



Red Hat Satellite 6.0

User Guide

A guide to using Satellite entitlement management software.

Red Hat Satellite 6.0 User Guide

A guide to using Satellite entitlement management software.

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. INTRODUCTION TO RED HAT SATELLITE	5
1.1. RED HAT SATELLITE 6 SYSTEM ARCHITECTURE	5
1.2. RED HAT SATELLITE 6 LAYOUT AND TOPOLOGY	6
1.3. RED HAT SATELLITE SERVER 6 BASIC CONFIGURATION WORKFLOW	9
CHAPTER 2. ACCESSING RED HAT SATELLITE	12
2.1. LOGGING IN TO RED HAT SATELLITE	12
2.2. CHANGING THE PASSWORD IN RED HAT SATELLITE	13
CHAPTER 3. CONFIGURING ORGANIZATIONS, LOCATIONS AND LIFE CYCLE ENVIRONMENTS	15
3.1. ORGANIZATIONS	15
3.2. LOCATIONS	18
3.3. LIFE CYCLE ENVIRONMENTS	19
CHAPTER 4. USING CONTENT MANAGEMENT	22
4.1. USING THE RED HAT SATELLITE CONTENT DASHBOARD	22
4.2. CONNECTED SATELLITE	23
4.3. DISCONNECTED SATELLITE	30
CHAPTER 5. USING CONTENT VIEWS	38
5.1. CREATING A CONTENT VIEW	38
5.2. ADDING REPOSITORIES TO THE CONTENT VIEW	39
5.3. FILTERING CONTENT	39
5.4. PUBLISHING A CONTENT VIEW	42
CHAPTER 6. SEARCHING FOR CONTENT	44
6.1. BASIC CONTENT SEARCH	44
6.2. CONTENT COMPARISON ACROSS ENVIRONMENTS	44
CHAPTER 7. VIEWING AND APPLYING ERRATA	45
7.1. INSPECTING AVAILABLE ERRATA	45
7.2. APPLYING ERRATA TO CONTENT HOSTS	47
7.3. SUBSCRIBING TO ERRATA NOTIFICATIONS	48
CHAPTER 8. WORKING WITH CONTAINERS	50
8.1. MANAGING CONTAINERS	50
8.2. WORKING WITH REPOSITORIES	54
8.3. ADDING AN EXTERNAL REGISTRY	55
8.4. IMPORTING IMAGES TO A COMPUTE RESOURCE	55
8.5. USING TAGS	55
CHAPTER 9. CONFIGURING ACTIVATION KEYS	57
9.1. CREATING AN ACTIVATION KEY	57
9.2. REMOVING AN ACTIVATION KEY	57
9.3. ADDING OBJECTS TO ACTIVATION KEYS	58
9.4. REMOVING OBJECTS FROM ACTIVATION KEYS	58
CHAPTER 10. CONFIGURING GPG KEYS	60
10.1. CREATING A GPG KEY	60
10.2. REMOVING A GPG KEY	60
CHAPTER 11. CONFIGURING THE PROVISIONING ENVIRONMENT	61
11.1. CREATING A HOST GROUP	61
11.2. PARAMETERS	62

11.3. CONFIGURING PROVISIONING SETTINGS	65
11.4. STORING AND MAINTAINING HOST INFORMATION	72
CHAPTER 12. CONFIGURING HOSTS	76
12.1. CREATING A HOST	76
12.2. CONFIGURING HOSTS FOR REGISTRATION	76
12.3. CONFIGURATION OPTIONS	77
12.4. CONFIGURING ADDITIONAL NETWORK INTERFACES	78
12.5. REGISTRATION	82
12.6. REMOVING A HOST	85
CHAPTER 13. USING THE FOREMAN DISCOVERY PLUG-IN	86
13.1. INSTALLING THE FOREMAN DISCOVERY PLUG-IN	86
13.2. CONFIGURING THE FOREMAN DISCOVERY PLUG-IN	86
13.3. EDITING THE DISCOVERED HOST	88
13.4. TROUBLESHOOTING THE FOREMAN DISCOVERY PLUG-IN	88
CHAPTER 14. CONFIGURING HOST COLLECTIONS	89
14.1. CREATING A HOST COLLECTION	89
14.2. ADDING HOSTS TO A HOST COLLECTION	89
14.3. ADDING CONTENT TO HOST COLLECTIONS	90
14.4. REMOVING CONTENT FROM A HOST COLLECTION	91
14.5. CHANGING THE ASSIGNED LIFE CYCLE ENVIRONMENT OR CONTENT VIEW FOR A HOST COLLECTION	91
14.6. REMOVING A HOST FROM A HOST COLLECTION	92
14.7. REMOVING A HOST COLLECTION	92
14.8. CLONING A HOST COLLECTION	92
14.9. REVIEWING HOST COLLECTION ACTIONS	93
CHAPTER 15. RED HAT SATELLITE CAPSULE SERVERS	94
15.1. RED HAT SATELLITE CAPSULE SERVER SCALABILITY	94
15.2. RED HAT SATELLITE CAPSULE SERVER PREREQUISITES	95
15.3. INSTALLING A RED HAT SATELLITE CAPSULE SERVER	98
15.4. CONFIGURING A RED HAT SATELLITE CAPSULE SERVER	99
15.5. USING POWER MANAGEMENT FEATURES ON MANAGED HOSTS	103
15.6. ADDING LIFE CYCLE ENVIRONMENTS TO A RED HAT SATELLITE CAPSULE SERVER	105
15.7. REMOVING LIFE CYCLE ENVIRONMENTS FROM THE RED HAT SATELLITE CAPSULE SERVER	106
15.8. REGISTERING HOST SYSTEMS TO A RED HAT SATELLITE CAPSULE SERVER	107
15.9. REFRESHING A RED HAT SATELLITE CAPSULE SERVER	107
CHAPTER 16. USERS AND ROLES	109
16.1. CONFIGURING LDAP AUTHENTICATION FOR RED HAT SATELLITE	109
16.2. CREATING AND MANAGING USERS	110
16.3. CREATING AND MANAGING ROLES	111
16.4. GRANULAR PERMISSION FILTERING	114
CHAPTER 17. COMMAND LINE REFERENCE	116
CHAPTER 18. BACKUP AND DISASTER RECOVERY	117
18.1. BACKING UP RED HAT SATELLITE	117
18.2. RESTORING RED HAT SATELLITE FROM BACKUP	120
CHAPTER 19. MAINTENANCE	124
19.1. LOGGING AND REPORTING	124
19.2. TROUBLESHOOTING	125

CHAPTER 20. CONFIGURING IDENTITY MANAGEMENT IN RED HAT SATELLITE	127
20.1. CONFIGURING RED HAT SATELLITE SERVER OR CAPSULE SERVER FOR IDM REALM SUPPORT	127
20.2. ADDING HOSTS TO AN IDM HOST GROUP	129
CHAPTER 21. RED HAT SATELLITE USER INTERFACE PLUG-INS	131
21.1. RED HAT ACCESS PLUG-IN	131
APPENDIX A. GLOSSARY OF TERMS	135
APPENDIX B. REVISION HISTORY	139

CHAPTER 1. INTRODUCTION TO RED HAT SATELLITE

Red Hat Satellite 6 is the evolution of Red Hat's life cycle management platform. It provides the capabilities that administrators have come to expect in a tool focused on managing systems and content for a global enterprise. Satellite 6 covers the use cases requested by Satellite 5 customers, but also includes functionality that enables larger scale, federation of content, better control of systems during the provisioning process, and a much more simplified approach to life cycle management. Satellite 6 also further evolves the inherent approach to certificate-based entitlements and integrated subscription management. Satellite 6 is based on years of customer feedback and is an evolution of previous versions.

[Report a bug](#)

1.1. RED HAT SATELLITE 6 SYSTEM ARCHITECTURE

Red Hat Satellite 6 is based upon several open source projects arranged in the following architecture.

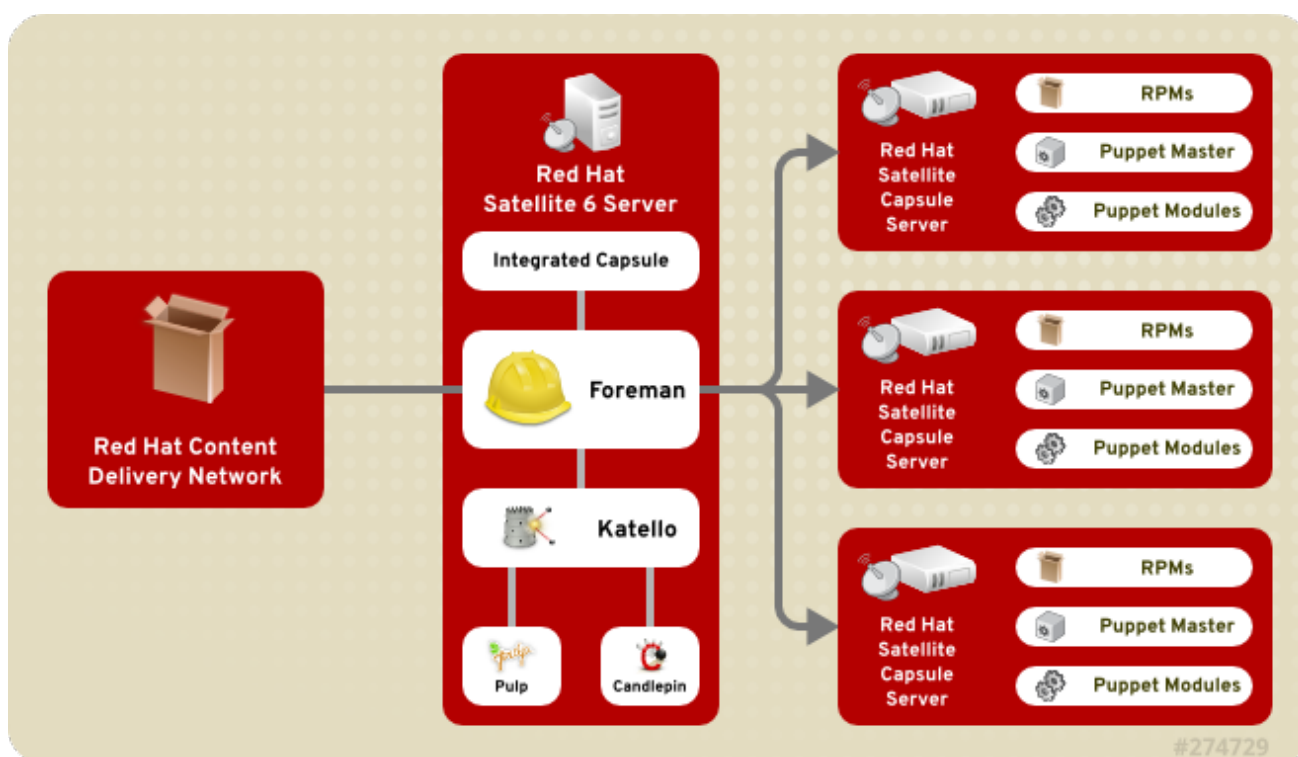


Figure 1.1. Red Hat Satellite 6 System Architecture

Foreman

Foreman is an open source application used for provisioning and life cycle management of physical and virtual systems. Foreman automatically configures these systems using various methods, including kickstart and Puppet modules. Foreman also provides historical data for reporting, auditing, and troubleshooting.

Katello

Katello is a subscription and repository management application. It provides a means to subscribe to Red Hat repositories and download content. You can create and manage different versions of this content and apply them to specific systems within user-defined stages of the application life cycle.

Candlepin

Candlepin is a service within Katello that handles subscription management.

Pulp

Pulp is a service within Katello that handles repository and content management.

Hammer

Hammer is a CLI tool that provides command line and shell equivalents of most Web UI functions.

REST API

Red Hat Satellite 6 includes a RESTful API service that allows system administrators and developers to write custom scripts and third-party applications that interface with Red Hat Satellite.

Capsule

Red Hat Satellite Capsule Server acts as a proxy for some of the main Satellite functions including repository storage, **DNS**, **DHCP**, and Puppet Master configuration. Each Satellite Server also contains integrated Capsule Server services.

[Report a bug](#)

1.2. RED HAT SATELLITE 6 LAYOUT AND TOPOLOGY

The Red Hat Satellite infrastructure layout has to be considered prior to installing Red Hat Satellite 6. Determining the organization of your infrastructure helps align the Satellite Server and Satellite Capsule Servers to best serve your requirements. The following topology diagrams provide examples of infrastructure layout.

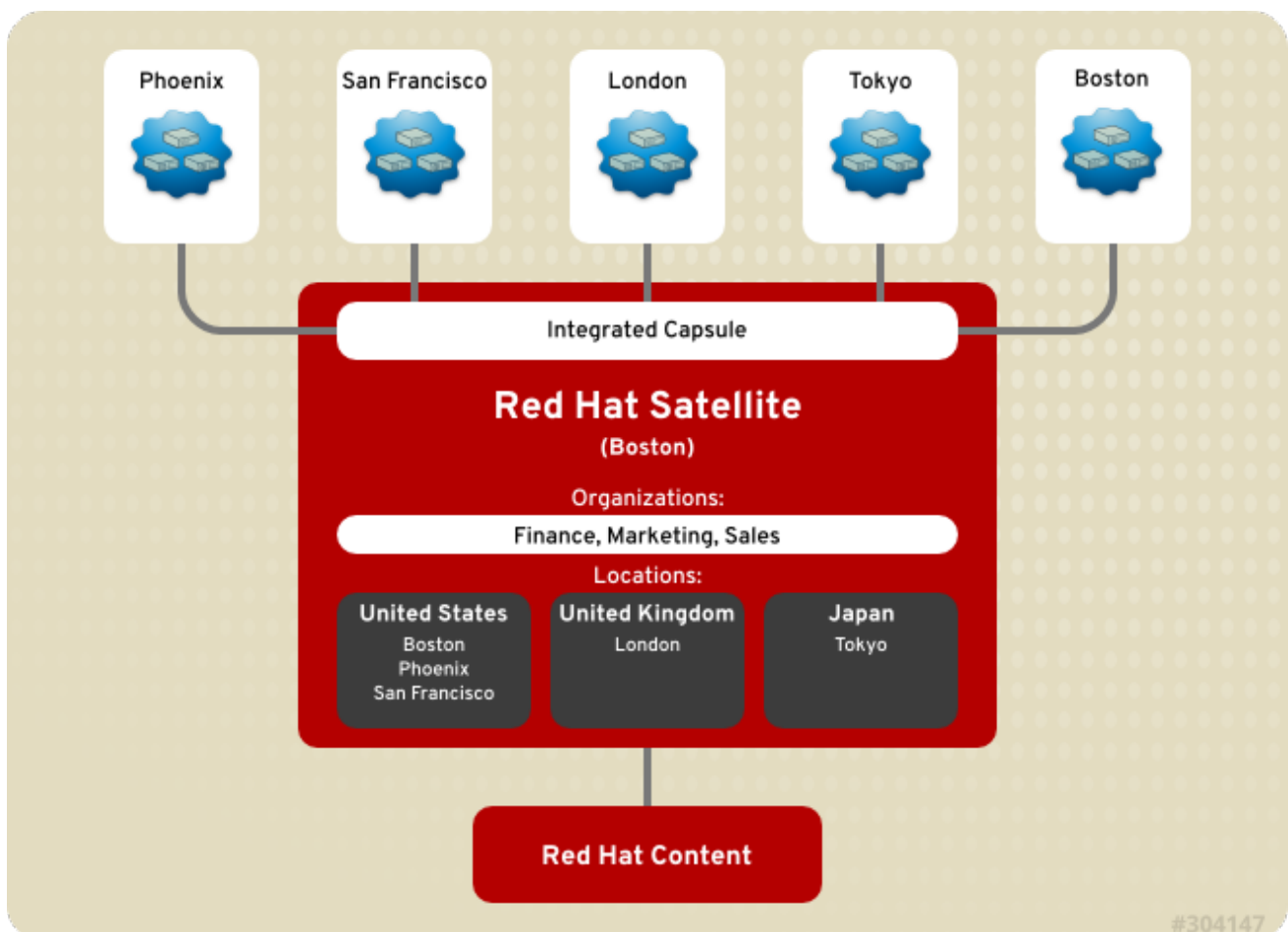


Figure 1.2. Single Satellite with Integrated Capsule

This topology demonstrates the basic use of Red Hat Satellite 6. In this example, there are five pools of hosts registered to the Satellite Server. Red Hat Satellite 6 categorizes these pools into three locations: United States, United Kingdom, and Japan. In addition, each department uses a distinct organization: Finance, Marketing, and Sales. All Satellite Server functions are shared among these Locations and Organizations.

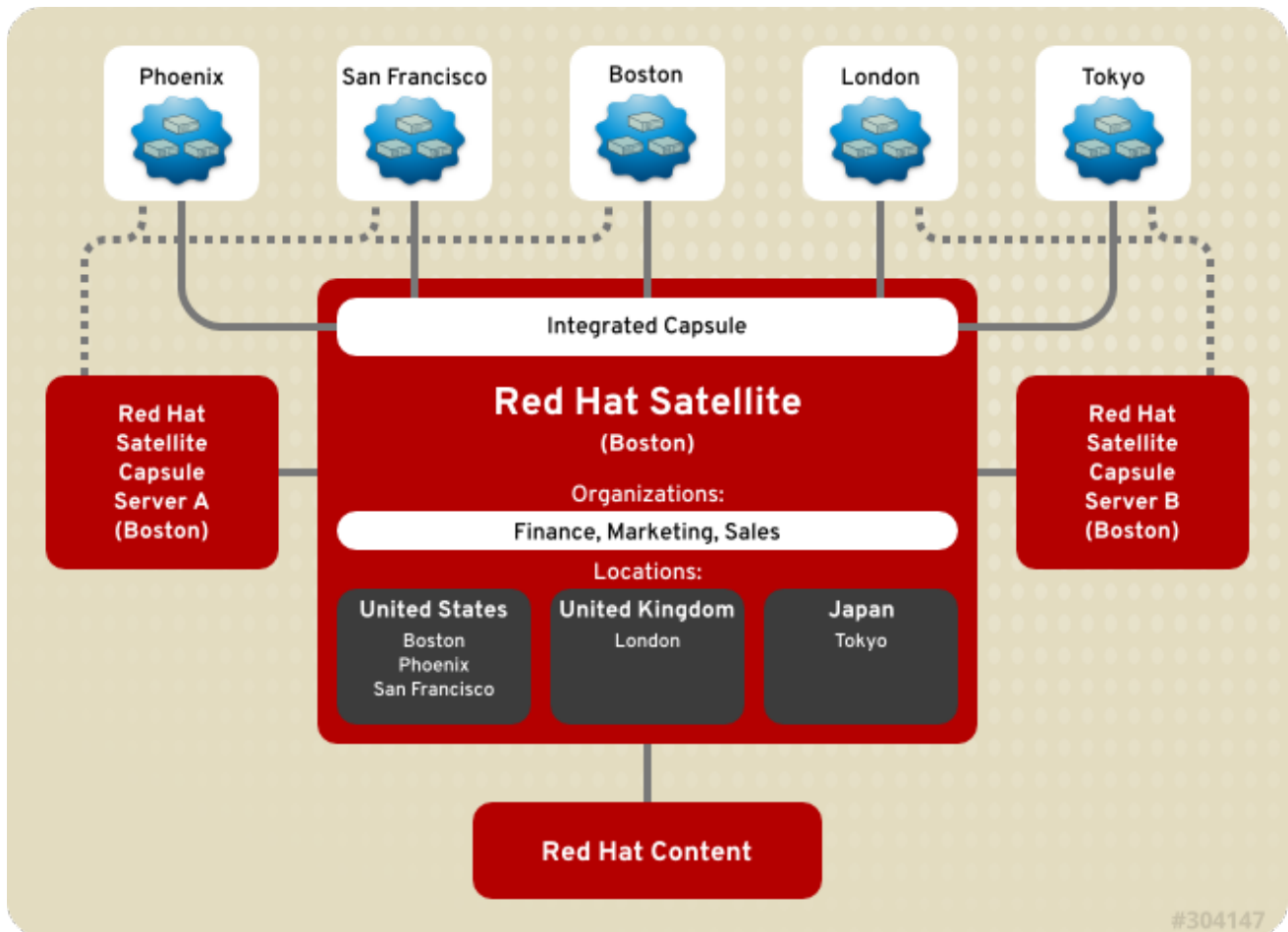


Figure 1.3. Single Satellite with Integrated Capsule and Backup Capsules

This topology shows the addition of two backup Satellite Capsule Servers based in Boston. One Capsule Server caters to the three US-based offices: Phoenix, San Francisco, and Boston. The other caters to the international offices: London and Tokyo. Assigning one backup Capsule Server to the United States location and the other to the United Kingdom and Japan locations, the load reduces on the main Satellite Server and its integrated Capsule. As new offices are created in these locations, the Satellite Server can add them to their location categories and the Capsule Server services the new offices.

