



Red Hat Satellite 6.1 Provisioning Guide

A guide to provisioning physical and virtual hosts from Red Hat Satellite servers.
Edition 1

Red Hat Satellite Documentation
Team

Red Hat Satellite 6.1 Provisioning Guide

A guide to provisioning physical and virtual hosts from Red Hat Satellite servers.

Edition 1

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 Provisioning Guide is a task-based document designed to help you install and configure Red Hat Satellite, ready to provision physical and virtual hosts. This includes setting up the required network topology, configuring the necessary services, and providing all of the other configuration information needed to provision hosts on your network. This guide is aimed primarily at Satellite administrators with sound networking knowledge and skills.

Table of Contents

Chapter 1. Introduction to Provisioning Using Red Hat Satellite	2
1.1. Creating a Red Hat Enterprise Linux Host	2
1.2. Setting up the Network Topology	2
1.3. Registering and Configuring the Host	2
Chapter 2. Configuring Red Hat Satellite Services	5
2.1. Configuring DNS, DHCP, and TFTP	5
2.2. Associating Objects with the Default Organization and Location	7
Chapter 3. Importing Subscriptions and Synchronizing Content	10
3.1. Creating a Manifest	10
3.2. Uploading a Manifest to your Satellite Server	10
3.3. Enabling Red Hat Repositories	10
3.4. Creating Custom Products and Repositories	11
3.5. Synchronizing Content	12
Chapter 4. Content Management and Promotion	13
4.1. Creating Application Life Cycle Environments	13
4.2. Creating Content Views	13
4.3. Adding Red Hat Enterprise Linux Repositories	13
4.4. Adding Puppet Modules	14
4.5. Publishing a Content View	14
4.6. Creating and Editing Activation Keys	15
Chapter 5. Finalizing Provisioning Configuration	16
5.1. Creating Provisioning Templates	16
5.2. Creating Host Groups	16
Chapter 6. Provisioning Hosts	18
6.1. Provisioning a Host Using PXE	18
6.2. Provisioning a Host Using a Boot Disk	18
6.3. Provisioning Hosts with Static IP Addresses	22
Appendix A. Glossary of Terms	24
Appendix B. Revision History	28

Chapter 1. Introduction to Provisioning Using Red Hat Satellite

This guide is designed to help you configure a Red Hat Satellite server to provision hosts. This includes installing Red Hat Enterprise Linux, describes a typical network topology and the expected services available, and registering the host to Red Hat Subscription Management.

1.1. Creating a Red Hat Enterprise Linux Host

Install Red Hat Enterprise Linux Server, version 6.6 or later on x86_64 using either the @Core or @Base package set. For more information about installing Red Hat Enterprise Linux, see [Red Hat Enterprise Linux 6 Installation Guide](#).

1.2. Setting up the Network Topology

This guide assumes that the host running Satellite 6 is deployed on a dedicated subnet where it can enable DHCP, DNS, and TFTP services. These examples use the 172.17.13.0/24 subnet. In addition, it uses the example.org DNS domain, which is managed by the Satellite.

These examples also assume the following network details for the Satellite host. Adjust these parameters to suit your deployment:

- » Hostname: satellite.example.org
- » IP address = 172.17.13.2
- » Netmask: 255.255.255.0

1.3. Registering and Configuring the Host

The following sections describe how to register your host, identify subscriptions, and attach those subscriptions so that the host can consume content.

1.3.1. Registering to Red Hat Subscription Management

The first step in this process is to register the host to Red Hat Subscription Management. This enables the host to subscribe to and consume content for any subscriptions available to the user. This includes content such as Red Hat Enterprise Linux, Red Hat Software Collections (RHSC), and Red Hat Satellite. Use the `subscription_manager register` command to register your Satellite:

```
# subscription-manager register
Username: demouser
Password:
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-
7683431bcf9a
```

1.3.2. Identifying the Satellite Subscription

After you have registered your host, you need to identify your Satellite subscription Pool ID. You need this ID so that you can attach the required subscription to your host. The Satellite

subscription provides access to the Satellite content, as well as Red Hat Enterprise Linux, Red Hat Software Collections (RHSC), and Red Hat Satellite. This is the only subscription required.

