling Red Hat Repositories" section.		
Revision 1-14	Mon Nov 17 2014	Megan Lewis
Added further changes for BZ#1139329.		
Revision 1-13	Sun Nov 16 2014	Megan Lewis
BZ#1139329 Added chapter about Foreman Discovery.		
Revision 1-12	Fri Nov 14 2014	Miroslav Svoboda
BZ#1153596 Removed sentence mentioning support of Windows installation media. BZ#1142477 Corrected procedure for Configuring Hosts for Registration.		
Revision 1-11.2	Friday Nov 14 2014	Athene Chan
BZ#1153567 Added a "Capsule Scalability" section. BZ#1152797 Added a "Troubleshooting" section.		
Revision 1-11.1	Mon Nov 10 2014	Athene Chan
BZ#1150412 Added "--complete-reload" to the firewall-cmd firewall commands. BZ#1141954 Changed "Installing the Katello Agent" to "Installing the Katello and Puppet Agents". Added information on puppet-agent in the section.		
Revision 1-11	Mon Nov 10 2014	Athene Chan
BZ#1161254 Added a new firewall rule to the list of firewall rules to allow katello-installer to run after initial install. Moved the firewall rules to the "Configuring Red Hat Satellite" sections to prevent errors. BZ#1110837 Implemented QE edits. BZ#1152630 Added RHEL7 firewall-cmd command examples for the firewall requirements.		
Revision 1-10	Fri Nov 7 2014	Megan Lewis
BZ#1149145 Defined the difference between All Hosts and Content Hosts and made sure all procedures pointed to the correct section. Removed all instances of non breaking spaces in titles.		
Revision 1-9	Thu Nov 6 2014	Athene Chan
BZ#1110837 Added a "Configuring Identity Management" chapter in the User Guide.		
Revision 1-8	Thu Nov 6 2014	Megan Lewis
BZ#1149144 Corrected steps to locate systems registered via subscription-manager.		
Revision 1-7	Thu Oct 30 2014	Megan Lewis

Removed help file output.

Revision 1-6	Thu Oct 23 2014	Megan Lewis
Implemented changes suggested by translation.		
Revision 1-5	Fri Oct 3 2014	Athene Chan
BZ#1140520 Changed all "ACME_Corporation" entries to the correct default organization entry "Default Organization".		
BZ#1141954 Added example repositories to the "Enabling Red Hat Repositories" section and a note to enable RH Common repositories for client systems.		
BZ#1140722 Added note to highlight that the command needs to change if the repository is different from the example command.		
Revision 1-4	Thu Oct 2 2014	Megan Lewis
Implemented brand changes.		
Added Glossary of Terms in an Appendix.		
Revision 1-3	Wed Oct 1 2014	Megan Lewis
Minor edits based on feedback from translation.		
Revision 1-2.01	Fri Sep 12 2014	Athene Chan
BZ#1140875 Added firewall rules after the Satellite Server and Capsule Server installation.		
Revision 1-2	Fri Sep 12 2014	David O'Brien
Patched "Red Hat" entries to conform with Brand standards.		
Revision 1-1	Thu Sep 11 2014	Athene Chan
BZ#1140422 Changed the repository names for Red Hat Satellite Server and Red Hat Satellite Capsule Server.		
Revision 1-0	Tue 9 Sep 2014	Megan Lewis
Red Hat Satellite 6.0 GA Release.		
Revision 0-23	Thu 21 Aug 2014	Megan Lewis
BZ#1131654 - Removed optional from Step 4 in 15.2.1. Red Hat Satellite Backup Procedure.		
BZ#1120722 - Corrected Step 2 in the note in 10.4.1. Registering a Host.		
BZ#1131655 - Corrected database name in sections 15.2.1. Red Hat Satellite Backup Procedure and 15.2.2. Red Hat Satellite Restore Procedure.		
BZ#1131613 - Section on creating a backup added to 1.3. Red Hat Satellite 6 Workflow.		
BZ#1131604 - Section 15.2.1 - Removed "/var/lib/katello" from list for backup.		
Revision 0-22	Fri 15 Aug 2014	Megan Lewis
BZ#1120722 - Note in 10.4.1. Registering a Host corrected to reference a Host instead of a System.		
BZ#1129841 - Added section 10.4.2. Installing the Katello Agent.		
BZ#1127285 - Added prefix to baseurl used when registering clients to capsules.		
BZ#1129578 - Removed sections 3.3.3 and 3.3.4.		
BZ#1104431 - Implemented peer review feedback for Chapters 1-3.		
Updated instructions for managing users and roles.		
Updated instructions for using host collections.		
Revision 0-21	Tue 12 Aug 2014	Athene Chan