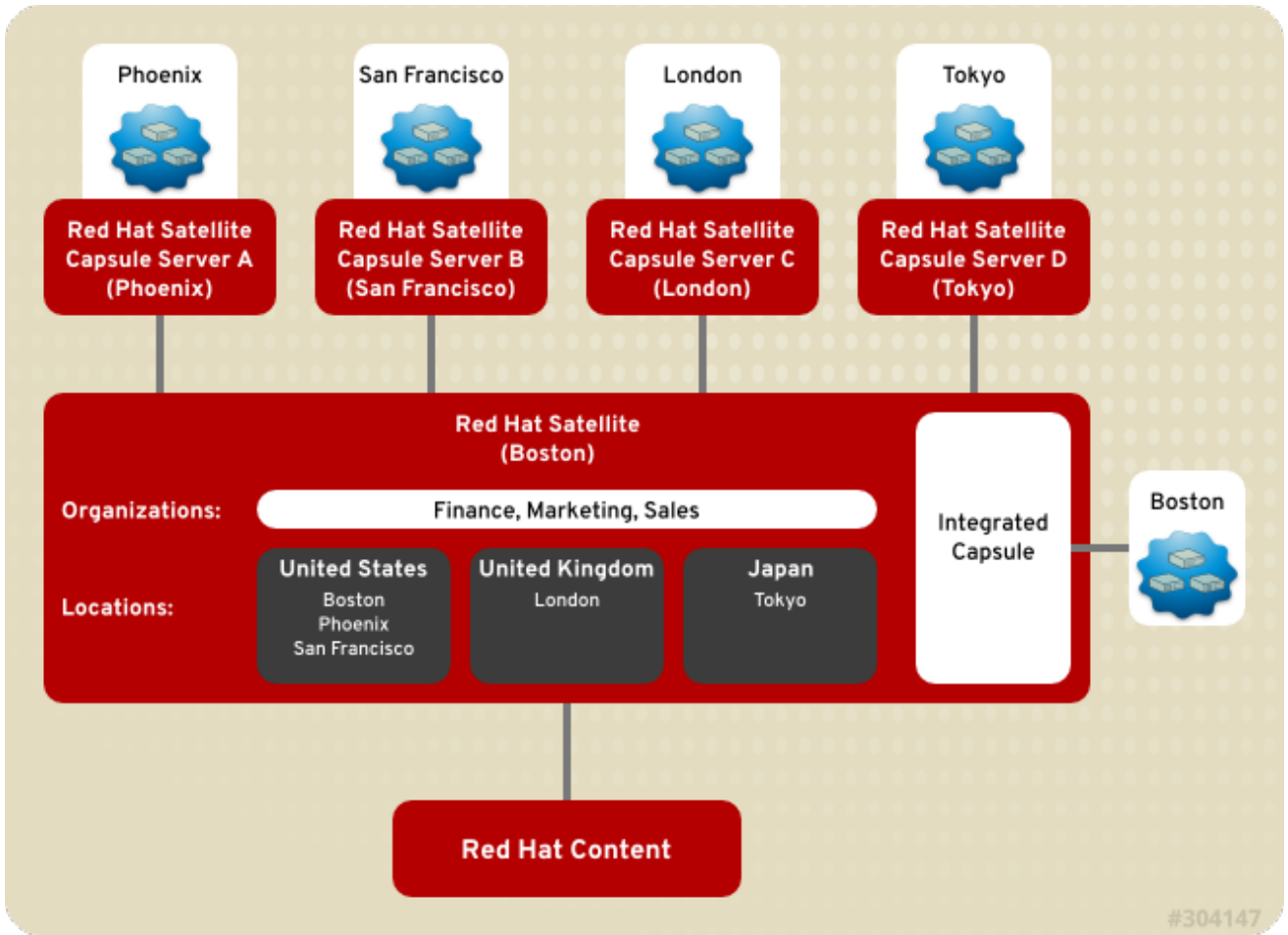
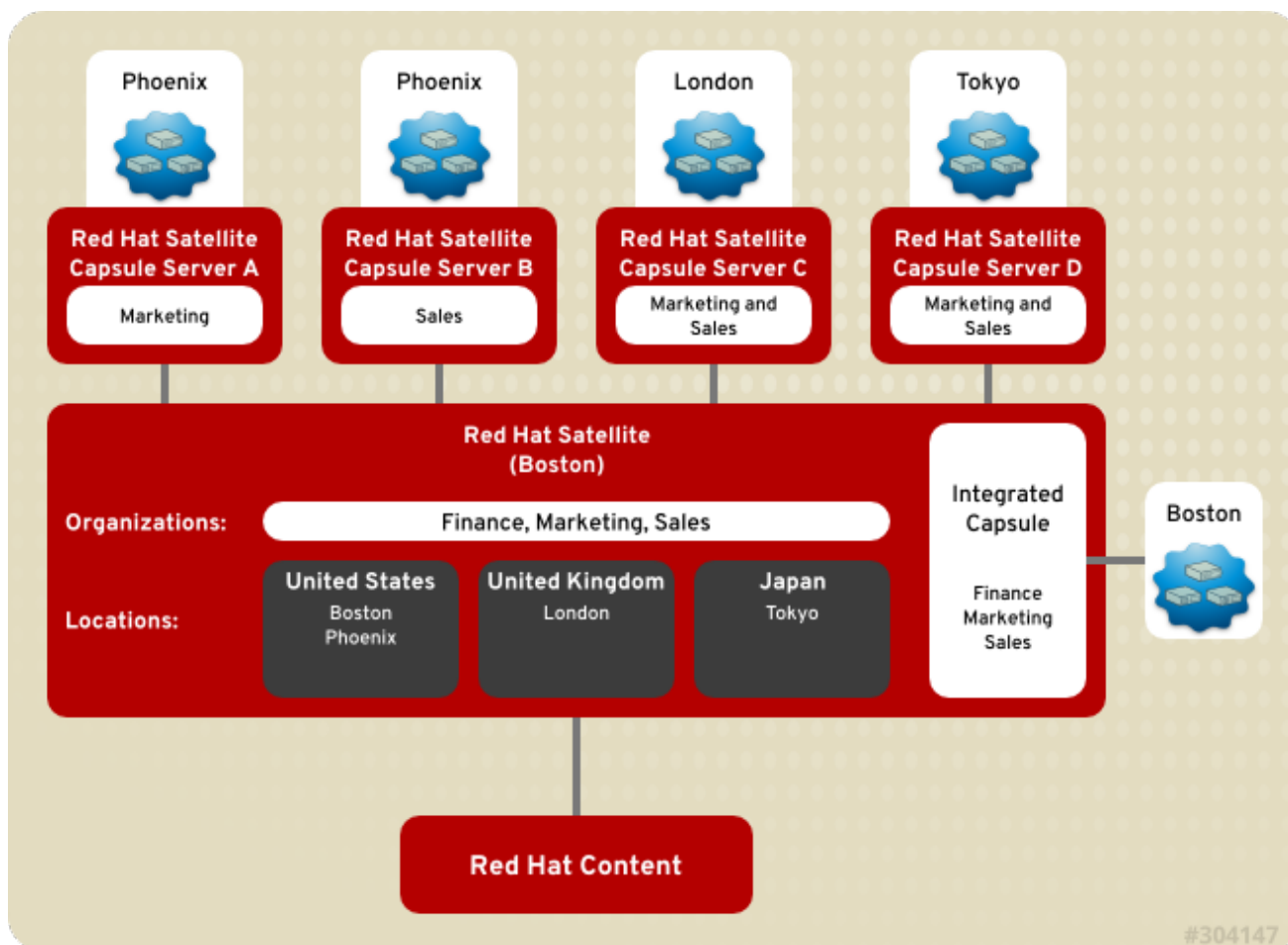


Figure 1.4. Remote Capsules Based on Location

This topology assigns Satellite Capsule Servers to specific locations. The Satellite Server can create a hierarchy of locations; for example, having cities attached to a country. The Satellite Capsule Servers based in these locations are registered to the central Satellite Server in Boston and assigned to their respective locations. Each Capsule services all hosts in each respective location.



#304147

Figure 1.5. Remote Capsules Based on Location and Organization

This topology demonstrates Satellite Capsule Servers assignment to organizations. For example, two Capsule servers are assigned to the Phoenix location, but to different organizations: one for Marketing and the other for Sales. Likewise, two Satellite Capsule Servers are assigned to two organizations: one for both the Sales and Marketing organizations in London, and the other for both the Sales and Marketing organizations in Tokyo. This shows how the combination of Satellite Server and associated Satellite Capsule Servers can manage the layout of multiple organizations in multiple locations working together.

[Report a bug](#)

1.3. 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 additional required remote capsules. See the *Red Hat Satellite 6 Server Installation Workflow* in the *Red Hat Satellite 6 Installation Guide* for further information.

Initial Configuration

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

1. Log in to the Satellite Server. This requires the administration user and password. 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 3.1.3, “Editing an Organization”](#) and [Section 3.2.2, “Editing a Location”](#) for more information.
3. Edit the desired location to select the resources to be associated with that location. See [Section 3.1, “Organizations”](#) for more information.
4. Edit the default organization to select the resources to be associated with that organization. See [Section 3.2, “Locations”](#) for more information.
5. Refresh the Satellite Capsule Server. See [Section 15.9, “Refreshing a Red Hat Satellite Capsule Server”](#) 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 11.3.1, “Domains”](#) for more information.
2. Create a subnet. See [Section 11.3.2, “Subnets”](#) for more information.
3. Create the desired life cycle environments. See [Section 3.3, “Life Cycle Environments”](#) for more information.
4. Create any desired custom products. See [Section 4.2.2.1, “Creating a Product”](#) for more information.
5. Choose the desired Red Hat Repositories.
 1. Create a manifest from the Red Hat Customer Portal. See [Section 4.2.1.1, “Setting up a Manifest”](#) for more information.
 2. Upload the manifest in the Satellite Server web interface. This will propagate the subscription information into the Satellite Server. See [Section 4.2.1.2, “Uploading a Subscription Manifest”](#) for more information.
 3. 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 [Section 4.2.1.2, “Uploading a Subscription Manifest”](#) for more information.
 4. Optional:
 1. Red Hat source repositories update content based on security errata, bug fixes, and enhancements. To ensure that the Satellite Server is updated automatically, [Section 4.2.3.2, “Creating a Synchronization Plan”](#) and [Section 4.2.3.3, “Applying a Synchronization Schedule”](#) are recommended practices.
6. Manually synchronize content. See [Section 4.2.3.1, “Synchronization Status”](#) 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 5, Using Content Views](#) for more information.
8. Optional:

1. Create a host collection and assign it to the desired life cycle environment and content view. See [Chapter 14, *Configuring Host Collections*](#) for more information.
9. Create an activation key assigning it to the desired life cycle environment and content view. See [Section 9.1, “Creating an Activation Key”](#) for more information.
10. Edit an existing provisioning template and associate it with the previously created operating system. See [Section 11.3.9, “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 11.3.7, “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 11.3.6, “Installation Media”](#) for more information.
13. Create a host group with the desired details. See [Section 11.1, “Creating a Host Group”](#) for more information.

Creating a Backup of a Red Hat Satellite Server

This is the procedure to create a backup of the Red Hat Satellite Server:

1. Create a backup of the Satellite Server containing the required configuration files, data files, repositories, and databases. See the [Section 18.1, “Backing up Red Hat Satellite”](#) for more information.

[Report a bug](#)

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.

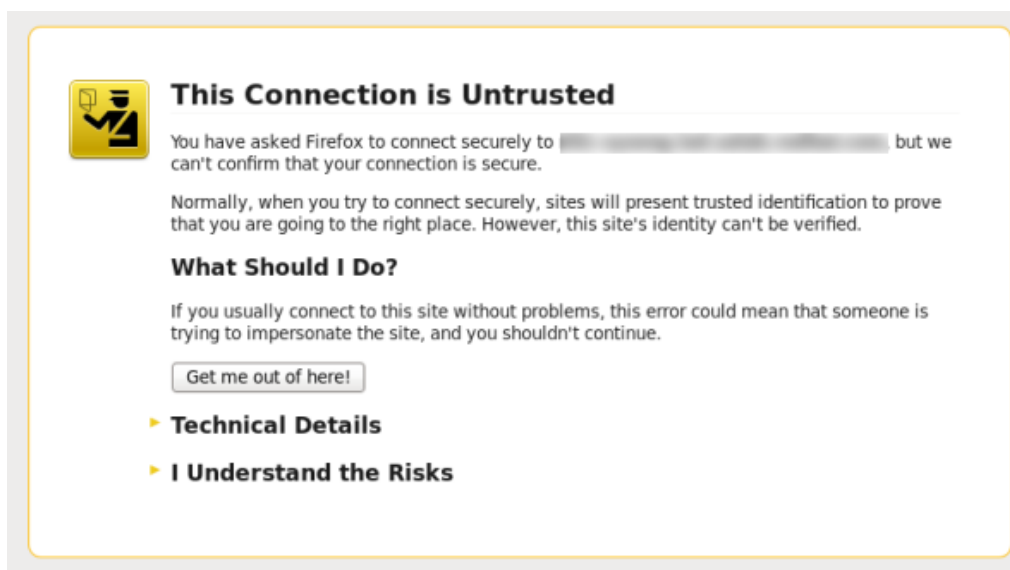


Figure 2.1. Untrusted Connection Warning

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.

[Report a bug](#)

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.

[Report a bug](#)

CHAPTER 3. 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.

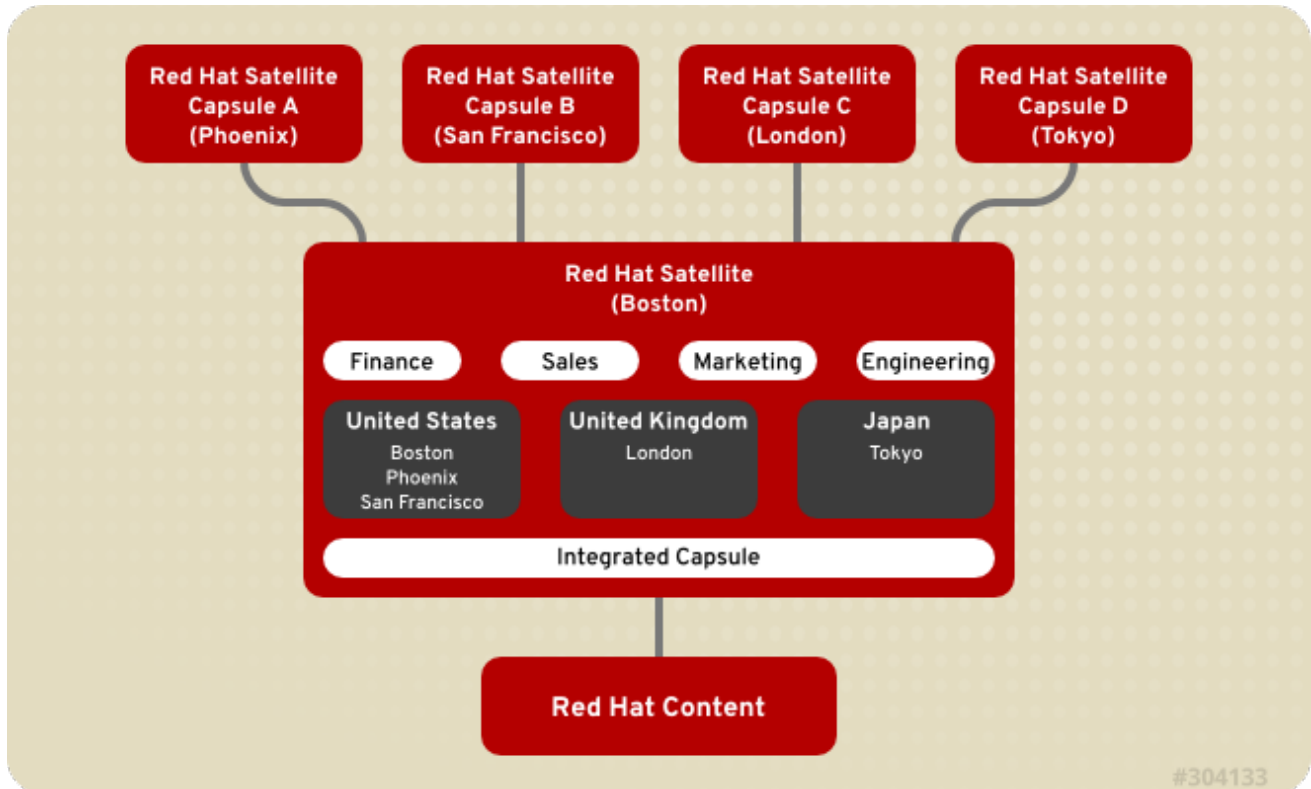


Figure 3.1. Example Topology for Red Hat Satellite 6

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.

[Report a bug](#)

3.1. 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.



IMPORTANT

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

[Report a bug](#)

3.1.1. Creating an Organization

These steps show how to create a new organization.

Procedure 3.1. Creating an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Click the **New Organization** button.
3. Type in the name of the new organization in the **Name** field.
4. Type in the label of the new organization in the **Label** field.
5. Type in a description of the new organization in the **Description** field.
6. Click the **Submit** button.
7. Select the hosts to assign to the new organization.
 - Click the **Assign All** button to assign all hosts with no organization to the new organization.
 - Click the **Manually Assign** button to manually select and assign the hosts with no organization.
 - Click the **Proceed to Edit** button to skip assigning hosts.

Result:

A new organization is created.

[Report a bug](#)

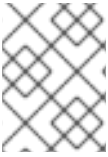
3.1.2. Creating an Organization Debug Certificate

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 3.2. Creating a New Organization Debug Certificate

1. Click the **Administer** → **Organizations** menu on the upper right corner.
2. Select an existing organization from the list on the left.

3. Click the **Generate and Download** button. This generates a debug certificate. Save the certificate 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.

Result:

Red Hat Satellite generates a debug certificate and saves it to a location of your choice.

[Report a bug](#)

3.1.3. Editing an Organization

Procedure 3.3. Editing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
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 the **Submit** button.

Result


The organization is updated and saved.

[Report a bug](#)

3.1.4. Removing an Organization

Procedure 3.4. Removing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
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 Name*?