Run the following command to identify your Satellite subscription:

```
# subscription-manager list --all --available
Subscription Name: Red Hat Satellite
Provides:          Oracle Java (for RHEL Server)
                  Red Hat Satellite 6 Beta
                  Red Hat Enterprise Linux Server
                  Red Hat Satellite
                  Red Hat Enterprise Linux Load Balancer (for RHEL Server)
SKU:              MCT0370
Pool ID:          8a85f9874152663c0541943739717d11
Available:        3
Suggested:        1
Service Level:    Premium
Service Type:     L1-L3
Multi-Entitlement: No
Ends:             10/07/2014
System Type:      Physical
```



Important

Make a note of the Pool ID; you need this value to attach your subscription to your Satellite host. In this example the Pool ID is 8a85f9874152663c0541943739717d11. The Pool ID for your subscription will be different.

Run the following command to attach your subscription to your Satellite. Ensure you substitute your own Pool ID:

```
# subscription-manager attach --pool=8a85f9874152663c0541943739717d11
Successfully attached a subscription for: Red Hat Satellite
```

Run the following command to ensure that the subscriptions were successfully attached:

```
# subscription-manager list --consumed
+-----+
Consumed Subscriptions
+-----+
Subscription Name: Red Hat Satellite
Provides:          Red Hat Satellite
                  Red Hat Enterprise Linux Server
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Satellite Beta
                  Red Hat Satellite 6 Beta
                  Red Hat Software Collections Beta (for RHEL Server)
                  Red Hat Satellite Capsule Beta
                  Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                  Red Hat Satellite with Embedded Oracle
                  Red Hat Satellite Capsule
                  Red Hat Enterprise Linux High Availability (for RHEL
```

```
Server)
SKU:           MCT0370
Contract:      10293569
Account:       5361051
Serial:        1653856191250699363
Pool ID:       8a85f9874152663c0541943739717d11
Active:        True
Quantity Used: 1
Service Level: Premium
Service Type:  L1-L3
Status Details:
Starts:        10/08/2013
Ends:          10/07/2014
System Type:   Physical
```

1.3.3. Configuring Repositories

You need to disable all existing repositories and only enable the Red Hat Enterprise Linux 6, Red Hat Software Collections, and Satellite 6 repositories. Run the following commands:

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhsc-6-rpms --enable rhel-6-server-satellite-6.0-rpms
```

1.3.4. Installing Satellite 6

Run the following command to install Satellite 6:

```
# yum install katello
```

Chapter 2. Configuring Red Hat Satellite Services

In this example configuration, the Satellite is responsible for provisioning hosts in the 172.17.13.0/24 subnet. This section describes how to configure DNS, DHCP, and TFTP to service the clients that are being provisioned on the subnet.

2.1. Configuring DNS, DHCP, and TFTP

This section describes how to configure Satellite to run BIND (**named**) to provide authoritative DNS services for the **example.org** domain and the 172.17.13.x subnet. This requires setting up a DNS zone for forward lookups, which will be contained in the **example.org** zone file. Additionally, a DNS zone for reverse lookups will be created for the 172.17.13.x subnet, which will be contained in the **13.17.172.in-addr.arpa** reverse zone file. This ensures that hosts provisioned from Satellite use the correct name resolution parameters. This section also describes how to configure the TFTP proxy so that hosts can boot using PXE.

Clients on this network will have the following characteristics:

- ✦ Have access to IP addresses in the range 172.17.13.100 to 172.17.13.150 for DHCP.
- ✦ Use the Satellite (**satellite.example.org** at 172.17.13.2) for DNS.
- ✦ Receive a **pxelinux.0** file from Satellite (**satellite.example.org** at 172.17.13.2) to enable PXE-booting.
- ✦ Have host names of **hostname.example.org**, where *hostname* is configured when the host is provisioned.



Important

This example enables DHCP services on the Satellite server. Consult your network administrator before proceeding.

2.1.1. Satellite Configuration Options

The following table describes the various options and the values required to correctly configure the Satellite server. The **katello-installer** command uses Puppet; consequently, it will install additional packages (**bind**, **dhcp**, **xinetd**, and so on) and configure them to add the requested functionality.

For a complete list of available options, run **katello-installer --help**.

Table 2.1. Satellite Configuration Options

Option	Description	Value
<code>--foreman-admin-username</code>	The user name for the initial administrator.	(User specified)
<code>--foreman-admin-password</code>	The password for the initial administrator.	(User specified)
<code>--capsule-dns</code>	Enable DNS proxy capability.	yes
<code>--capsule-dns-interface</code>	Which interface named should listen on.	eth0

Option	Description	Value
--capsule-dns-zone	The Forward DNS zone that the Satellite will host.	example.org
--capsule-dns-forwarders	The DNS server that unknown queries are forwarded to.	172.17.13.1
--capsule-dns-reverse	The Reverse DNS zone the Satellite hosts. This is usually the first three octets of the IP address (172.17.13) reversed, and appended with ".in-addr.arpa".	13.17.172.in-addr.arpa
--capsule-dhcp	Enable DHCP proxy capability.	yes
--capsule-dhcp-interface	The interface that DHCP listens on.	eth0
--capsule-dhcp-range	The range of IP addresses to issue to clients.	172.17.13.100 172.172.13.150
--capsule-dhcp-gateway	The default gateway IP to issue to clients.	172.17.13.1
--capsule-dhcp-nameservers	The host that the clients should use for name resolution. This should be configured with the Satellite's IP in this deployment model.	172.17.13.2
--capsule-tftp	Enable TFTP proxy capability. This is needed to PXE boot the clients.	yes
--capsule-tftp-servername	Set the TFTP host name. Set this to match the server's host name (satellite.example.org).	\$(hostname)
--capsule-puppet	Enable the Puppet Master.	yes
--capsule-puppetca	Enable the Puppet CA.	yes

2.1.2. Configuring Satellite Services

Run the following **katello-installer** command as root, using the specified options to configure the required services on the Satellite server. Remember to substitute your desired administrator user name and password.



Important

- ✦ If you have already installed Satellite using the instructions in the *Installation Guide*, do not include the **--foreman-admin-username** and **--foreman-admin-password** options in the following command.
- ✦ If you do not specify the administrator user name and password, the default user **admin** is created, and the password is automatically generated. The credentials are displayed at the end of the installation process. Make a note of this password. You can also retrieve the password from **admin_password** parameter in the **/etc/katello-installer/answers.katello-installer.yaml** file.

```
katello-installer \
--foreman-admin-username admin-username \
--foreman-admin-password admin-password \
--capsule-dns true \
--capsule-dns-interface eth0 \
```

```
--capsule-dns-zone example.org \
--capsule-dns-forwarders 172.17.13.1 \
--capsule-dns-reverse 13.17.172.in-addr.arpa \
--capsule-dhcp true \
--capsule-dhcp-interface eth0 \
--capsule-dhcp-range "172.17.13.100 172.17.13.150" \
--capsule-dhcp-gateway 172.17.13.1 \
--capsule-dhcp-nameservers 172.17.13.2 \
--capsule-tftp true \
--capsule-tftp-servername $(hostname) \
--capsule-puppet true \
--capsule-puppetca true
```

At the end of the installation process, **katello-installer** displays the status of the installation.