BZ#1128872 - Removed stray ; in Table 9.1.

Revision 0-20	Fri 18 July 2014	Athene Chan
----------------------	-------------------------	--------------------

BZ#1120713 - Corrected section xml to prevent validation errors.

Revision 0-19	Fri 11 July 2014	Megan Lewis
----------------------	-------------------------	--------------------

BZ#1109747 - Information added regarding organizations and subscription manifests.

Revision 0-18	Thu 10 July 2014	Athene Chan
----------------------	-------------------------	--------------------

BZ# 1117861 - Section 10.3.1 Corrected CA Certificate URL.

BZ#1104914 - Section 5 Partial peer review implementation.

Revision 0-17	Wed 9 July 2014	Megan Lewis
----------------------	------------------------	--------------------

BZ#1116888 - Section 4.2.2.3 References to Katello CLI corrected to Hammer CLI.

BZ#1116543 - Section 10.3.1 Corrected RPM name.

BZ#1117503 - Section 5.3.1 Removed extra step.

Revision 0-16	Wed 25 Jun 2014	Athene Chan
----------------------	------------------------	--------------------

Preparing book for Beta release.

Revision 0-15	Mon 11 Nov 2013	Dan Macpherson
----------------------	------------------------	-----------------------

Fixing minor issues.

Revision 0-14	Mon 11 Nov 2013	Dan Macpherson
----------------------	------------------------	-----------------------

Preparation for MDP2.

Revision 0-13	Wed 09 Oct 2013	Dan Macpherson
----------------------	------------------------	-----------------------

Adding table for synchronization content directories.

Revision 0-12	Wed 09 Oct 2013	Dan Macpherson
----------------------	------------------------	-----------------------

Finalizing QE review implementation

Revision 0-11	Tue 1 Oct 2013	Athene Chan
----------------------	-----------------------	--------------------

BZ#887680 Minor typo corrections.

Revision 0-10	Mon 30 Sep 2013	Dan Macpherson
----------------------	------------------------	-----------------------

Rebuild from typo verification.

Revision 0-09	Wed 18 Sep 2013	Athene Chan
----------------------	------------------------	--------------------

Minor tagging errors corrected.

Revision 0-08	Tue 17 Sep 2013	Athene Chan
----------------------	------------------------	--------------------

BZ#956256, 969922, 864115 Implemented suggested changes to information on the User Guide.

Revision 0-07	Fri 13 Sep 2013	Athene Chan
----------------------	------------------------	--------------------

Book product changed.

Revision 0-06	Thu 12 Sep 2013	Athene Chan
----------------------	------------------------	--------------------

Minor grammatical edits.

Added book component to the ent file.

Revision 0-05	Thu 12 Sep 2013	Athene Chan
BZ#1004566, 1004567, 1004568, 1004570, 1004571, 1004581, 1004586, 1004588, 1004590 1004595, 1004597, 1004598, 1004600 Quality assurance edits implemented throughout the book.		
Revision 0-04	Mon 12 Aug 2013	Dan Macpherson
Removing draft watermark.		
Revision 0-03	Mon 12 Aug 2013	Dan Macpherson
Creating build for technical review.		
Revision 0-02	Tue 28 May 2013	Athene Chan
Initial book creation		