4. Click the **OK** button.

Result

The organization is removed from **Red Hat Satellite**.

[Report a bug](#)

3.2. LOCATIONS

Locations divide organizations into logical groups based on geographical location.

Each location must be created and used by a single Red Hat customer account, however each account can manage multiple locations and organizations.

By default, Red Hat Satellite will have one location already created, called **Default**, which can be modified to suit your own installation, or deleted.



IMPORTANT

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

[Report a bug](#)

3.2.1. Creating a Location

These steps show how to create a location.

Procedure 3.5. Creating a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.
2. Click the **New Location** button.
3. Type in the name of the new location in the **Name** field and click the **Submit** button.
4. Select the hosts to assign to the new location.
 - Click the **Assign All** button to assign all hosts with no location to the new location.
 - Click the **Manually Assign** button to manually select and assign the hosts with no location.
 - Click the **Proceed to Edit** button to skip assigning hosts.

Result:

A location is created.

[Report a bug](#)

3.2.2. Editing a Location

Procedure 3.6. Editing a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.
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 the **Submit** button.

Result

The location is updated and saved.

[Report a bug](#)

3.2.3. Removing a Location

These steps show how to remove an existing location.

Procedure 3.7. Removing a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.
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 Name*

3. Click the **OK** button.

Result

The location is removed from **Red Hat Satellite**.

[Report a bug](#)

3.3. 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.

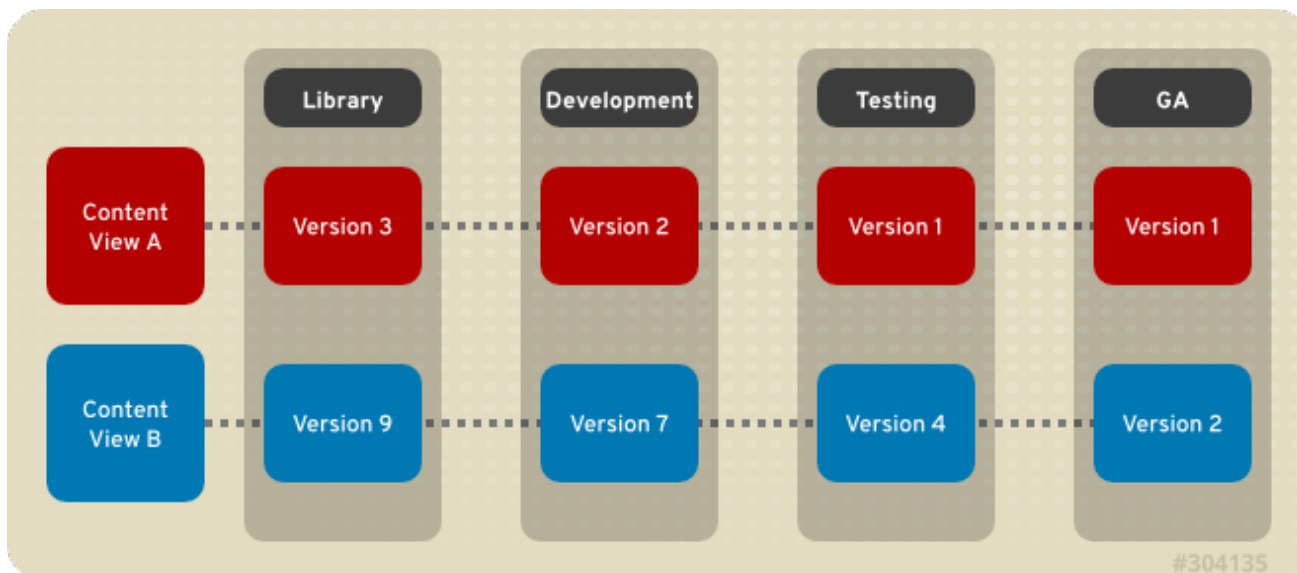


Figure 3.2. An Environment Path Containing Four Environments

[Report a bug](#)

3.3.1. Creating Life Cycle Environments

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

Procedure 3.8. To Create a Life Cycle Environment:

1. On the main menu, click **Content** → **Life Cycle Environments** and then click **New Environment Path**.
2. Enter a name and label for the life cycle environment. The **Description** field is optional.
3. Click **Save** to create the environment.

[Report a bug](#)

3.3.2. 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.

Procedure 3.9. 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**.

**NOTE**

You can only promote content views to the next environment in the promotion path. You cannot skip environments.

5. After the promotion has completed, the **Versions** tab updates to display the new status of your content views.

3.3.3. Removing Life Cycle Environments

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

Procedure 3.10. 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.

[Report a bug](#)

CHAPTER 4. USING CONTENT MANAGEMENT

4.1. USING THE RED HAT SATELLITE CONTENT DASHBOARD

The Satellite Content Dashboard

The 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.




Satellite is used to manage entitlements for client machines. Each entitlement provides access to a specified number of certificates. Each certificate grants the right for the client machine to download, update, and receive support for a product.

The dashboard is accessed by clicking the **Monitor** → **Content Dashboard** menu. The dashboard can be rearranged by clicking on a section title and dragging the section to another position.

Content Host Subscription Status

The **Content Host Subscription Status** gives an overview of the status of the subscriptions currently being managed by **Satellite**. A subscription is a purchased certificate that unlocks access to software, upgrades, and security fixes for hosts.

Table 4.1. 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.	

Latest Notifications

All messages produced by the host are listed in the **Latest Notifications** section. This includes administration information, product and subscription changes, and any errors. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

This section should be monitored for global notifications sent to all users as well as to pick up 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 the **Sync Overview** section. Click a product name to view the synchronization status. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

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. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

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. Clicking on the number for each type of subscription will show a list of those subscriptions.

Content Views Overview

A list of all Content Views in **Satellite** and their publish status. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

Errata Overview

A list of all errata in **Satellite**. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

[Report a bug](#)

4.2. CONNECTED SATELLITE

Red Hat Satellite provides different types of content to subscribed client hosts. Content types include packages, errata updates, kickstart trees, and installation images.

Satellite Server requires a source to provide this content. The content is configured by uploading a *subscription manifest* file to the Satellite. This file can be obtained through the Red Hat Customer Portal, or by contacting Red Hat Support. Manifests provide subscriptions to client hosts through the Red Hat Satellite rather than through Red Hat Network.

This chapter outlines the process of populating your Red Hat Satellite Server, whether it is a connected Red Hat Satellite Server or a disconnected Red Hat Satellite Server, with the content it requires so client hosts can be subscribed to it and receive updates.

[Report a bug](#)

4.2.1. Using Red Hat Content Providers

4.2.1.1. Setting up a Manifest

A *subscription manifest* can be obtained through the method below or by contacting Red Hat Support. The manifest is used to set up Red Hat content providers and contains repository information and subscriptions. It is used as a basis of dispensing subscriptions and Red Hat Network (RHN) content to client systems from **Red Hat Satellite**.

Prerequisites

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

- A Customer Portal user name and password.
- Sufficient subscriptions to add to the manifest.

These steps show how to obtain the subscription manifest from the Customer Portal:

1. Log in to the [Customer Portal](#).
2. Click **Subscriptions** → **Subscriptions Management** → **Subscriptions Management Applications** and then click **Satellite**.
3. On the upper right corner of the Subscriptions Management Applications page, click **Register a Satellite**.
4. Create a name to distinguish your Satellite from the other Satellite systems in your account.
5. Select **6.0** from the drop-down menu as the **Satellite Version**. It is important to select the correct version as each version requires a certain subset of packages.
6. Click **Register**.
7. Click **Attach a subscription**, add the subscriptions required for Red Hat Satellite, and then click **Attach Selected**. See [How to generate a certificate](#) for more information.
8. Click **Download manifest** to generate an archive in .zip format that contains the manifest for Red Hat Satellite.

Result:

A subscription manifest is created and downloaded for Red Hat Satellite.

[Report a bug](#)

4.2.1.2. Uploading a Subscription Manifest

This section describes how to upload a subscription manifest to an organization. Because subscription manifests are assigned to an organization, ensure you select an organization before you try to upload a subscription manifest. Failing to do so will cause a permission denied error (Error 403).

Procedure 4.1. To Upload Subscription Manifest:

1. Log in to the **Satellite** server.
2. Click **Any Context** → **Any Organization** and select the organization that you want to assign the subscription manifest to.
3. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest** at the upper right of the page.
4. In the **Subscription Manifest** section, click **Actions** and under the **Upload New Manifest** subsection, click **Browse**.
5. Select the manifest file to upload, and then click **Upload**.

[Report a bug](#)

4.2.1.3. Enabling Red Hat Repositories

The Red Hat Satellite manifest file provides access to Red Hat products and repositories. Because most products have several architectures and product versions, Red Hat Satellite Server allows the Satellite administrators to choose which repositories are required by their organizations. You need to enable the repositories in Red Hat Satellite Server to prepare them for synchronization.

Procedure 4.2. To Enable Red Hat Repositories:

1. On the main menu, click **Content** → **Red Hat Repositories** and then click the tab for the type of content that you want to enable.
2. Click the product name for which you want to add repositories. This expands the list of available repository sets.
3. Click each repository set from which you want to select repositories, and select the check box for each required repository. The repository is automatically enabled.



IMPORTANT

Ensure you enable the Satellite Tools repository. This repository provides the katello-agent and puppet-agent packages for clients registered to the Satellite Server.

The following is an example set of subscriptions that contain repositories with the latest packages for Red Hat Enterprise Linux 6:

- Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server Repository
- Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server Repository
- Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64 Repository

[Report a bug](#)

4.2.2. Using Products

4.2.2.1. Creating a Product

These steps show how to create a new product in **Red Hat Satellite**.

Procedure 4.3. Creating a Product

1. Click **Content** → **Products**.
2. Click the + **New Product** link.
3. Type in the name of the new product in the **Name** field.
4. Type in label for the new product in the **Label** field.
5. Select a GPG key from the **GPG Key** drop down menu.

6. Select a synchronization plan from the **Sync Plan** drop down menu. Alternatively select the **+ New Sync Plan** link to create a new synchronization plan.
7. Type in a description of the new product in the **Description** field.
8. Click the **Save** button to save your new product.

Result:

A new product is created.

[Report a bug](#)

4.2.2.2. Adding Repositories to a Product

These steps show how to add repositories to a product in **Red Hat Satellite**.

Procedure 4.4. Adding Repositories to a Product

1. Click **Content** → **Products**.
2. Click the product you wish to add a repository to.
3. Click the **Repositories** subtab.
4. Click the **Create Repository** button.
5. Type in the name of the new repository in the **Name** field.
6. Type in a label for the new repository in the **Label** field.
7. Select the type of the repository from the **Type** drop down menu.
8. Type in the URL of the repository in the **URL** field.
9. Choose whether to publish the repository via HTTP by clicking the **Publish via HTTP** check box.
10. Select a GPG key for the repository from the **GPG Key** drop down menu.
11. Click the **Create** button to save your new repository.

Result:

A new repository is added to your product.

[Report a bug](#)

4.2.2.3. Using Bulk Actions for Products

This section describes how to use bulk actions to synchronize or remove products in Red Hat Satellite. The procedure described here requires that at least one product be available.

Procedure 4.5. To Perform Tasks on Multiple Products:

1. Click **Content** → **Products**.

2. Select the check box for the products you want to work with.
3. Click **Bulk Actions**.
4.
 - To synchronize all selected products, click the **Product Sync** tab and then click **Sync Now**.
 - To remove all selected products, click **Remove Products** and then click **Remove**.

Updating Synchronization Plans

You can also update the synchronization plans for multiple products at the same time.

- To create a new synchronization plan, click **Create Sync Plan**.
- To remove the synchronization plans from the selected products, click **Unattach Sync Plan**.
- To update the synchronization plans for the selected products, click **Update Sync Plan**.

[Report a bug](#)

4.2.2.4. Using Repository Discovery

Repository discovery allows you to search a URL to discover repositories available there to include in a product.

Procedure 4.6. Using Repository Discovery

1. Click the **Content** → **Products** menu.
2. Click the **Repo Discovery** button.
3. Enter the URL where the repositories are located in the **Yum Repo Discovery** field.
4. Click the **Discover** button.
5. A list of the repositories at the URL is displayed under **Results**.
6. Click the **Discovered URLs** check box for the repositories to be added to a product.
7. Click the **Create selected** button.
8. Choose whether to add the repositories to an existing product or create a new product.
 - a. To add the repositories to an existing product:
 - i. Select the **Existing Product** radio button.
 - ii. Select the required product from the drop down menu.
 - b. To create a new product to add the repositories to:
 - i. Select the **New Product** radio button.
 - ii. Enter the **Name** and **Label** for the new product and select a **GPG Key** from the drop down menu.
9. Select the **Serve via HTTP** check box to serve the repository via HTTP.

10. Edit the **Name** and **Label** for the **Selected URLs**.

11. Click the **Create** button.

Result:

The repositories have been discovered and added to a product.

[Report a bug](#)

4.2.2.5. Removing a Product

This section describes how to remove products from Red Hat Satellite.

Procedure 4.7. To Remove a Product from Satellite:

1. Click **Content** → **Products**.
2. Select the check box next to the products you want to remove.
3. Click **Bulk Actions** and then click **Remove Products**.
4. Click **Remove** to confirm that you want to remove the products.

[Report a bug](#)

4.2.3. Synchronizing Content

Synchronization is the act of coordinating updates between the Red Hat Satellite repositories and the source repositories being used. It is a required step after enabling repositories, in order to populate the Red Hat Satellite with content from the source repositories.

Constant, scheduled synchronization will result in:

- Data integrity between packages
- Updated packages, security fixes, and errata

Satellite's synchronization management capabilities allow organization administrators to create synchronization plans to configure how often a host should look for and install updates. Synchronization plans are then paired with the product repositories to come up with a synchronization schedule that will allow products to be updated at specific intervals that are convenient for the organization's network.

[Report a bug](#)

4.2.3.1. Synchronization Status

**IMPORTANT**

The manual synchronization of repositories is required after enabling them. It is at this point that the local repository in the Satellite is populated by the required packages.

These steps show how to synchronize products in **Red Hat Satellite**.

Procedure 4.8. Synchronize Products

1. Click **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product channels available for synchronization is displayed.
2. Click the arrow next to the product name to see available content.
3. Select the content you wish to synchronize.
4. Click the **Synchronize Now** button to starting synchronizing. The status of the synchronization process will appear in the **Result** column. If synchronization is successful, **Sync complete** will appear in the **Result** column. If synchronization failed, **Error syncing** will appear.

Result:

A product is synchronized.

**NOTE**

Content synchronization can take a long time. The length of time required is dependent on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

[Report a bug](#)

4.2.3.2. Creating a Synchronization Plan

Regular, frequent synchronization is required to maintain data integrity between packages as well as making sure that packages are updated to the latest security fixes. Red Hat Satellite provides the ability to create scheduled synchronization plans that allow package updates at intervals convenient to the organization.

Procedure 4.9. To Create a Synchronization Plan:

1. Click **Content** → **Sync Plans**.
2. Click the **New Sync Plan** link to create a new synchronization plan.
3. Enter the **Name**, **Description** and other details for the plan.
4. Click **Save** to create the synchronization plan.

[Report a bug](#)

4.2.3.3. Applying a Synchronization Schedule

After you have created a synchronization plan, you need to associate products with that plan to create a synchronization schedule. The following procedure describes how to create a synchronization schedule in Red Hat Satellite 6.

Procedure 4.10. To Create a Synchronization Schedule:

1. Click **Content** → **Sync Plans** and select the synchronization plan you want to implement.
2. Click **Products** → **Add** in the synchronization plan main page.

3. Select the check box of the product to associate with the synchronization plan.
4. Click **Add Selected**.

[Report a bug](#)

4.3. DISCONNECTED SATELLITE

In high security environments where hosts are required to function in a closed network, disconnected from the internet, the Red Hat Satellite Server can provision systems with the latest security updates, errata, and packages. This is achieved by using two important components: the **katello-disconnected** utility and a synchronization host.

The diagram below illustrates how a disconnected Satellite is able to keep its content updated even without an internet connection. An intermediary system with an internet connection is needed to act as a synchronization host. This synchronization host is in a separate network from the Satellite server.

The synchronization host imports content from the Red Hat Content Delivery Network (CDN) through pulp. The content is then exported onto a media, such as DVDs, CDs, or external hard drives and transferred to the disconnected Satellite server. The following sections in this chapter will guide you through the whole process.

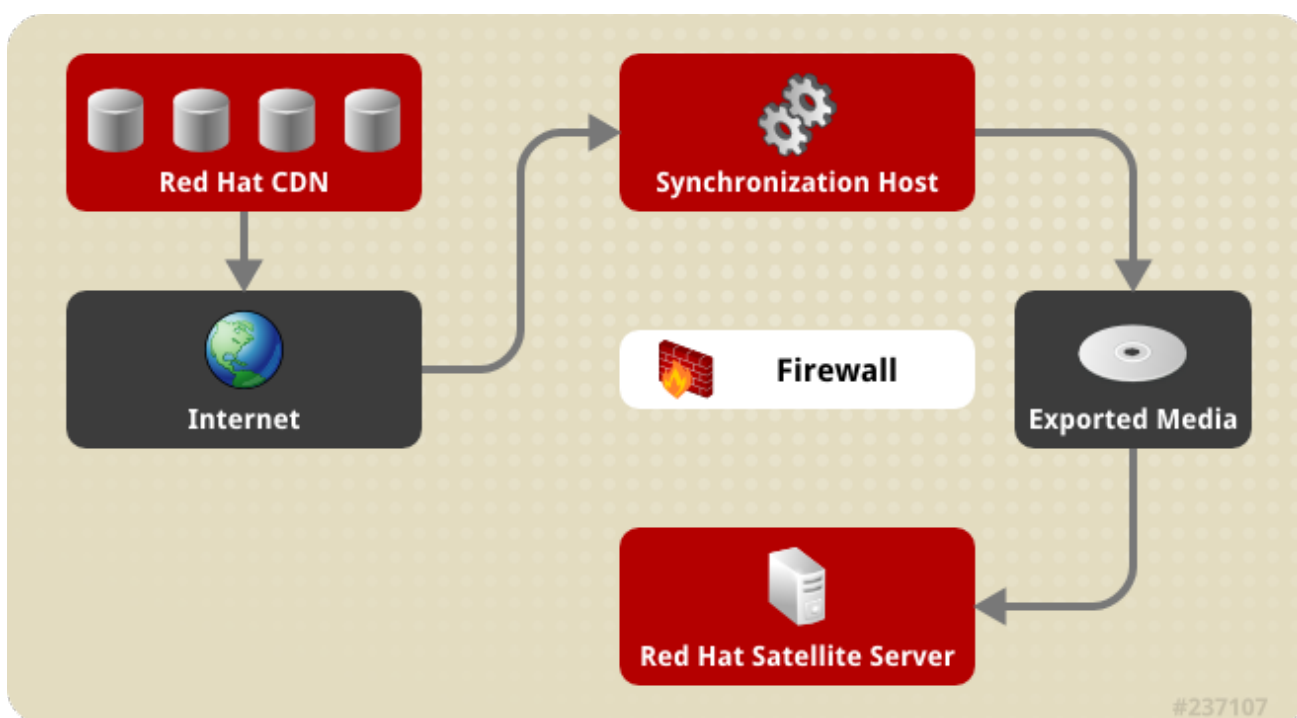


Figure 4.1. Disconnected Satellite

[Report a bug](#)

4.3.1. Configuring the Synchronization Host

Prerequisites

To import content from the Red Hat Content Distribution Network (CDN), the synchronization host requires:

- An Internet connection

- Valid Red Hat Network subscriptions
- A valid manifest (See [Section 4.2.1.1](#), “Setting up a Manifest” for instructions on how to obtain one.)

Procedure 4.11. To Configure a Host to Synchronize and Export Content from the Red Hat CDN:

1. Use Red Hat Subscription Manager to register the synchronization host to RHN.
2. List all the available subscriptions to find the correct Red Hat Satellite product to allocate to your system:

```
# subscription-manager list --available --all
```

This command displays output similar to the following:

```
+-----+
  Available Subscriptions
+-----+

ProductName:      Red Hat Satellite
ProductId:        SKU123456
PoolId:           e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Quantity:         10
Multi-Entitlement: No
Expires:          08/20/2013
MachineType:     physical
```



NOTE

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

3. Subscribe to the pool using the following command:

```
# subscription-manager subscribe --pool=Red_Hat_Satellite_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_Id
```

4. Disable all existing repositories:

```
# subscription-manager repos --disable "*"

```

5. Enable the Red Hat Satellite and Red Hat Enterprise Linux and Red Hat Software Collections repositories. Ensure the Red Hat Enterprise Linux repository matches the specific version you are using.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc1-6-rpms \
--enable rhel-6-server-satellite-6.0-rpms
```

**NOTE**

The commands above are based on Red Hat Enterprise Linux 6. If you are using a different version of Red Hat Enterprise Linux, change the repository based on your specific version.

6. Install `katello-utils` and associated RPMs:

```
# yum install python-qpidd-qmf python-qpidd qpidd-cpp-server katello-
utils
```

`katello-utils` includes the **katello-disconnected** utility that is required to set up repositories for import while `qpidd` related packages are necessary for **pulp** configuration.

7. Generate a secret 32-character alphanumeric string for the **`oauth_secret`** entry in the `/etc/pulp/server.conf` file:

```
# tr -dc "[:alnum:]" < /dev/urandom | head -c 32
randomly_generated_value
```

8. In the `/etc/pulp/server.conf`, uncomment the **`[oauth]`** entry and add the randomly generated value from the previous step as the **`oauth_secret`** value:

```
[oauth]
enabled: true
oauth_key: katello
oauth_secret: randomly_generated_value
```

9. Disable authentication in `/etc/qpidd/qpidd.conf`:

```
# Configuration file for qpidd. Entries are of the form:
# name=value
#
# (Note: no spaces on either side of '=').
# Run "qpidd --help" or see "man qpidd" for more details.

auth=no
```

All incoming connections authenticate using the Satellite's default realm.

10. Configure the connection from **katello-disconnected** to Pulp with the previously generated value as your **`--oauth-secret`** option:

```
# katello-disconnected setup --oauth-key=katello --oauth-
secret=randomly_generated_value
```

This places a configuration value in `~/.katello-disconnected`.

11. Configure Pulp on the Synchronization Server:

```
sudo service qpidd start
```

```

sudo chkconfig qpidd on
sudo service mongod start
sleep 10
sudo chkconfig mongod on
sudo -u apache pulp-manage-db
sudo service httpd restart
sudo chkconfig pulp_workers on
sudo service pulp_workers start
sudo chkconfig pulp_celerybeat on
sudo service pulp_celerybeat start
sudo chkconfig pulp_resource_manager on
sudo service pulp_resource_manager start

```

12. Import the manifest:

```
# katello-disconnected import -m ./manifest.zip
```

Importing the manifest sets up the list of available repositories to synchronize to based on the subscriptions you selected.

The synchronization host is now ready to synchronize content from the Red Hat CDN.

[Report a bug](#)

4.3.2. Synchronizing Content

By default, **katello-disconnected** enables all repositories that are included in the manifest for synchronization. Synchronization time is directly related to the amount of repositories to be synchronized. If the manifest has a large amount of repositories, the synchronization will take time and network resources.

katello-disconnected allows for the synchronization of specific repositories. This section will set up Pulp for synchronizing content.

1. Disable all repositories:

```
# katello-disconnected disable --all
```

katello-disconnected enables all repositories by default.

2. Choose which repositories you wish to sync by listing all available repositories from the manifest:

```

# katello-disconnected list --disabled
rhel-5-server-debug-rpms-5Server-i386
rhel-5-server-debug-rpms-5Server-ia64
rhel-5-server-debug-rpms-5Server-x86_64
rhel-5-server-debug-rpms-5_7-i386
rhel-5-server-debug-rpms-5_7-ia64
rhel-5-server-debug-rpms-5_7-x86_64
rhel-5-server-debug-rpms-5_8-i386
rhel-5-server-debug-rpms-5_8-ia64
rhel-5-server-debug-rpms-5_8-x86_64
rhel-5-server-debug-rpms-5_9-i386

```

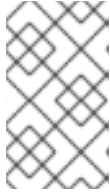
```
rhel-5-server-debug-rpms-5_9-ia64
rhel-5-server-debug-rpms-5_9-x86_64
rhel-5-server-isos-5Server-i386
```

3. Enable the chosen repositories for synchronization:

```
# katello-disconnected enable -r rhel-6-server-sam-rpms-6_4-x86_64
```

4. Create the repositories and push them to **Pulp** to allow synchronization:

```
# katello-disconnected configure
```



NOTE

The `configure` option for **katello-disconnected** reads the manifest, creates pulp repositories, and generates scripts before synchronization. It needs to be run each time a repository is enabled or disabled.

5. Synchronize the repositories:

```
# katello-disconnected sync
```

You can use the **watch** option to monitor the synchronization process.

```
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-sam-rpms-6_4-x86_64

running:
rhel-6-server-sam-rpms-6_4-x86_64
...
finished:
rhel-6-server-sam-rpms-6_4-x86_64

Watching finished
```

Content is now synchronized.

[Report a bug](#)

4.3.3. Exporting Content

Prerequisites

An external export media such as a CD, DVD, or external hard drive.

The synchronized content needs to be exported to enable importing into the disconnected Red Hat Satellite. To do so:

1. Export the synchronized repositories:

```
# katello-disconnected export -t /var/tmp/export
```

The output will look similar to:

```
# katello-disconnected export -t /var/tmp/export
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-sam-rpms-6_4-x86_64

finished:
rhel-6-server-sam-rpms-6_4-x86_64
Watching finished
Done watching ...
  Copying content to /var/tmp/export
  Archiving contents of /var/tmp/export into 4600M tar archives.
  NOTE: This may take a while.
tar: Removing leading `/' from member names

Done exporting content, please copy /var/tmp/export/* to your
disconnected host
```

This operation will create the following files in **/var/tmp/export**:

```
# ls /var/tmp/export/
content-export-00 content-export-01 content-export-02
expand_export.sh
```

2. Copy the files from `/var/tmp/export` into the external media.



NOTE

If the files are too big for your external media, the files can be copied sequentially in a series of DVDs.

The synchronized content has now been exported and ready for importing to the disconnected Satellite server.

[Report a bug](#)

4.3.4. Importing Content to a Disconnected Satellite Server

Prerequisites

Ensure that the directory and file system containing the exports has enough space to contain the extracted archives. For example, if your export is 40 GB, the disconnected Satellite Server directory and file system where you are importing the content will need an extra 40 GB of space to expand it on the same file system.

1. Copy ALL of the Satellite Content ISOs to a directory that the Satellite can access. This example uses **/root/isos**.

2. Create a local directory that will be shared via **httpd** on the Satellite. This example uses **/var/www/html/sat-import/**.

```
# mkdir -p /var/www/html/sat-import/
```

3. Recursively copy the contents of the first ISO to the local directory:

```
# mount -o loop /root/isos/first iso /mnt/iso
# cp -ruv /mnt/iso/* /var/www/html/sat-import/
# umount /mnt/iso
```

4. Repeat the above step for each ISO until you have copied all the data from the series of ISOs into the local directory **/var/www/html/sat-import/**.

5. Ensure that the SELinux contexts is correct:

```
# restorecon -rv /var/www/html/sat-import/
```

6. Change the CDN URL to reference the location that the ISOs were copied to. This example uses the Satellite fully qualified domain name (FQDN) **server.example.com**, so the URL is:

```
http //server.example.com/sat-import/
```



NOTE

The Satellite is now acting as its own CDN with the files located in **http://localhost/content**. This is not a requirement. The CDN can be hosted on a different machine inside the same disconnected network as long as it is accessible to the Satellite server via HTTP.

7. Add the CDN address to the Satellite web interface:
 - a. Log in to the Satellite web interface.
 - b. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest**.
 - c. On the **Subscription Manifest** information screen, scroll to **Red Hat Provider Details**. Click the edit icon on the **Repository URL** entry and change the entry to the CDN's repository URL.
 - d. Click **Browse** to choose the manifest file.
 - e. Click **Upload** to import your manifest.
8. Enable the repositories from the local CDN:
 - a. Click **Content** → **Red Hat Repositories**
 - b. Enable the repositories that were enabled and synchronized in the Synchronizing Content section.
9. Click **Content** → **Sync Status**.

10. Select the repositories you want to synchronize and click **Synchronize Now**.

Once the synchronize finishes, the disconnected Satellite is now ready to serve the content to client systems.

[Report a bug](#)

CHAPTER 5. USING CONTENT VIEWS

Content views are managed selections of content, which contain one or more repositories (yum / puppet) 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.

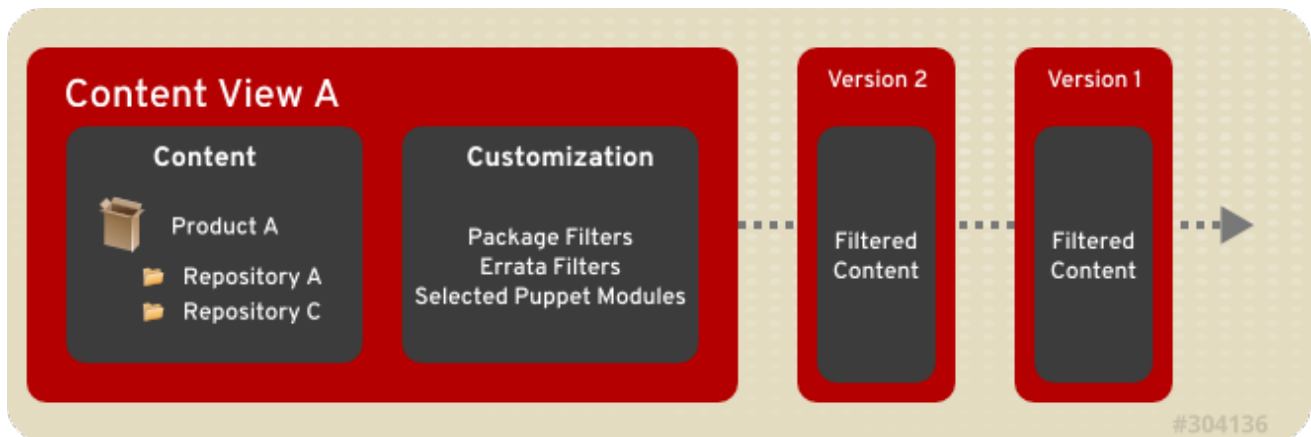


Figure 5.1. This diagram details the creation of new versions of a Content View. These content view versions are promoted along an environment path during the application life cycle.

Published content views are used with life cycle environments.

[Report a bug](#)

5.1. CREATING A CONTENT VIEW

A user with administrator privileges creates content views for use within the life cycle environments. To create a content view:

1. Log in as a Satellite administrator.
2. Click **Content** → **Content Views**.
3. Click **Create New View**.
4. Fill in the following fields:
 - **Name**
 - **Label** - this field is automatically populated when the **Name** field is filled out.
 - **Description**
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**.

[Report a bug](#)

5.2. ADDING REPOSITORIES TO THE CONTENT VIEW

A Repository provides storage for a collection of content. For example, a YUM repository, Puppet repository, or a Docker repository. Perform the following steps to associate a repository with a specific content view:

1. Click **Content** → **Content Views** and choose the Content View to add repositories to.
2. Depending on the type of content you wish 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. Once all the intended repositories have been selected, click **+Add Repositories**.

Repositories have now been added to the Content View.

[Report a bug](#)

5.3. FILTERING CONTENT

Filters are created to prevent packages from being promoted to subsequent environments. Package names or regular expressions are added to the filter to create the rules to blacklist packages and the filter is then associated to entire products or individual repositories within any product.

5.3.1. Creating a Filter

These steps show how to create a filter.

Procedure 5.1. Creating a Filter

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click on the **+New Filter** button.
5. Type in the name of the new filter in the **Name** field.
6. Choose a content type from the **Content Type** drop down menu.
7. Choose whether the filter includes or excludes the selected content type by selecting the **Type** drop down menu.

8. Optionally, enter a description in the **Description** field.
9. Click the **Save** button to save your new filter.

Result:

A filter is created.

[Report a bug](#)

5.3.2. Adding Content to a Filter

Prerequisites

Requires a created Filter.

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click a created package filter's name. Depending on the type of filter selected, the readout will be different.
 - a. If the filter is made for Packages:
 - i. Enter a package name and select a **Detail** value from the dropdown menu. Click **+Add** to add the package to the filter.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click **Update Repositories**.
 - b. If the filter is made for package groups:
 - i. Click on the **Add** subtab, and choose the desired package group. Click the **+Add Package Group** button.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
 - c. If the filter is made for Errata:
 - i. Click on the **Add** subtab. Check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then choose a start date and end date. Click the **+Add Errata** button.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
 - d. If the filter is made for Errata - Date and Type:

- i. Under the **Erratum Date Range** subtab, check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then choose a start date and end date. Click the **Save** button.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
5. Click the **Publish New Version** button. Enter a comment if desired, then click the **Save** button.

Result

Packages are added to the filter.

[Report a bug](#)

5.3.3. Removing Content from a Filter

Prerequisites

Requires a created Filter.

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click a created package filter's name. Depending on the type of filter selected, the readout will be different.
 - a. If the filter is made for Packages:
 - i. Click the **Packages** subtab then click the **Package Name** checkbox next to the package to remove. Click the **Remove Packages** button to remove the package from the filter.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
 - b. If the filter is made for package groups:
 - i. Click the **List/Remove** subtab then click the **Name** checkbox next to the package group to remove. Click the **Remove Package Group** button to remove the package group from the filter.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
 - c. If the filter is made for Errata:
 - i. Click the **List/Remove** subtab then click the **Errata ID** checkbox next to the errata to remove. Click the **Remove Errata** button to remove the errata from the filter.

- ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
 - d. If the filter is made for Errata - Date and Type:
 - i. Under the **Erratum Date Range** subtab, check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then edit the start date and end date. Click the **Save** button.
 - ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
5. Click the **Publish New Version** button. Enter a comment if desired, then click the **Save** button.

Result

Packages are removed from the filter.

[Report a bug](#)

5.3.4. Removing a Filter

These steps show how to remove a filter.

Procedure 5.2. Remove a Filter

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click the checkbox next to the name of the package filter you wish to remove.
5. Click the **Remove Filters** button.

Result:

A filter is removed.

[Report a bug](#)

5.4. PUBLISHING A CONTENT VIEW

Once 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.

To publish a content view definition:

1. Click **Content** → **Content Views**.
2. Click on the content view to be published.

3. Click the **Publish New Version** button.
4. Fill in a comment.
5. Click the **Save** button.

A published content view is now available.

[Report a bug](#)

CHAPTER 6. SEARCHING FOR CONTENT

6.1. BASIC CONTENT SEARCH

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

Procedure 6.1. 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**.

[Report a bug](#)

6.2. CONTENT COMPARISON ACROSS ENVIRONMENTS

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

Procedure 6.2. 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.

[Report a bug](#)

CHAPTER 7. 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 5, *Using Content Views*](#) for more information on content views.

To apply the latest updates, make sure you have correctly synchronized the Satellite content. For more information on synchronizing content, see [Section 4.2.3, “Synchronizing Content”](#) for a connected Satellite and [Section 4.2.3, “Synchronizing Content”](#) for a disconnected Satellite. 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, two keywords describe errata with regard to their relationship to the available content hosts:

- **Applicable**: errata 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**: errata 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 that have permissions to manage content hosts, but are not entitled for errata management at higher levels.

7.1. 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 7.1. To Inspect Available Errata:

1. Navigate to **Content** → **Errata** to view the list of available errata.
2. Use the filtering tools on the top of the page to limit the number of displayed errata:

- o Select the repository to be inspected from the drop-down list. **All Repositories** is selected by default.
- o 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.
- o To search the table of errata, type the query in the **Search** field in the form of:

```
parameter operator value
```

See [Table 7.1, “Parameters Available for Errata Search”](#) for the list of parameters available for search, find the list of applicable operators in [Table 16.2, “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 **Errata ID** of the errata you want to inspect:

- o The **Details** tab contains the description of the updated package as well as documentation of important fixes and enhancements provided by the update.
- o On the **Content Hosts** tab, you can apply the errata to selected content hosts as described in [Section 7.2, “Applying Errata to Content Hosts”](#).
- o The **Repositories** tab lists repositories that already contain the errata. You can filter repositories by the environment and content view, and search for them by the repository name.

Table 7.1. 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 <i>"Feb 16,2015"</i> , or use keywords, for example <i>"Yesterday"</i> , or <i>"1 hour ago"</i> . The time range can be specified with the use of the "<" and ">" operators.	issued < "Jan 12, 2015"
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

Parameter	Description	Example
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. One of <i>Critical</i> , <i>Important</i> , <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"

7.2. APPLYING ERRATA TO CONTENT HOSTS

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

Procedure 7.2. To Apply a Single Errata to Content Hosts:

1. Navigate to **Content** → **Errata** to view the list of available errata.
2. Click **Errata ID** of the errata 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 on the top of the page, only the content hosts that already have the installable errata in their life cycle environment are displayed.
4. Click **Apply to Hosts**.
 - If the errata 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 errata on the content host when promoting the updated content view. Otherwise, errata 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 7.4, “Applying Installable Errata to a Content Host”](#).
 - If the errata is *installable*, which means it is already present on selected content hosts but was not installed yet, no new content view version is created.
5. Click **Confirm**.
 - If the errata is *applicable*, a new task is started for the update procedure. After the task is completed successfully, you can investigate the updated content host at **Host** → **Content Hosts**.
 - If the errata is *installable*, it is installed on selected content hosts. You can investigate the updated content host at **Host** → **Content Hosts**.

Procedure 7.3. To Apply Multiple Errata to Content Hosts:

Procedure 7.3. 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 on 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 all selected errata are installable, they are installed without creating a new content view version.

Procedure 7.4. Applying Installable Errata to a Content Host

If the content host's life cycle environment contains installable errata, you can install them from the **Content Hosts** page. This way, errata can be applied by users that have permissions to manage content hosts, but are not entitled for errata management at higher levels.

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

7.3. SUBSCRIBING TO ERRATA NOTIFICATIONS

The following procedure shows how to configure email notifications about errata.

Procedure 7.5. To Configure Errata Notifications:

1. Navigate to **Administer** → **Users**.
2. Click 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 relevant to errata management:
 - **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**.

[Report a bug](#)

CHAPTER 8. 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 [Get Started with Docker Formatted Container Images on Red Hat Systems](#) on the Red Hat Customer Portal.

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

- **Container** (in the narrow sense of the word) is 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** is a static snapshot of the containers' configuration. Image is a read-only layer that is never modified, all changes are made in top-most writable layer, and can be saved only by creating a new image. 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 read-only, so any changes are reflected in the copied images stacked on top of it. See [Get Started with Docker Formatted Container Images on Red Hat Systems](#) for information on how to access Red Hat Enterprise Linux platform images. See [Example 8.1, "Creating a Red Hat Enterprise Linux Container in Satellite"](#).
- **Registry** is 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 8.1.1, "Creating a New Container"](#)
- **Tags** are added to images to differentiate them from similar images in a repository. In practice, they typically mark versions of the application inside the image. **Repositories** are used to group similar images together in a container registry. Images only have unique alphanumeric identifiers, so repositories provide a way to name images (by using tags). Naming in form or *repository:tag* is a human-readable way of identifying images. See [Section 8.5, "Using Tags"](#), or [Section 8.2, "Working with Repositories"](#) for details.

With Red Hat Satellite, you can create an on premise registry, import images from various sources and distribute them to containers using content views. Satellite provides a Docker compute resource, that acts as a server for running containers. This way, you can import an image, start a container based on this image, monitor the container's activity, and commit it's state to a new image layer that can be further propagated. For more information on loading images to a content view, see [Section 5.2, "Adding Repositories to the Content View"](#).

8.1. MANAGING CONTAINERS

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

8.1.1. Creating a New Container

In Red Hat Satellite, you can deploy containers only on a compute resource of the Docker provider type. Therefore, when attempting to view or create containers for the first time, you are prompted to create the Docker compute resource. To do so, follow the steps described in [Section 11.3.4, "Compute Resources"](#).

Once there is at least one Docker compute resource present on your Satellite, you can create and view containers. To create a new container, follow the steps described in [Procedure 8.1, “Creating a Container”](#). For instructions on how to investigate the already created containers, see [Section 8.1.2, “Investigating Containers”](#).

To create a container, you first have to 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 **Satellite** 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 8.2, “Working with 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 8.3, “Adding an External Registry”](#).



NOTE

Note that you can not change the container configuration once the container is created. To alter the configuration, you have to create a replacement container with modified settings as described in [Procedure 8.1, “Creating a Container”](#). Therefore, make sure that containers can be easily replaced in your workflow.

Procedure 8.1. Creating 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. See [Section 11.3.4, “Compute Resources”](#) for more information on 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 **Satellite** tab to import an 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 an image from the Docker Hub registry. Once you type an image name to the **Search** field, Satellite automatically searches the compute resource first. Click the looking glass icon to search the Docker Hub. If the image is found,

Satellite displays the image metadata and populates the **Tag** field with tags available for the selected image name.

- Select the **External registry** tab to import an 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. See [Section 8.3, “Adding an External Registry”](#) for details.

Click **Next** to proceed.

4. In the **Configuration** stage of container creation set the following parameters:

- Select a name for the container.
- 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, the newly created container is inactive, for instructions how to start it see [Procedure 8.3, “Starting and Stopping a Container”](#).



NOTE

Red Hat Satellite currently supports only Red Hat Enterprise Linux 7 container hosts.

Example 8.1. 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 8.3, “Adding an External Registry”](#). Specify `registry.access.redhat.com` as a registry URL and insert your access credentials for the Red Hat Customer Portal.
2. Create a new container as described in [Section 8.1.1, “Creating a New Container”](#). In the **Image** stage of container creation, navigate to the **External registry** tab and select the registry created in the previous step from the drop-down list. 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.

8.1.2. Investigating Containers

Red Hat Satellite provides means to monitor the status of containers as well as processes running inside of 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 view a list of containers present in the current organization and how to investigate the container metadata.

Procedure 8.2. Investigating 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 only the initial standard output produced when the container started.

8.1.3. Starting, Committing, and Removing Containers

A new container is by default disabled. By enabling a container, you start the processes of the containerized application in the compute resource that acts as a container server. Hosts are then able to communicate with the container as with a web application. The following procedure shows how to start a container:

Procedure 8.3. Starting and Stopping 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.

By committing a container, you create a new image layer that stores the status of the container. The following procedure shows how to commit the container to an image.

Procedure 8.4. Committing 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 then prompts you to provide the following information:

- 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, when you commit this container, the changes are pushed back to the Docker Hub.

Procedure 8.5. Removing 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.

8.2. WORKING WITH REPOSITORIES

This section shows how to create an internal repository for container images. You can use such repository to create containers as described in [Section 8.1.1, “Creating a New Container”](#).

8.2.1. Creating a Repository

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 selected product, follow the steps described in [Procedure 4.4, “Adding Repositories to a Product”](#). Select **docker** as a repository type.

8.2.2. Uploading Images to a Repository

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

Procedure 8.6. Uploading Images to a Repository

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

6. Click **Upload**

8.3. ADDING AN EXTERNAL REGISTRY

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

Procedure 8.7. Adding an External Registry

1. Navigate to **Containers** → **Registries**.
2. Click **New Registry**.
3. On the **Registry** tab, specify the following parameters:
 - Specify the name of the registry. This setting is required.
 - Specify the URL of the registry. This setting is required.
 - Provide a brief description of the registry.
 - Specify a user name if required for accessing the registry.
 - Specify a password if required for authentication to the registry.
4. On the **Locations** tab, select the locations where the new registry will be available.
5. On the **Organizations** tab, select the organizations where the new registry will be available.
6. Click **Submit** to create the registry.

8.4. IMPORTING IMAGES TO A COMPUTE RESOURCE

Importing an image is a necessary step for creating any container as described in [Procedure 8.1, “Creating a Container”](#). You can also import images to a compute resource before creating a container as described in the following procedure.

Procedure 8.8. Adding Images to a Compute Resource

1. Navigate to **Infrastructure** → **Compute resources** to view a list of compute resources.
2. Select the compute resource you want to edit. The compute resource must be of type Docker.
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**.

8.5. USING TAGS

Tags are convenient for organizing images, especially when you operate with several versions of an image. A combination of a repository name and a tag in the form of *repository:tag* is a human-readable alternative to an image ID.

Procedure 8.9. Searching Registries by Tags

1. Navigate to **Content** → **Docker tags**.
2. Use the search field to filter tags by the **image** name, tag **name**, 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
```

See [Table 16.2, “Supported Operators for Granular Search”](#) for the list of alternative comparison operators. By default, the search field recognizes the input string as a tag name. For example, the following query searches for all *centos* tags:

```
centos
```

3. Click the name of the tag you want to observe. 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.

CHAPTER 9. CONFIGURING ACTIVATION KEYS

Activation Keys are preset keys used when registering the host and define:

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

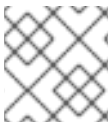
[Report a bug](#)

9.1. CREATING AN ACTIVATION KEY

This section describes how to create an activation key.

Procedure 9.1. 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.
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 on the **Details** tab of the Activation Key.

[Report a bug](#)

9.2. REMOVING AN ACTIVATION KEY

This section describes how to remove an activation key.

Procedure 9.2. 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** detail panel, click **Remove**.
4. In the alert box, click **Remove** to confirm that you want to remove the key.

[Report a bug](#)

9.3. ADDING OBJECTS TO ACTIVATION KEYS

This section describes how to add different types of objects to activation keys.

9.3.1. Adding Subscriptions to an Activation Key

This section describes how to add subscriptions to an activation key.

Procedure 9.3. To Add a Subscription to an Activation Key:

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

[Report a bug](#)

9.3.2. Adding Host Collections to an Activation Key

These steps show how to add host collections to an activation key.

Procedure 9.4. 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**.

[Report a bug](#)

9.4. REMOVING OBJECTS FROM ACTIVATION KEYS

9.4.1. Removing Subscriptions from an Activation Key

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

Procedure 9.5. Remove Subscriptions from an Activation Key

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

Result:

Subscriptions are removed from your activation key.

[Report a bug](#)

9.4.2. Removing Host Collections from an Activation Key

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

Procedure 9.6. 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 wish to remove host collections from.
3. Click the **Host Collections** subtab.
4. A list of host collections attached to the Activation Key is displayed. Tick the checkbox of the host collections you wish to remove.
5. Click the **Remove Selected** button to remove host collections from the activation key.

Result:

Host collections are removed from your activation key.

[Report a bug](#)

CHAPTER 10. 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.

[Report a bug](#)

10.1. CREATING A GPG KEY

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

Procedure 10.1. 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.

[Report a bug](#)

10.2. REMOVING A GPG KEY

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

Procedure 10.2. 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.

[Report a bug](#)

CHAPTER 11. CONFIGURING THE PROVISIONING ENVIRONMENT

11.1. 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 11.1. 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 11.1. 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.

Submenu	Options	Description
	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 Environment	Defines the activation keys made available in templates as <code>@host.params['kt_activation_keys']</code> .

[Report a bug](#)

11.2. 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**.

[Report a bug](#)

11.2.1. Creating a Global Simple Parameter

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

1. Click **Configure** → **Global Parameters**.
2. Click the **New Parameter** button.
3. Type a **Name** for the parameter's key.
4. Type a **Value** for the parameter.
5. Click the **Submit** button.

[Report a bug](#)

11.2.2. Creating a Puppet Class

This procedure shows how to add a new Puppet Class to Satellite.

1. Click **Configure** → **Puppet Classes**.
2. Click the **New Puppet class** button.
3. Type a **Name** for the Puppet Class.
4. Type a **Puppet Environments** for the Puppet Class.
5. Select one or more **Host groups** that own the Puppet Class.
6. Click **Submit**.

[Report a bug](#)

11.2.3. Configuring Smart Parameters

The following procedure configures smart parameters in a Puppet class.

Procedure 11.2. 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.

Result:

Satellite configures the smart parameter.

[Report a bug](#)

11.2.4. Importing Parameterized Classes from a Puppet Master

The following procedure imports parameterized classes from your Puppet Master.

Procedure 11.3. To import parameterized classes**NOTE**

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

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.

Result:

Satellite imports the Puppet Master's parameterized classes.

[Report a bug](#)

11.2.5. Configuring Parameterized Classes

The following procedure configures parameterized classes.

Procedure 11.4. 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 the a parameter from the left-hand list.
5. Edit the **Description** text box to add any plain text notes.
6. Click the **Override** checkbox to allow Satellite control over this variable. If the checkbox 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.

Result:

Satellite configures the parameters for the class.

[Report a bug](#)

11.3. CONFIGURING PROVISIONING SETTINGS

11.3.1. 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.

11.3.1.1. Creating a Domain

This procedure shows how to add a Domain.

Procedure 11.5. Creating a Domain

1. Click **Infrastructure** → **Domains**.
2. Click the **New Domain** button.
3. Type a **Name** for the Domain. This is the DNS domain name.
4. Type a **Description** for the Domain.
5. Choose a DNS-enabled Red Hat Satellite Capsule Server from the **DNS Capsule** selection box.
6. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
7. Click the **Organizations** tab and click 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.

8. Click **Submit**.

Satellite creates a Domain and registers it on the DNS server configured with the selected Smart Proxy.

[Report a bug](#)

11.3.2. 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.

11.3.2.1. Creating a Subnet

The following procedure shows how to create a Subnet:

Procedure 11.6. Creating a Subnet

1. Click **Infrastructure** → **Subnets**.
2. Click the **New Subnet** button.

3. Type a **Name** for the Subnet.
4. Type the IP address for the Subnet into the **Network** box.
5. Type the mask for the network's IP address into the **Network mask** box.
6. Type the **Gateway address** for the Subnet.
7. Type the **Primary DNS server** and **Secondary DNS server** for the Subnet.
8. Define the IP assignment range with the **Start of IP range** and **End of IP range** fields.
9. Define the **VLAN ID** for the subnet.
10. Select the applicable domain for the subnet from the **Domains** tab.
11. Click the **Capsules** tab, and select a Red Hat Satellite Capsule Server for the **DHCP Proxy**, **TFTP Proxy**, and **DNS Proxy** services.
12. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
13. Click the **Organizations** tab and click 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.

14. Click the **Submit** button.

[Report a bug](#)

11.3.3. 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.

11.3.3.1. Creating an Architecture

This procedure shows how to create an architecture.

Procedure 11.7. To Create an Architecture:

1. Click **Hosts** → **Architectures** and then click **New Architecture**.
2. Type 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**.

[Report a bug](#)

11.3.4. 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.

11.3.4.1. Creating a Compute Resource

The following procedure shows how to add a Compute Resource.

Procedure 11.8. Creating a Compute Resource

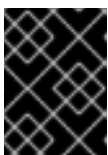
1. Navigate to **Infrastructure** → **Compute Resources**.
2. Click **New Compute Resource**.
3. Type a **Name** for the Compute Resource.
4. Select a **Provider** type.
5. Optionally, enter a **Description**.
6. Depending on the provider type chosen, the next few fields ask for authentication and datacenter details. Refer to the following table for more information about each provider type.

Table 11.2. 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.

Type	Description
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.

- Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
- Click the **Organizations** tab and click 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.

- Click **Submit**.

[Report a bug](#)

11.3.5. 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.

11.3.5.1. Creating a Hardware Model

This procedure shows how to add a Hardware Model.

Procedure 11.9. Creating a Hardware Model

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

[Report a bug](#)

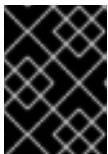
11.3.6. Installation Media

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

11.3.6.1. Adding Installation Media

This procedure shows how to add new Installation Media to Satellite.

1. Click **Hosts** → **Installation Media**.
2. Click the **New Installation Medium** button.
3. Type a **Name** for the Installation Media.
4. Type a **Path** to the Installation Medium. Options include either a URL or a valid NFS server.
5. Select an **Operating System Family** to define the Installation Media's type.
6. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
7. Click the **Organizations** tab and click 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.

8. Click the **Submit** button.

[Report a bug](#)

11.3.7. 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.

11.3.7.1. Adding an Operating System

This procedure shows how to add a Operating System to Satellite.

Procedure 11.10. Adding an Operating System

1. Click **Hosts** → **Operating Systems**.
2. Click the **New Operating system** button.
3. Type a **Name** for the Operating System.
4. Define the **Major Version** of the Operating System.
5. Define the **Minor Version** of the Operating System.
6. Select the **OS Family** to define the Operating System type.
7. Select the **Architectures** from the list of available Architectures. If none are available, create and assign them under **Hosts** → **Architectures**.
8. Click the **Partition tables** tab, then add the applicable file system layouts from the list.
9. Click the **Installation Media** tab, then add the applicable file system layouts from the list.
10. Click the **Submit** button.

[Report a bug](#)

11.3.8. 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.

11.3.8.1. Defining a New Partition Table

This procedure shows how to define a new Partition Table for new installations.

1. Click **Hosts** → **Partition Tables**.
2. Click the **New Partition Table** button.
3. Type a **Name** for the partition table.
4. Enter the **Layout** for the Partition Table. The **Layout** textbox also accepts dynamic disk partitioning scripts.
5. Select the operating system from the **OS Family** tab to define the Operating System type for the partitions.

6. Click the **Submit** button.

[Report a bug](#)

11.3.9. 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.

11.3.9.1. Creating a Provisioning Template

This procedure shows how to create a Provisioning Template.

Procedure 11.11. Creating a Provisioning Template

1. Click **Hosts** → **Provisioning Templates**.
2. Click the **New Template** button.
3. Type a **Name** for the template.
4. Enter your template in the **Template editor** field. Alternatively, upload your template with the **Template** file browser below the **Template editor** textbox. This replaces the content in the **Template editor** field with the content of your chosen file.
5. Enter 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.
6. Click the **Type** tab, then select **Snippet** to store the template code without defining it as particular script or template type, or select the type from the **Type** dropdown menu.
7. Select the **Association** tab to associate the template to Hostgroups, Environments and Operating Systems. Select the operating systems from the **Applicable Operating Systems** list. Click the **Add Combination** button and select a **Hostgroup** and **Environment** to limit the template's use.
8. Click the **Submit** button.

[Report a bug](#)

11.4. 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.

11.4.1. 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.

11.4.2. 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 11.1, "Using Conditional Expressions with Facts"](#) for an example of using conditional expressions.

Example 11.1. 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^{\[1\]}](#) and [Core Facts^{\[2\]}](#) for more information.

11.4.2.1. 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 (**factor -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 11.12. 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.

11.4.2.2. 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 11.13. 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.

[Report a bug](#)

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

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

CHAPTER 12. 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. Once 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.

12.1. CREATING A HOST

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

Procedure 12.1. To Create a Host:

1. Click **Hosts** → **New Host** to open the **New Host** page.
2. On the **Host** tab, enter the required details.
3. On the **Network** tab, 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.
4. Enter the **Primary Interface** details. The **MAC Address** setting is required. Select a subnet from the drop-down menu, and specify an IP address. If there is a DHCP-enabled Capsule Server on the selected subnet, the IP address is automatically suggested. Click **IP address auto-suggest** to automatically select an address.
5. It is possible to include an additional network interface by clicking **Add Interface**. See [Section 12.4, “Configuring Additional Network Interfaces”](#) for details.
6. 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.
7. 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.
8. On the **Additional Information** tab, enter any required information.
9. Click **Submit** to complete your provisioning request.

[Report a bug](#)

12.2. CONFIGURING HOSTS FOR REGISTRATION

Red Hat Enterprise Linux hosts register to 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. Address the following requirements before proceeding:

- Hosts must be the following Red Hat Enterprise Linux Version:
 - 5.8 or later

- 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:

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

[Report a bug](#)

12.3. CONFIGURATION OPTIONS

12.3.1. Automated Configuration

These steps show how to automatically configure your client system to register to **Red Hat Satellite**.

1. Take note of the **Red Hat Satellite** hostname or the fully qualified domain name (fqdn).
2. Open a terminal console and login as root on the command line.
3. Download and install a copy of the CA Certificate for **Red Hat Satellite**:

```
yum -y --nogpgcheck install http://[hostname]/pub/katello-ca-consumer-latest.noarch.rpm
```

IMPORTANT

yum in Red Hat Enterprise Linux 5 does not support installation via HTTP. If registering a Red Hat Enterprise Linux 5 client, download the RPM package first and then run **yum** on the package. For example:

```
# wget http://[hostname]/pub/katello-ca-consumer-latest.noarch.rpm
# yum -y --nogpgcheck install 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.

12.3.2. Manual Configuration

These steps show how to manually configure your client system to register to **Red Hat Satellite**.

1. Make the following changes in `/etc/rhsm/rhsm.conf`:

```
[server]
hostname =[satellite_fqdn]

[rhsm]
baseurl=https://[fqdn_pulp]/pulp/repos/
repo_ca_cert = %(ca_cert_dir)scandlepin-local.pem
ca_cert_dir = /etc/rhsm/ca/
```

2. Change directories to the `ca` directory, remote copy and move the `andlepin-ca.crt` certificate:

```
# cd /etc/rhsm/ca
scp [satellite.fqdn]:/etc/andlepin/certs/andlepin-ca.crt .
mv andlepin-ca.crt andlepin-local.pem
```

[Report a bug](#)

12.4. CONFIGURING ADDITIONAL NETWORK INTERFACES

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 12.1, “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 12.4.2, “Adding a Virtual Interface”](#), or [Section 12.4.1, “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 channel 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 12.4.3, “Adding a Bonded Interface”](#) for details.
- **BMC:** *Baseboard Management Controller* (BMC) allows you to remotely monitor and manage physical state of machines. See [Section 15.5, “Using Power Management Features on Managed Hosts”](#) for more information on BMC, and [Section 12.4.4, “Adding a 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.

12.4.1. Adding a Physical Interface

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

Procedure 12.2. Adding 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 12.4, “Adding a Bonded Interface”](#)), VLANs and aliases (in the **Attached to** field, see [Procedure 12.3, “Adding a Virtual Interface”](#)).
7. Specify the **DNS name** associated with the host's IP address. Satellite saves this name in the "DNS A" and "DNS PTR" fields in the Capsule Server associated with the selected subnet. 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.
12. Select the **Virtual NIC** check box to create a virtual interface. See [Section 12.4.2, “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.

12.4.2. Adding a Virtual Interface

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

Procedure 12.3. Adding 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 12.4.1, “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.

12.4.3. Adding a Bonded Interface

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

Procedure 12.4. Adding 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 12.4.1, “Adding a Physical Interface”](#). Bonded interfaces use IDs in the form of *bond0* in the **Identifier** field. It is sufficient if you specify just 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 12.1, “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 `miiimon=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 12.1. 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.
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.

12.4.4. Adding a BMC Interface

The following steps show how to configure and additional BMC interface for a host. See [Section 15.5, “Using Power Management Features on Managed Hosts”](#) for more information on BMC.

Procedure 12.5. Adding 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 12.4.1, “Adding a Physical Interface”](#).
6. Specify the configuration options specific to BMC interfaces:
 - **Username, Password:** Here you can specify authentication credentials required by BMC.

- **Provider:** Specify the BMC provider, currently the only supported provider is the Intelligent Platform Management Interface (IPMI).

7. Click **OK** to save the interface configuration. Then click **Submit** to apply the changes to the host.

[Report a bug](#)

12.5. REGISTRATION

12.5.1. 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**.

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

- Ensure that all steps in [Section 12.2, “Configuring Hosts 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 9.1, “Creating an Activation Key”](#) for instructions on creating an activation key.

Procedure 12.6. Registering Systems

1. Open a terminal console and login as root 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

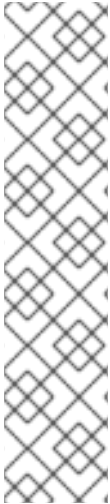
Activation keys will allow you to add environments, provisioning templates and dictate what subscriptions are available and should be applied to the registering system.

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 version 6.0. To ensure that it is pointing to the 6.3 repository, please follow these steps:

1. On **Red Hat Satellite**, select **Hosts** → **Content Hosts**.
2. Select the system that needs to be changed and click **Edit**.
3. Click on the **Operating System** tab.
4. Select '6.3' from the **Operating system** drop-down menu.
5. Click **Save**.

12.5.2. Installing the Katello Agent

Prerequisites

The Red Hat Common repository must be enabled in the Red Hat Satellite Server as it provides the required packages.

These steps show how to install and enable the Katello agent. The katello-agent must be enabled so that the Red Hat Satellite Server or Capsule Server can provide information about errata that are applicable for the system.

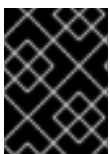
To install katello-agent:

1. Open a terminal console and log in as root on the command line.
2. Install the katello-agent using the following command:

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

12.5.3. 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.



IMPORTANT

The puppet package is part of the Red Hat Common repository. Ensure you enable this repository before you proceed.

Procedure 12.7. Installing and Enabling the Puppet Agent

1. Open a terminal console and log in as root.
2. Install the Puppet agent:

```
# yum install puppet
```

3. Configure the puppet agent to start at boot:

- o On Red Hat Enterprise Linux 6:

```
# chkconfig puppet on
```

- o On Red Hat Enterprise Linux 7:

```
# systemctl enable puppet
```

Procedure 12.8. 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 Red Hat Common 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

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

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**.



NOTE

Once the Puppet agent is configured on the host it will be listed under **All Hosts** but only when **Any Context** is selected as the host will not be assigned to an organization or location. To assign the host to an organization see [Section 3.1.3, “Editing an Organization”](#) and to assign the host to a location see [Section 3.2.2, “Editing a Location”](#).

[Report a bug](#)

12.6. REMOVING A HOST

To remove a host from Red Hat Satellite:

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.

[Report a bug](#)

CHAPTER 13. USING THE FOREMAN DISCOVERY PLUG-IN

The Foreman Discovery plug-in in Red Hat Satellite Server adds Metal-as-a-Service (MaaS) features. Bare-metal hosts on Satellite Server-managed networks can be booted over the network through PXE into stripped-down Red Hat Enterprise Linux systems running from memory to collect and send hardware facts to the Satellite Server.

After they have booted, these systems appear as discovered hosts on the Satellite Server. Administrators can use these collected hardware facts to provision systems, removing the need to manually collect MAC addresses and other hardware information and reducing the time required to provision hosts.

After the provisioning configuration is complete, the Foreman discovery plug-in sends a reboot command to the discovered host and the installation begins. The host then moves from the **Discovered Hosts** list in the Satellite Server to the **All Hosts** list.

13.1. INSTALLING THE FOREMAN DISCOVERY PLUG-IN

This section describes the prerequisites and installation process for the Foreman Discovery plug-in.

Prerequisites

Ensure your deployment satisfies the following requirements before installing the Foreman Discovery Plug-in:

- Red Hat Satellite Server version 6.0.5 or later with bare-metal provisioning configured.
- At least one Capsule Server with DHCP and TFTP services enabled.
- At least one host available for discovery, with at least 1 GB of RAM installed.

Run the following command as the **root** user to install the Foreman Discovery plug-in:

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

[Report a bug](#)

13.2. CONFIGURING THE FOREMAN DISCOVERY PLUG-IN

Prerequisites

Ensure the following are correctly configured before you configure the Foreman Discovery plug-in:

- Ensure that bare-metal provisioning is functional. See [Chapter 11, *Configuring the Provisioning Environment*](#) for more information. To test bare-metal provisioning, create a host entry with a MAC address and power on the system to provision it. After Foreman Discovery is configured, host creation will be automated.
- The network DHCP server must be able to serve unknown clients and it must point unknown clients to the TFTP server where the discovery image is stored. If Internet Systems Consortium (ISC) DHCP was installed via the capsule installer on the Red Hat Enterprise Linux host, this will already be configured. If you are using a different configuration or a different DHCP server, ensure the **nextserver** option is configured to return the correct TFTP server, which is under the Red Hat Satellite Server's control.
- Satellite 6.0 has a template locking capability which prevents some templates from being edited

or renamed because the application relies on their presence and name. The locked default PXELinux template prevents the configuration of Discovery. To circumvent this, start a Rails console on the Satellite Capsule and unlock the template, as described below:

```
# foreman-rake console
> ct = ConfigTemplate.find_by_name("PXELinux global default");
ct.locked = false; ct.save!
> exit
```

Procedure 13.1. To Configure the Foreman Discovery Plug-in:

1. Click **Hosts** → **Provisioning templates** and edit the PXELinux global default template. Add the following entries at the end of the template:

```
LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/foreman-discovery-image-latest.el6.iso-vmlinuz
APPEND rootflags=loop initrd=boot/foreman-discovery-image-latest.el6.iso-img root=live:/foreman.iso rootfstype=auto ro
rd.live.image rd.live.check rd.lvm=0 rootflags=ro crashkernel=128M
elevator=deadline max_loop=256 rd.luks=0 rd.md=0 rd.dm=0 rd.bootif=0
rd.neednet=0 nomodeset selinux=0 stateless foreman.url=FOREMAN_URL
IPAPPEND 2
```

The **foreman.url** option on the APPEND line identifies the location of the Foreman instance. Ensure that this is set correctly in global settings or the discovered hosts will not register to Foreman.

The **IPAPPEND 2** option is important because it adds the **BOOTIF=MAC** option. This is reported by Facter as **discovery_bootif**, which is a key fact used for provisioning. It is expected that the interface that it is booted from will be the provisioning interface as well. The same interface is used as the primary interface and is used to retrieve DNS configuration from DHCP.

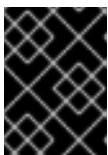
2. Set the **ONTIMEOUT** parameter to **discovery** to make the **foreman.url** option the default:

```
ONTIMEOUT discovery
```

Alternatively, the discovery image can search for a DNS SRV record called **_xforeman._tcp**. If the DNS server is configured for this, do not configure the **foreman.url** option because it will override the DNS SRV record. The following is an example of the configuration for the ISC DNS server:

```
_xforeman._tcp SRV 0 5 443 foreman
```

3. Click **Hosts** → **Provisioning templates** and then click **Build PXE Default** to deploy the configuration file on the TFTP server.



IMPORTANT

The Foreman Discovery image does not support SELinux. Set **selinux=0** in the template. The discovery image is read-only.

[Report a bug](#)

13.3. EDITING THE DISCOVERED HOST

An unknown host is not visible in the Red Hat Satellite Server web interface until the unknown host boots, reports, and registers to Foreman. Only then is the host discovered and available on the Satellite Server interface for editing. At the time of reporting, the discovered host will provide Foreman with system details which populates the Discovered Host's profile.

When the discovered host appears in the web UI, edit the discovered host's provisioning profile to allow the Satellite Server to provision the host with the correct requirements.

Procedure 13.2. Editing the Discovered Host

1. Click **Hosts** → **Discovered Hosts**.
2. Select the host and click **Provision**.
3. Edit the required details and then click **Save**.

After you have saved the changes to the provisioning details, the discovered host will reboot and install the chosen operating system with the correct requirements.

[Report a bug](#)

13.4. TROUBLESHOOTING THE FOREMAN DISCOVERY PLUG-IN

This section provides information about troubleshooting the Foreman Discovery plug-in. For example, instances where the unknown hosts booting up and failing to register with Foreman have a number of common causes.

- If the machine fails to boot to the correct image, verify that the `/var/lib/tftp/pxelinux.cfg/default` file has been configured as described in [Section 13.2, “Configuring the Foreman Discovery Plug-in”](#)
- If the machine booted the correct image but failed to contact Foreman, check the `foreman.url` option on the PXELinux Template. Check that the DNS is working for the image or add an IP address in the `foreman.url` option to check if the DHCP is handling IP addresses to the booted image correctly.
- If the root account is locked on the image and SSH access is disabled but log-in access is still available on the terminal. Provide the `rootpw` option on the command line. Run the following command to generate a salted password:

```
$ openssl passwd salt RH redhat  
RHhwCLrQXB8zE
```

[Report a bug](#)

CHAPTER 14. 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.

[Report a bug](#)

14.1. CREATING A HOST COLLECTION

These steps show how to create Host Collections in **Red Hat Satellite**.

Procedure 14.1. Create Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the **New Host Collection** button.
3. Add the Name and Description of the Host Collection.
4. Uncheck the **Unlimited Content Hosts** button 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 the **Save** button.

Result:

A new host collection is created.

[Report a bug](#)

14.2. ADDING HOSTS TO A HOST COLLECTION

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

Task Prerequisites

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

- A host must be registered to **Red Hat Satellite** in order to add it to a Host Collection. Refer to [Section 12.5.1, “Registering a Host”](#) for information on how to register a host.

Procedure 14.2. Create Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the host should be added.
3. Click the **Content Hosts** subtab.

4. Click the **Add** tab.
5. Check the box next to the desired host and click the **Add Selected** button.

[Report a bug](#)

14.3. ADDING CONTENT TO HOST COLLECTIONS

14.3.1. Adding Packages to a Host Collection

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

Task Prerequisites

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

- 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 14.3. Adding Packages to Host Collections

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

Result:

The selected packages or package groups within the host collection will be installed or updated.

[Report a bug](#)

14.3.2. Adding Errata to a Host Collection

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

Prerequisites

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

- 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 14.4. Adding Errata to Host Collections

1. Click **Hosts** → **Host Collections**.
2. Choose the host collection where the errata should be added.
3. Click the **Collection Actions** subtab.
4. Click the **Errata Installation** link.
5. Choose the errata you wish to push to the host collection and click **Install Selected**.

Result:

The selected errata will be installed in the hosts within the host collection.

[Report a bug](#)

14.4. REMOVING CONTENT FROM A HOST COLLECTION

These steps show how to remove packages from host collections in **Red Hat Satellite**.

Procedure 14.5. Removing Content from Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the package should be removed.
3. Click the **Collection Actions** subtab.
4. Click the **Package Installation, Removal, and Updates** link.
5. In the field provided, type in the package or package group name. Then click **Remove**.

Result:

The package or package group will be removed from all hosts within the host collection.

[Report a bug](#)

14.5. CHANGING THE ASSIGNED LIFE CYCLE ENVIRONMENT OR CONTENT VIEW FOR A HOST COLLECTION

These steps show change the assigned life cycle environment or content view of a host collections in **Red Hat Satellite**.

Procedure 14.6. Changing the Assigned Life Cycle Environment or Content View

1. Click **Hosts** → **Host Collection**.
2. Choose the host collection where the life cycle environment or content view should be changed.
3. Click the **Collection Actions** subtab.
4. Select **Change assigned Life Cycle Environment or Content View**.

5. Select a life cycle environment by checking the check box next to the required life cycle environment.
6. Select the required content view.
7. Click **Assign**.

[Report a bug](#)

14.6. REMOVING A HOST FROM A HOST COLLECTION

These steps show how to remove hosts from host collections in **Red Hat Satellite**.

Procedure 14.7. Remove Hosts from Host Collections

1. Click **Hosts** → **Host Collections**.
2. Choose the desired Host Collection.
3. Click the **Content Hosts** subtab.
4. Check the box next to the host you wish to remove from the host collection.
5. Click the **Remove Selected** button to remove the host from the host collection.

Result:

A host is removed from the host collection.

[Report a bug](#)

14.7. REMOVING A HOST COLLECTION

These steps show how to remove a host collection in **Red Hat Satellite**.

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

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

4. Click the **Remove** button.

Result:

The host collection is removed.

[Report a bug](#)

14.8. CLONING A HOST COLLECTION

These steps show how to clone a host collection in **Red Hat Satellite**.

1. Click **Hosts** → **Host Collections**.
2. On the left hand panel, click the host collection you wish to clone.
3. On the right hand corner of the host collection details, click **Copy Collection**.
4. Add the desired name of the newly cloned host collection.
5. Click the **Create** button.

Result

A cloned copy of the host collection is created.

[Report a bug](#)

14.9. REVIEWING HOST COLLECTION ACTIONS

Prerequisites

Requires an existing host collection.

Procedure 14.8. Reviewing Host Collection Actions

1. Click **Hosts** → **Host Collections**.
2. Click the host collection you wish to view the actions of.
3. Click on the **Details** subtab.

Result

All events history and actions performed on the host collection is displayed.

[Report a bug](#)

CHAPTER 15. RED HAT SATELLITE CAPSULE SERVERS

The Red Satellite Capsule Server is a Satellite component that provides federated services to discover, provision, and configure hosts outside of the primary Satellite server. A Satellite Capsule Server provides the following features:

- Pulp Server/Content Node features, including:
 - Repository synchronization
 - Content delivery
- Red Hat Satellite Provisioning Smart Proxy features, including:
 - DHCP, including ISC DHCP servers
 - DNS, including Bind and MS DNS servers
 - Any UNIX-based TFTP server
 - Puppet Master servers from 0.24
 - Puppet CA to manage certificate signing and cleaning
 - Baseboard Management Controller (BMC) for power management

The Satellite Capsule Server is a means to scale out the Satellite installation. Organizations can create various capsules in different geographical locations where the data centers are located. These are centrally managed through the Satellite Server. When a Satellite user promotes content to the production environment, the Satellite Server will push the content from the Satellite Server to each of the Satellite Capsule Servers. Host systems pull content and configuration from the Satellite Capsule Servers in their location and not from the central Satellite Server.

Creating various Satellite Capsule Servers will decrease the load on the central server, increase redundancy, and reduce bandwidth usage.

[Report a bug](#)

15.1. RED HAT SATELLITE CAPSULE SERVER SCALABILITY

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit but has been tested on a Satellite Server with a Red Hat Enterprise Linux 6.5 and 7 host systems. To date, running 14 capsules with two vCPUs has been successfully tested.

Capsule scalability depends heavily on the following factors, especially when managing puppet clients:

1. Number of CPUs
2. Run-interval distribution
3. Number of puppet classes

The Capsule Server has a concurrency limitations of 100 concurrent puppet agents running at any single point in time. Running more than 100 concurrent puppet agents will result in a 503 HTTP error.

For example, assuming that the puppet agent runs are evenly distributed with less than 100 concurrent puppet agents running at any single point during a run-interval, a Capsule Server with four CPUs can

expect a maximum of 1250-1600 puppet clients with a moderate workload of 10 puppet classes assigned to each puppet client. Depending on the number of puppet clients required, the Satellite installation can scale out the number of Capsule Servers to support them.

Based on the following assumptions:

1. There are no external puppet clients reporting directly to the Satellite 6 integrated capsule.
2. All other puppet clients report directly to an external capsule.

Puppet scalability within Satellite on Red Hat Enterprise Linux 6.5 Capsules are as follows:

1. On the minimum amount of CPUs (two CPUs):
 1. At 1 puppet class per host: Not tested
 2. At 10 puppet classes per host: Maximum of 1020-860
 3. At 20 puppet classes per host: Maximum of 375-330
2. On the recommended amount of CPUs (four CPUs):
 1. At 1 puppet class per host: Maximum of 2250-1875
 2. At 10 puppet classes per host: Maximum of 1600-1250
 3. At 20 puppet classes per host: Maximum of 700-560



NOTE

The maximums in the above given numbers represent an evenly distributed run interval of all puppet agents. Any deviation runs the risk of filling the passenger request queue and is subject to the concurrency limitation of 100 concurrent requests.

[Report a bug](#)

15.2. RED HAT SATELLITE CAPSULE SERVER PREREQUISITES

The Satellite Capsule's requirements are identical to the Satellite Server. These conditions must be met before installing Red Hat Satellite Capsule:

Base Operating System

Red Hat Satellite Capsule is supported on Red Hat Enterprise Linux 6.5 or later, as well as Red Hat Enterprise Linux 7. Install the operating system from disc, local ISO image, kickstart, or any other methods that Red Hat supports. Red Hat Satellite Capsule requires Red Hat Enterprise Linux installations with the @Base package group with no other package-set modifications, and without third-party configurations or software that is not directly necessary for the direct operation of the server. This restriction includes hardening or other non-Red Hat security software. If such software is required in your infrastructure, install and verify a complete working Red Hat Satellite Capsule first, then create a backup of the system before adding any non-Red Hat software.

When installing Red Hat Enterprise Linux from CD or ISO image, there is no need to select any package groups; Red Hat Satellite Capsule only requires the base operating system installation. When installing the operating system via kickstart, select the @Base package group.

- There should be at least one networked host with the following minimum specifications:
 - 64-bit architecture
 - Red Hat Enterprise Linux 6.5 or later
 - A minimum of two CPU cores, but four CPU cores are recommended
 - A minimum of 8 GB of memory but ideally 12 GB of memory for each Satellite instance. It is also recommended to use 4 GB of swap space where possible.
 - A minimum of 5 GB of storage for the base install of Red Hat Enterprise Linux, 300 MB for the installation of Red Hat Satellite Capsule and at least 10 GB storage for each unique software repository to be synchronized in the /var file system.

Packages that are duplicated in different channels are only stored once on the disk. Additional repositories containing duplicate packages will require less additional storage.



NOTE

The bulk of storage resides on the `/var/lib/mongodb` and `/var/lib/pulp` directories. These end points are not manually configurable. Ensure that sufficient storage is available on the `/var` file system to prevent storage issues.

- No Java virtual machine installed on the system, remove any if they exist.
 - No Puppet RPM files installed on the system
 - No third-party unsupported yum repositories enabled. Third-party repositories may offer conflicting or unsupported package versions that may cause installation or configuration errors.
- Administrative user (**root**) access
 - Full forward and reverse DNS resolution using a fully qualified domain name. Check that **hostname** and **localhost** resolve correctly, using the following commands:

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

- Available subscriptions on the Red Hat Satellite Server.



IMPORTANT

Make sure that the host system is fully updated before installing Red Hat Satellite. Attempts to install on host systems running Red Hat Enterprise Linux that are not fully updated may lead to difficulty in troubleshooting, as well as unpredictable results.

Application Specifications

Satellite application installation specifications are as follows:

It is recommended that a time synchronizer such as **ntpd** is installed and enabled on Satellite. Run the following command to start the **ntpd** service and have it persist across restarts:

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

Required Network Ports

The following conditions must be met before continuing with this task:

- Port 443 for HTTPS (secure WWW) must be open for incoming connections.
- Port 5671 must be open for SSL communication with managed systems.
- Port 80 for HTTP (WWW) must be open to download the bootstrap files.
- Port 8080 for TCP must be free for java connections.
- Port 8140 must be open for incoming Puppet connections with the managed systems.
- Port 9090 must be open for Foreman Smart Proxy connections with the managed systems.

Run the following commands to configure the firewall with the **iptables** command and to make these rules persistent during reboots:

1. For Red Hat Enterprise Linux 6:

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5671 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 9090 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8080 -j
ACCEPT \
# iptables-save > /etc/sysconfig/iptables
```

2. For Red Hat Enterprise Linux 7:

```
# firewall-cmd --permanent --add-port="443/tcp" --add-
port="5671/tcp" --add-port="80/tcp" --add-port="8140/tcp" --add-
port="9090/tcp" --add-port="8080/tcp"
# firewall-cmd --complete-reload
```

Red Hat Satellite Server

The Satellite Server must have the Red Hat Software Collections repositories enabled and synchronized prior to the Capsule Server installation.



IMPORTANT

Red Hat recommends that the Satellite Capsule system be a freshly provisioned system that serves no other function except as a Satellite Capsule.

[Report a bug](#)

15.3. INSTALLING A RED HAT SATELLITE CAPSULE SERVER

Prerequisites

The Capsule Server must be registered to the Red Hat Satellite Server to use the Red Hat Satellite Server products and subscriptions:

1. Install the Red Hat Satellite Server's CA certificate in the Capsule Server:

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

2. Register the Capsule Server under your chosen organization's name:

```
# subscription-manager register --org "your organization"
```

This procedure installs a Red Hat Satellite Capsule Server onto a host.

Procedure 15.1. To Install a Satellite Capsule Server on a Certificate-managed System:

1. List all the available subscriptions to find the correct Red Hat Satellite and Red Hat Enterprise Linux product to allocate to your system:

```
# subscription-manager list --available --all
```

The screen displays:

```
+-----+
| Available Subscriptions |
+-----+

ProductName:      Red Hat Satellite
ProductId:        SKU123456
PoolId:           e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Quantity:         10
Multi-Entitlement: No
Expires:          08/20/2013
MachineType:     physical
```



NOTE

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

2. Subscribe to the pool using the following command:

```
# subscription-manager subscribe --pool=Red_Hat_Satellite_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_Id
```

3. Disable all existing repositories:

```
# subscription-manager repos --disable "*"

```

4. Enable the Satellite and Red Hat Enterprise Linux repositories by running **subscription-manager**. You might need to alter the Red Hat Enterprise Linux repository to match the specific version you are using.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc1-6-rpms \
--enable rhel-6-server-satellite-capsule-6.0-rpms
```

5. Install the `katello-installer` and `cyrus-sasl-plain` packages using the **yum install** command as the root user:

```
# yum install katello-installer cyrus-sasl-plain
```

The `katello-installer` provides the **capsule-installer** functionality while `cyrus-sasl-plain` is required for **pulp**.

Result:

The Satellite Capsule Server is installed on your host system. The Satellite Capsule Server must be configured before it can be used.

[Report a bug](#)

15.4. CONFIGURING A RED HAT SATELLITE CAPSULE SERVER

Prerequisite

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

- Install the Red Hat Satellite Server.
- Set the SELinux permissions on the system designated as the Satellite Capsule Server as enforcing.

The following procedures configure a Satellite Capsule Server for use with your Red Hat Satellite Server. This includes the following types of Satellite Capsule Servers:

- Satellite Capsule Server with Smart Proxy
- Satellite Capsule Server as a Content Node
- Satellite Capsule Server as a Content Node with Smart Proxy

To configure a Satellite Capsule Server:

1. On the Satellite Server:

a. Generate a Satellite Capsule Server certificate:

```
capsule-certs-generate --capsule-fqdn capsule_FQDN --certs-tar
~/capsule.example.com-certs.tar
```

Where:

- **capsule_FQDN** is the Satellite Capsule Server's fully qualified domain name. (REQUIRED)
- **certs-tar** is the name of the tar file to be generated that contains the certificate to be used by the Satellite Capsule installer.

Running **capsule-certs-generate** will generate the following output message:

```

    To finish the installation, follow these steps:
    1. Ensure that the capsule-installer is available on the
    system.
       The capsule-installer comes from the katello-installer
    package and
       should be acquired through the means that are appropriate to
    your deployment.
    2. Copy ~/capsule.example.com-certs.tar to the capsule system
    capsule.example.com
    3. Run the following commands on the capsule (possibly with the
    customized
       parameters, see capsule-installer --help and
       documentation for more info on setting up additional
    services):
    rpm -Uvh http://master.com/pub/katello-ca-consumer-
    latest.noarch.rpm
    subscription-manager register --org "Default Organization"
    capsule-installer --parent-fqdn          "sat6.example.com"\
                      --register-in-foreman "true"\
                      --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                      --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                      --pulp-oauth-secret
"doajBEXqNcANY93ZbciFyyswaiwt6BWU"\
                      --certs-tar
"~/capsule.example.com-certs.tar"\
                      --puppet              "true"\
                      --puppetca            "true"\
                      --pulp                "true"
```

- b. Copy the generated tarball, capsule.example.com-certs.tar, from the Satellite Server to the Satellite Capsule host system.

2. On the Satellite Capsule Server:

a. Register your Satellite Capsule Server to the Satellite Server:


```
# rpm -Uvh http://sat6host.example.redhat.com/pub/katello-ca-
consumer-latest.noarch.rpm
# subscription-manager register --org "Default Organization" --
env [environment]/[content_view_name]
```

**NOTE**

The Satellite Capsule Server must be assigned to an organization as the Satellite Capsule Server requires an environment to synchronize content from the Satellite Server. Only organizations have environments.

Assigning a location is optional though recommended to indicate proximity to the hosts that the Satellite Capsule Server is managing.

- b. Depending on the desired Satellite Capsule Server type, choose one of the following options:
- i. Option 1: Satellite Capsule Server with Smart Proxy: This installs a Satellite Capsule Server with Smart Proxy features (DHCP, DNS, Puppet). Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn
"satellite.example.com"\
    --register-in-foreman "true"\
    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
    --foreman-oauth-secret
"w5ZDpyPJ24eSBN053AFybcnqoDYXgLUA"\
    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyyswaiwt6BWU"\
    --certs-tar
"/root/capsule.example.com-certs.tar"\
    --puppet "true"\
    --puppetca "true"\
    --pulp "true"\
    --tftp "true"\
    --dhcp "true"\
    --dhcp-interface "virbr1"\
    --dns "true"\
    --dns-forwarders "8.8.8.8"\
    --dns-forwarders "8.8.4.4"\
    --dns-interface "virbr1"\
    --dns-zone "example.com"
```

- ii. Option 2 - Satellite Capsule Server as a Content Node with Smart Proxy: This installs a Satellite Capsule Server with all the features. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn "sat6.example.com"\
    --register-in-foreman "true"\
    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
    --foreman-oauth-secret
"w5ZDpyPJ24eSBN053AFybcnqoDYXgLUA"\
```

```

--pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
--certs-tar
"/root/capsule.example.com-certs.tar"\
--puppet "true"\
--puppetca "true"\
--pulp "true"\
--tftp "true"\
--dhcp "true"\
--dhcp-interface "virbr1"\
--dns "true"\
--dns-forwarders "8.8.8.8"\
--dns-forwarders "8.8.4.4"\
--dns-interface "virbr1"\
--dns-zone "example.com"

```

- Run the following commands to configure the firewall to limit **elasticsearch** to the **foreman**, **katello** and root users and make these rules persistent during reboots:

- For Red Hat Enterprise Linux 6:

```

iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner --
uid-owner foreman -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner -
uid-owner katello -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner -
uid-owner root -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -j DROP
iptables-save > /etc/sysconfig/iptables

```

- For Red Hat Enterprise Linux 7:

```

firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner foreman -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
foreman -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
katello -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
katello -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP

```



NOTE

If the configuration is successful, run this command as the root user on the Satellite Capsule Server:

```
# echo $?
```

This command should return a "0" to indicate success. If it does not, check `/var/log/kafo` to debug the cause of failure. `/var/log/kafo` is the log file for the output generated by the commands `capsule-certs-generate` and `capsule-installer`.

The Satellite Capsule Server should also appear in the Satellite Server's User Interface under **Infrastructure** → **Capsules**.

Result:

The Satellite Capsule Server is now configured and registered with the Satellite Server.

[Report a bug](#)

15.5. USING POWER MANAGEMENT FEATURES ON MANAGED HOSTS

When you enable the BMC (Baseboard Management Controller) module on the Capsule Server, you can use power management commands on managed hosts using the IPMI (Intelligent Platform Management Interface) or a similar protocol.

The BMC service on the satellite Capsule Server allows you to perform a range of power management tasks. The underlying protocol for this feature is IPMI; also referred to as the BMC function. IPMI uses a special network interface on the managed hardware that is connected to a dedicated processor that runs independently of the host's CPUs. In many instances the BMC functionality is built into chassis-based systems as part of chassis management (a dedicated module in the chassis).

To take advantage of BMC features you need to add a new network interface of type "BMC" to each managed host. IPMI interfaces are nearly always password protected, to prevent unauthorized people on the same network from gaining control of that host. Satellite uses this NIC to pass the appropriate credentials to the host. See [Section 12.4.4, "Adding a BMC Interface"](#).

Red Hat Satellite supports by extension everything that either `ipmitool` or `freeipmi` BMC providers support. You can switch between the two per capsule. Note that different hardware vendors might not implement all IPMI specifications, bugs, and so on.

15.5.1. Enabling BMC Power Management

You can enable the BMC module as part of the installation process, or you can update the required configuration files on an existing system.

Enabling BMC During Installation

To enable BMC power management as part of the installation process, add the following lines to the `capsule-installer` command:

```
--bmc "enabled"\  
--bmc_default_provider "freeipmi"
```

See "Installing a Red Hat Satellite Capsule Server" in the [Red Hat Satellite Installation Guide](#) for full details on installing a Capsule Server.

Enabling BMC After Installation

The following procedure describes how to enable BMC power management on an existing Capsule Server. This requires editing the appropriate configuration file and restarting the required services.

Procedure 15.2. To Enable BMC Power Management on an Existing Capsule:

1. Add the following lines to 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.

15.5.2. Configuring a BMC Interface

This section describes how to configure a BMC interface for a host that supports this feature.

15.5.2.1. Prerequisites

Ensure the following prerequisites are satisfied before proceeding:

- BMC is enabled on the Capsule, as described in [Section 15.5.1, "Enabling BMC Power Management"](#).
- 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.

15.5.2.2. Adding a BMC Interface

This section describes how to add the actual interface configuration to the Capsule server.

Procedure 15.3. To Add a BMC Interface:

1. On the main menu, click **Hosts** → **All Hosts** and then click the name of the host that you want to configure.
2. Click **Edit** to display the host configuration page, and then click the **Network** tab.

3. In the **Interface** section, click **Add Interface**.
4. Select BMC as the interface type, and then complete the other fields. All fields are required.
5. Click **Submit**. The web UI refreshes and you should see a **BMC** tab listed with the **Properties**, **Metrics**, and **Templates** tabs.

15.6. ADDING LIFE CYCLE ENVIRONMENTS TO A RED HAT SATELLITE CAPSULE SERVER

If the newly created Red Hat Satellite Capsule Server has *Content Node* features enabled, the Satellite Capsule Server needs an environment added to the Satellite Capsule Server. Adding an environment to the Red Hat Satellite Capsule Server will allow the Satellite Capsule Server to synchronize content from the Satellite Server and provide content to host systems.



IMPORTANT

The Satellite Capsule Server is configured through the Satellite Server's command line interface (CLI). Execute all **hammer** commands on the Satellite Server.

To add environments to your Satellite Capsule Server:

1. Log in to the Satellite Server CLI as root.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **id**:

```
# hammer capsule list
```

The Satellite Capsule Server's details can be verified using the command:

```
# hammer capsule info --id capsule_id_number
```

3. Verify the list of life cycle environments available for the Red Hat Capsule Server and note down the **environment id**:

```
# hammer capsule content available-lifecycle-environments --id capsule_id_number
```

Where:

- o **available-lifecycle-environments** are life cycle environments that are available to the Satellite Capsule but are currently not attached to the Satellite Capsule.

4. Add the life cycle environment to the Satellite Capsule Server:

```
# hammer capsule content add-lifecycle-environment --id capsule_id_number --lifecycle-environment-id environment_id_number
```

Where:

- o **--id** is the Satellite Capsule Server's identification number.

- `--lifecycle-environment-id` is the life cycle environment's identification number.

Repeat this step for every life cycle environment to be added to the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

```
# hammer capsule content synchronize --id capsule_id_number
```

When a Satellite Capsule Server has various life cycle environments, and only one life cycle environment needs to be synchronized, it is possible to target a specific environment by specifying the environment identification:

```
# hammer capsule content synchronize --id 1 --environment-id 1
```

Result:

The chosen environments now consume packages from repositories on the desired Satellite Capsule Server.

[Report a bug](#)

15.7. REMOVING LIFE CYCLE ENVIRONMENTS FROM THE RED HAT SATELLITE CAPSULE SERVER

There are multiple reasons to remove life cycle environments from the Red Hat Satellite Capsule Server. For example:

- When life cycle environments are no longer relevant to the host systems
- When life cycle environments have been incorrectly added to the Satellite Capsule Server

To remove a life cycle environment from the Satellite Capsule Server:

1. Log in to the Satellite Server CLI as the root user.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **id**:

```
# hammer capsule list
```

The Satellite Capsule Server's details can be verified using the command:

```
# hammer capsule info --id capsule_id_number
```

3. Verify the list of life cycle environments currently attached to the Red Hat Capsule Server and note down the **environment id**:

```
hammer capsule content lifecycle-environments --id capsule_id_number
```

4. Remove the life cycle environment from the Satellite Capsule Server:

```
# hammer capsule content remove-lifecycle-environment --id capsule_id_number --environment-id environment_id
```

Where:

- o `--id` is the Satellite Capsule Server's identification number.
- o `--environment-id` is the life cycle environment's identification number.

Repeat this step for every life cycle environment to be removed from the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

```
# hammer capsule content synchronize --id capsule_id_number
```

Result:

The chosen environments are removed from the Satellite Capsule Server.

[Report a bug](#)

15.8. REGISTERING HOST SYSTEMS TO A RED HAT SATELLITE CAPSULE SERVER

Prerequisite

Client system must be configured for registration. The following chapters in the *Red Hat Satellite User Guide* can assist with configuration:

1. *Configuring Hosts for Registration*
2. *Automated Configuration*
3. *Manual Configuration*
4. *Creating a New Activation Key*

Systems can be registered to the parent host while using a Satellite Capsule as a content source. Register the system to the Satellite 6 server through **subscription-manager** but reference the Satellite Capsule by using the `--baseurl` flag provided by Subscription Manager using `/pulp/repos` as the prefix.

```
# subscription-manager register --org [org_name] --activationkey  
[your_activation_key] --baseurl https://satcapsule.example.com/pulp/repos
```

[Report a bug](#)

15.9. REFRESHING A RED HAT SATELLITE CAPSULE SERVER

Procedure 15.4. Refreshing a Red Hat Satellite Capsule Server

1. Select **Infrastructure** → **Capsules**.
2. Select **Refresh Features** from the drop down menu to the right of the name of the capsule you want to refresh.

Result

The capsule is refreshed with the changes that have been made.

[Report a bug](#)

CHAPTER 16. 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 16.2, “Creating and Managing Users”](#) for more information on working with users.

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 16.3, “Creating and Managing Roles”](#).

16.1. CONFIGURING LDAP AUTHENTICATION FOR RED HAT SATELLITE

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. The following procedure shows how to configuring LDAP authentication.

Procedure 16.1. To Configure LDAP Authentication:

1. Navigate to **Administer** → **LDAP Authentication**.
2. Click **New authentication source**.
3. On the **LDAP Server** tab, enter the LDAP server's name, hostname, port, and server type. The default port is 389. Select the **LDAPS** check box enable encryption.
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:

```
cn=users,cn=accounts,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**.

[Report a bug](#)

16.2. CREATING AND MANAGING USERS

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

16.2.1. Creating a User

The following steps show how to create a user:

Procedure 16.2. 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.

16.2.2. Editing a User

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

Procedure 16.3. 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.

16.2.3. Assigning Roles to Users

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 16.3.1, “Creating a Role”](#). You can apply a similar procedure to user groups.

Procedure 16.4. 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 16.1, “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.

16.2.4. Removing Users

The following procedure describes how to remove an existing user.

Procedure 16.5. 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.

[Report a bug](#)

16.3. CREATING AND MANAGING ROLES

Red Hat Satellite provides a set of predefined roles with permissions sufficient for standard tasks, as listed in [Table 16.1, “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 Foreman plug-ins create roles automatically.

Table 16.1. Predefined Roles Available in Red Hat Satellite

Role	Description ^[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.

Role	Description ^[a]
Discovery reader	View hosts and discovery rules.
Boot disk access	Download the boot disk.
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 Foreman tasks.
Tasks reader	View Foreman tasks.
<p>[a] The exact set of allowed actions associated with predefined roles can be viewed by the privileged user as described in Section 16.3.3, “Viewing Permissions Assigned to a Role” .</p>	

16.3.1. Creating a Role

The following steps show how to create a role.

Procedure 16.6. 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 16.3.2, “Adding Permissions to an Existing Role”](#).

**NOTE**

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. To clone a role, navigate to **Administer** → **Roles** and select **Clone** from the drop-down list to the right of the role to be copied. Select the name for the new role and alter the permissions as needed.

16.3.2. Adding Permissions to an Existing Role

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

Procedure 16.7. 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. 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**. This option is selected by default, which means that the permission is applied on all resources of the selected type. When you disable the **Unlimited** check box, the **Search** field activates. In this field you can specify further filtering with use of the Red Hat Satellite 6 search syntax. See [Section 16.4, “Granular Permission Filtering”](#) for details.
6. Click **Next**.
7. Click **Submit** to save changes.

16.3.3. Viewing Permissions Assigned to a Role

The following steps show how to view permissions assigned to an existing role.

Procedure 16.8. To View Permissions of 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.

16.3.4. Removing a Role

The following steps show how to remove an existing role.

Procedure 16.9. 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.

[Report a bug](#)

16.4. GRANULAR PERMISSION FILTERING

As mentioned in [Section 16.3.2, “Adding Permissions to an Existing Role”](#), Red Hat Satellite provides an 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 16.2, “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 16.2. 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 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 16.1, "Creating an Organization-specific Manager Role"](#).

Example 16.1. 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 can not 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.

[Report a bug](#)

CHAPTER 17. 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:

```
[root@sat.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
```


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

This section describes how to prepare and back up your Red Hat Satellite Server to facilitate recovery in the event of disaster. This example uses the `/backup` directory as the target directory for backup archives, and is described in several parts:

- Preparing the backup location and backing up configuration and data files
- Backing up the repositories
- Backing up the databases
- Verifying the backup



NOTE

If SELinux is enabled and in **enforcing** mode, ensure that any local content requiring synchronization is labeled with "httpd_sys_content_t".

Procedure 18.1. To Prepare the Backup Location and Back up Configuration and Data Files:

1. Prepare your backup location:

```
# umask 0027
# export BDIR=/backup
# mkdir $BDIR
# chgrp postgres $BDIR
# chmod 770 $BDIR
# cd $BDIR
```

2. Back up the configuration and data files:

```
# tar --selinux -czvf config_files.tar.gz \
/etc/katello \
/etc/elasticsearch \
/etc/candlepin \
/etc/pulp \
/etc/pki/katello \
/etc/pki/pulp \
/etc/qpidd.conf \
/etc/sysconfig/katello \
/etc/sysconfig/elasticsearch \
/root/ssl-build \
/var/www/html/pub/*

# tar --selinux -czvf elastic_data.tar.gz /var/lib/elasticsearch
```

Backing up Satellite Repositories

The RPM files in repositories already use compression and cannot be compressed any further. Therefore, depending on instance size, the resulting backup archive (**pulp_data.tar**) can grow quite large. Ensure you have sufficient space to store the resulting file.

There are two options for backing up repositories: online backup; and offline backup.

Procedure 18.2. To Perform an Online Repository Backup:

- Run the following commands to perform a checksum of all time stamps, back up the repository, and perform the checksum again.

```
# find /var/lib/pulp -printf '%T@\n' | md5sum
# tar --selinux -cvf pulp_data.tar /var/lib/pulp /var/www/pub
# find /var/lib/pulp -printf '%T@\n' | md5sum
```

If the checksums match, the online backup is correct and usable. If the checksums do not match, perform the repository backup again.



NOTE

You can use the **rsync** command to speed up file copying. This can help to ensure the checksums match.

Procedure 18.3. To Perform an Offline Repository Backup:

- Run the following commands to stop the required services, perform the backup, and restart the services:

```
# katello-service stop
# tar --selinux -cvf pulp_data.tar /var/lib/pulp /var/www/pub
# katello-service start
```



NOTE

While the **katello-service** is stopped, Red Hat Satellite and the yum clients will be unable to maintain a connection. Any repository actions performed on Red Hat Satellite will fail during this period.

Backing up the Databases

You can perform either online or offline database backups. You do not need to do both. Offline backups require that the Satellite Server be completely inactive.

**WARNING**

Red Hat Satellite must be completely inactive to do an offline backup. Following this procedure while Satellite is running may result in corrupted data.

This method archives all data from the PostgreSQL and MongoDB databases. Red Hat recommends that you perform this backup during maintenance periods.

Procedure 18.4. To Perform a Complete Offline Database Backup:

1. Ensure the Satellite Server is completely stopped:

```
# katello-service stop
```

2. Run the following commands to back up the PostgreSQL and MongoDB databases:

```
# tar --selinux -czvf mongo_data.tar.gz /var/lib/mongodb
# tar --selinux -czvf pgsql_data.tar.gz /var/lib/pgsql/data/
```

3. Restart the required services:

```
# katello-service start
```

Performing Online Database Backups

If you prefer, you can perform separate online backups of your databases. This is not necessary if you have performed complete offline backups.

Procedure 18.5. To Perform an Online Backup of the PostgreSQL Database:

1. Determine the name of the Red Hat Satellite PostgreSQL database. The default name is **foreman** and is specified in the `/usr/share/katello-installer/modules/foreman/manifests/database/postgresql.pp` file. If you chose a different name for your database, it is stored as the value of **db_database** in the `/etc/katello-installer/answers.katello-installer.yaml` file.

**NOTE**

If you used the default database name, this variable has no value in the `answers.katello-installer.yaml` file.

The Candlepin database name, **candlepin**, is not currently user configurable. It is specified in the `/usr/share/katello-installer/modules/candlepin/manifests/init.pp` file.

2. Run the following commands to create online database backups. It is not necessary to stop PostgreSQL or Red Hat Satellite, and this process does not block logged-in users. However, the process can take several minutes to finish depending on database sizes.

```
# runuser - postgres -c "pg_dump -Fc foreman > /backup/foreman.dump"  
# runuser - postgres -c "pg_dump -Fc candlepin >  
/backup/candlepin.dump"
```

Procedure 18.6. To Perform an Online Backup of the MongoDB Database:

- Run the following command in the **/backup** directory to create the **/backup/mongo_dump/pulp_database** directory, including JSON files.

```
# mongodump --host localhost --out $BDIR/mongo_dump
```

Verifying Your Backups

It is important to verify the results of your backups. The process creates the following archive files and directory:

```
# ls $BDIR  
candlepin.dump  
config_files.tar.gz  
elastic_data.tar.gz  
foreman.dump  
mongo_dump/  
pulp_data.tar
```

If you performed the optional offline backup of the databases, the following files also appear:

```
mongo_data.tar.gz  
pgsql_data.tar.gz
```

[Report a bug](#)

18.2. RESTORING RED HAT SATELLITE FROM BACKUP

The following process describes a full Red Hat Satellite restoration. This process deletes all data from an existing Red Hat Satellite instance. During this process, ensure that:

- You restore 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 /
```

**WARNING**

The following procedure deletes all data from an existing Red Hat Satellite instance.

Procedure 18.7. To Restore a Red Hat Satellite Instance:

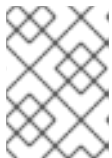
1. Prepare the Red Hat Satellite host for restoration. This process restores the backup to the same server that generated the backup. If the original system is unavailable, provision the same configuration with the same settings (host name, IP address, and so on) and run **katello-installer** with the same options using the file from the configuration backup:

```
# tar --selinux -xzvf config_files.tar.gz -C /tmp
# katello-configure --answer-file=/etc/katello-
  installer/answers.katello-installer.yaml
```

2. Configure and change to the backup directory.

```
# export BDIR=/backup
# chgrp postgres -R $BDIR
# cd $BDIR
```

3. Determine the name of the Red Hat Satellite PostgreSQL database. The default name is **foreman** and is specified in the **/usr/share/katello-installer/modules/foreman/manifests/database/postgresql.pp** file. If you chose a different name for your database, it is stored as the value of **db_database** in the **/etc/katello-installer/answers.katello-installer.yaml** file.

**NOTE**

If you used the default database name, this variable has no value in the **answers.katello-installer.yaml** file.

The Candlepin database name, **candlepin**, is not currently user configurable. It is specified in the **/usr/share/katello-installer/modules/candlepin/manifests/init.pp** file.

4. Stop all services prior to restoring the databases:

```
# katello-service stop
# service postgresql stop
```

5. Restore the system files. Make sure that the files extract on the correct host. Run the following commands in the backup directory to restore all system files:

```
# tar --selinux -xzvf config_files.tar.gz -C /
# tar --selinux -xzvf elastic_data.tar.gz -C /
# tar --selinux -xvf pulp_data.tar -C /
```

- Drop any existing Red Hat Satellite PostgreSQL databases:

```
# service postgresql start
# runuser - postgres -c "dropdb foreman"
# runuser - postgres -c "dropdb candlepin"
```



NOTE

The following error might appear during a database drop:

```
database xxx is being accessed by other users
```

This typically means that some Satellite processes are still running. Ensure all processes are stopped.

- Run the following commands to restore the Red Hat Satellite PostgreSQL databases:

```
# runuser - postgres -c "pg_restore -C -d postgres
/backup/katello.dump"
# runuser - postgres -c "pg_restore -C -d postgres
/backup/candlepin.dump"
```

- Ensure MongoDB is running and delete the old data:

```
# service mongod start
# echo 'db.dropDatabase();' | mongo pulp_database
```

- Run the following command in the **/backup** directory to restore the MongoDB database:

```
# mongorestore --host localhost mongo_dump/pulp_database/
```

- Restart all Red Hat Satellite processes:

```
# service postgresql start
# katello-service start
```

- Inspect the appropriate log files for errors, and inspect the **audit.log** file for AVC denials. Attempt to ping the Red Hat Satellite instance:

```
# hammer -u admin -p admin ping
```

If no errors are apparent, and you can ping the Satellite Server, the restoration should be complete and successful.

Further Reading

- The **mongodump** and **mongorestore** manual pages and <http://www.mongodb.org/display/DOCS/Backups>
- The **pg_dump** and **pg_restore** manual pages and <http://www.postgresql.org/docs/8.4/static/backup.html>

[Report a bug](#)

CHAPTER 19. MAINTENANCE

19.1. LOGGING AND REPORTING

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

Examples of helpful log files for troubleshooting are:

Table 19.1. Relevant Log Files

Log File	Description
<code>/var/log/elasticsearch</code>	Errors concerning the UI search index display
<code>/var/log/candlepin</code>	Errors concerning subscription management
<code>/var/log/foreman</code>	Errors concerning foreman
<code>/var/log/foreman-proxy</code>	Errors concerning the foreman proxy
<code>/var/log/httpd</code>	Errors concerning the apache http server
<code>/var/log/katello-installer</code>	Errors concerning the Satellite installer
<code>/var/log/libvirt</code>	Errors concerning the virtualization API
<code>/var/log/mongodb</code>	Errors concerning the database
<code>/var/log/pulp</code>	Errors in repository management
<code>/var/log/puppet</code>	Errors in configuration management
<code>/var/log/rhsm</code>	Errors in the subscription management tool
<code>/var/log/tomcat6</code>	Issues concerning the apache webserver

Reports can also be generated to view and monitor information about the hosts being maintained. The **foreman-debug** command collects configuration and log data for Red Hat Satellite, its back-end services and system information. This information is collected into a tarball.



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.

[Report a bug](#)

19.1.1. Viewing Import History

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

Procedure 19.1. Viewing Import History

1. Click **Content** → **Red Hat Subscriptions**.
2. Click the **Manage Manifest** button.
3. Click the **Import History** tab.

Result:

Details of the import history are displayed.

[Report a bug](#)

19.2. TROUBLESHOOTING

19.2.1. Changing Your Red Hat Satellite's Fully Qualified Domain Name (FQDN)

Prerequisite

The Satellite FQDN has been changed correctly and the `/etc/sysconfig/network` in Red Hat Enterprise Linux 6 or the `/etc/hostname` file in Red Hat Enterprise Linux 7 has been modified accordingly.

Upon installation, the host system's FQDN is recorded by the Red Hat Satellite Server. Changing the FQDN without the proper precautions can prevent the Satellite Server from running correctly. It also renders all custom server certificates incorrect.

Procedure 19.2. Updating Your Red Hat Satellite Configuration After an FQDN Change

To make sure that Red Hat Satellite continues to run properly even with the FQDN change, follow these steps:

1. Verify that the FQDN is being properly reported and reflects the hostname/FQDN:

```
# facter fqdn
```

2. Update the katello-installer answer file to replace the old FQDN with the new one:

```
# sed -i "s/$OLD_FQDN/$NEW_FQDN/g" /etc/katello-
installer/answers.katello-installer.yaml
```

Where:

1. `$OLD_` is the Satellite Server's previous FQDN.
2. `$NEW_FQDN` is the Satellite Server's new FQDN.
3. Delete the amqp-client certificate from the NSS database:

```
# certutil -D -d '/etc/pki/katello/nssdb' -n 'amqp-client'
```

-
- 4. Regenerate the server certificates by running **katello-installer**:

```
# katello-installer --certs-update-all
```

- 5. On client systems registered to the Red Hat Satellite Server, uninstall the existing `katello-ca-consumer` package since it contains the existing SSL certificate with the old FQDN information and update the package from the new FQDN:

```
# rpm -e $(rpm -qa "katello-ca-consumer*")  
#rpm -Uvh http://NEW_FQDN/pub/katello-ca-consumer-latest.noarch.rpm
```

The Red Hat Satellite is now fully updated with the new FQDN and server certificates.

CHAPTER 20. CONFIGURING IDENTITY MANAGEMENT IN RED HAT SATELLITE

Identity Management (IDM) deals with the management of individual identities, their credentials and privileges used in a networking environment. IDM can help to increase the security of your system and ensure that the right people have access to the right information when they need it.

Red Hat Satellite has a realm feature that will automatically manage the life cycle of any system registered to a realm or domain provider. This section will explain how you need to configure the Satellite Server or Capsule Server for IDM and how to automatically add client systems to the Satellite 6 Identity Management host group.

[Report a bug](#)

20.1. CONFIGURING RED HAT SATELLITE SERVER OR CAPSULE SERVER FOR IDM REALM SUPPORT

The initial step to use Identity Management (IDM) in Red Hat Satellite is to configure the Red Hat Satellite Server or Red Hat Satellite Capsule Server.

Prerequisites

Make sure that the following are setup before configuring IDM:

1. A Satellite Server registered to the content delivery network or an independent Capsule Server registered to the Satellite Server
2. A realm or domain provider such as Red Hat Identity Management configured and set up

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 a FreeIPA or Red Hat Identity Management server for use with the Foreman Smart Proxy. It creates a dedicated role with the permissions needed for Foreman, 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:

- o 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

These options may also be run at the initial configuration of Red Hat Satellite Server.

- o 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:
 1. Realm - provide the realm name, the type of realm to use and the realm proxy.
 2. Locations - choose the locations where the new realm is intended for use.
 3. Organizations - 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.

[Report a bug](#)

20.2. 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.

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:

1. *hostgroup_name* is the hostgroup's name.
 2. *hostgroup_description* is the hostgroup's description.
2. Create an automembership rule:

```
# ipa automember-add --type=hostgroup automember_rule
-----
Added automember rule "automember_rule"
-----
```

Automember Rule: *automember_rule*

Where:

1. ***automember-add*** flags the group as an automember group.
 2. ***--type=hostgroup*** identifies that the target group is a host group, not a user group.
 3. *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:

1. ***automember-add-condition*** allows you to add regular expression conditions to identify group members.
2. ***--key=userclass*** specifies the key attribute as userclass.
3. ***--type=hostgroup*** identifies that the target group is a host group, not a user group.
4. ***--inclusive-regex=*^webserver**** is a regular expression pattern to identify matching values.
5. *hostgroup_name* is the target hostgroup'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.

[Report a bug](#)

CHAPTER 21. RED HAT SATELLITE USER INTERFACE PLUG-INS

21.1. RED HAT ACCESS PLUG-IN

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 help 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.

[Report a bug](#)

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.

To search for solutions from the Red Hat Satellite Server:

1. In the top right, click **Red Hat Access** → **Search**.
2. To log into the Red Hat Customer Portal: In the main panel top right, click **Log In**.



NOTE

To access Red Hat Customer Portal resources, you need to 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.

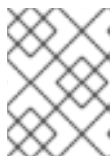
[Report a bug](#)

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. Send the log snippets through the Customer Portal diagnostic tool chain to help with problem solving.

To use the logs diagnostic tool, from the Red Hat Satellite Server:

1. In the top right, click **Red Hat Access** → **Logs**.
2. In the mail panel top right, click **Log In** to log into the Red Hat Customer Portal. If you are already logged in, skip this step.



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 filename.
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 pre-populates with the highlighted text from the log file. See [Section 21.1.5, “Creating New Support Cases Using the Red Hat Access Plug-in”](#).

[Report a bug](#)

21.1.3. Viewing Existing Support Cases Using the Red Hat Access Plug-in

To view existing support cases from the Red Hat Satellite Server:

1. In the top right, click **Red Hat Access** → **Support** → **My Cases**.
2. In the main panel top right, click **Log In** to log into the Red Hat Customer Portal. If you are already logged in, skip this step.



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:
 1. In the **Search** field, provide a key word or phrase.
 2. From the drop-down list, choose a specific **Case Group**. Your organization has defined **Case Groups** inside the Red Hat Customer Portal.

3. 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.

[Report a bug](#)

21.1.4. Modifying Existing Support Cases Using the Red Hat Access Plug-in

Prerequisites

Complete the instructions from the previous section.

Update Support Cases from the Red Hat Satellite Server web interface. When viewing the support case, scroll down to the sections marked to do the following:

- **Attachments:** - Attach a local file from the system. Add a filename to make it easier to identify.



NOTE

Filenames must have less than 80 characters. The maximum file size for web uploaded attachments 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**.

[Report a bug](#)

21.1.5. Creating New Support Cases Using the Red Hat Access Plug-in

1. In the top right, click **Red Hat Access** → **Support** → **New Case**.
2. In the main panel top right, click **Log In** to log into the Red Hat Customer Portal. If you are already logged in, skip this step.



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.



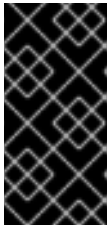
NOTE

Based on the summary, recommendations for possible solutions display in the main panel.

4. Click **Next**. A second screen displays.
5. Choose the appropriate options, as follows:
 - o **Severity**: Select the ticket urgency as 4 (low), 3 (normal), 2 (high) or 1 (urgent).
 - o **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

Filenames must have less than 80 characters. The maximum file size for web uploaded attachments 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.

[Report a bug](#)

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, EC2, and VMWare.

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 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) ^[3] organization or department, this is the Definitive Media Library ^[4] (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](#) [5]
- [The Structure of Satellite Certificates \(Classic Style of Certificates\)](#) [6]

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.

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.

[Report a bug](#)

[3] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[4] http://en.wikipedia.org/wiki/Definitive_Media_Library

[5] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/MigratingRHN/sat-certs.html

[6] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/Subscription_Concepts_and_Workflows/index.html#subscr-legacy

APPENDIX B. REVISION HISTORY

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 Server.		
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 BZ#1109747 - Information added regarding organizations and subscription manifests.	Fri 11 July 2014	Megan Lewis
Revision 0-18 BZ# 1117861 - Section 10.3.1 Corrected CA Certificate URL. BZ#1104914 - Section 5 Partial peer review implementation.	Thu 10 July 2014	Athene Chan
Revision 0-17 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.	Wed 9 July 2014	Megan Lewis
Revision 0-16 Preparing book for Beta release.	Wed 25 Jun 2014	Athene Chan
Revision 0-15 Fixing minor issues.	Mon 11 Nov 2013	Dan Macpherson
Revision 0-14 Preparation for MDP2.	Mon 11 Nov 2013	Dan Macpherson
Revision 0-13 Adding table for synchronization content directories.	Wed 09 Oct 2013	Dan Macpherson
Revision 0-12 Finalizing QE review implementation	Wed 09 Oct 2013	Dan Macpherson
Revision 0-11 BZ#887680 Minor typo corrections.	Tue 1 Oct 2013	Athene Chan
Revision 0-10 Rebuild from typo verification.	Mon 30 Sep 2013	Dan Macpherson
Revision 0-09 Minor tagging errors corrected.	Wed 18 Sep 2013	Athene Chan
Revision 0-08 BZ#956256, 969922, 864115 Implemented suggested changes to information on the User Guide.	Tue 17 Sep 2013	Athene Chan
Revision 0-07 Book product changed.	Fri 13 Sep 2013	Athene Chan
Revision 0-06 Minor grammatical edits. Added book component to the ent file.	Thu 12 Sep 2013	Athene Chan
Revision 0-05 BZ#1004566, 1004567, 1004568, 1004570, 1004571, 1004581, 1004586, 1004588, 1004590, 1004595, 1004597, 1004598, 1004600 Quality assurance edits implemented throughout the book.	Thu 12 Sep 2013	Athene Chan
Revision 0-04 Removing draft watermark.	Mon 12 Aug 2013	Dan Macpherson
Revision 0-03 Creating build for technical review.	Mon 12 Aug 2013	Dan Macpherson
Revision 0-02	Tue 28 May 2013	Athene Chan

Initial book creation