```
Success!
* Katello is running at https://satellite.example.org
  Default credentials are 'admin:*****'
* Capsule is running at https://satellite.example.org:9090
* To install additional capsule on separate machine continue by running:"

  capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
  "~/${CAPSULE}-certs.tar"

The full log is at /var/log/katello-installer/katello-installer.log
```

Use a web browser to navigate to <https://satellite.example.org> to display the Satellite home page. This example uses the default organization (Default_Organization) and the default location.

2.2. Associating Objects with the Default Organization and Location

Because Satellite 6 supports multiple organizations (logical management divisions) and locations (physical divisions of content delivery), you need to associate your templates, subnets, and other items needed for provisioning with the default organization (Default_Organization) and the default location (Default_Location).

Use the following procedures to make all the "pre-seeded" content available to the default organization (Default_Organization):

Procedure 2.1. To Specify the Default Location:

1. On the main menu, click **Administer** → **Locations** and then click **Default_Location** in the **Name** column.
2. Click **Organizations** to display the list of organizations.
3. Click **Default_Organization** to add it to the **Selected items** list, and then click **Submit**.

Procedure 2.2. To Specify the Default Organization:

1. On the main menu, click **Administer** → **Organizations** and then click **Default_Organization** in the **Name** column.
2. Click **Locations** to display the list of available locations.
3. Click **Default_Location** to add it to the **Selected items** list, and then click **Submit**.

Procedure 2.3. To Associate the Domain with the Default Organization:

1. On the main menu, click **Infrastructure** → **Domains** to open the **Domains** screen.
2. Click **example.org** in the **Description** column. This opens the **Edit Domain** screen where you can update the details of the domain.
3. On the **Domain** tab, change the DNS domain to reflect the host name of the Satellite.
4. Set the **DNS Capsule** value to the Satellite server.
5. On the **Locations** tab, click **Default_Location** to add it to the **Selected items** list. This associates the domain with the default location.
6. On the **Organizations** tab, click **Default_Organization** to add it to the **Selected items** list. This associates the domain with the default organization.
7. Click **Submit** to apply your changes.

Procedure 2.4. To Select the Default Subnet:

1. On the main menu, click **Infrastructure** → **Subnets**.
2. Click **New Subnet** and then complete the following information. Remember to update the details to suit your own deployment:
 - ✦ Name: Provisioning_Net
 - ✦ Network address: 172.17.13.0
 - ✦ Network mask: 255.255.255.0
 - ✦ Gateway Address: 172.17.13.1
 - ✦ Primary DNS Server: 172.17.13.2
 - ✦ Secondary DNS Server: Leave blank
 - ✦ Start of IP Range: 172.17.13.100
 - ✦ End of IP Range: 172.17.13.150
 - ✦ VLAN ID: Leave blank
3. Click **Submit**.
4. Click **Provisioning_Net** to edit the subnet.
5. On the **Domains** tab, select **example.org**
6. On the **Capsules** tab, change the DNS, DHCP, and TFTP capsules to reflect the host name of the Satellite.

7. On the **Locations** tab, select Default_Location under **All items** to associate the domain with the default location.
8. On the **Organizations** tab, select Default_Organization under **All items** to associate the domain with the default organization.

Procedure 2.5. To Associate Installation Media with Organizations and Locations:

1. On the main menu, click **Hosts** → **Installation Media**
2. In the **Name** column, click the name of the media that you want to use.
3. On the **Locations** tab, add the required location to the list of selected items.
4. On the **Organizations** tab, add the required organization to the list of selected items, and then click **Submit**.

Chapter 3. Importing Subscriptions and Synchronizing Content

This section describes how to set up Satellite to download and manage content. This includes uploading a manifest file to the Satellite server, enabling Red Hat repositories, creating custom products, and synchronizing content.

3.1. Creating a Manifest

This sections describes how to create a suitable manifest for your Red Hat Satellite.

Procedure 3.1. To Create a Manifest for Satellite 6:

1. Navigate to access.redhat.com and click **SUBSCRIPTIONS** on the main menu.
2. In the **Red Hat Subscription Management** section, under **Subscription Management Applications**, click **Satellite**.
3. Locate the system for which you need to create a manifest. Ensure you select the correct version.
4. For each subscription that you want to attach, select the check box for that subscription, and specify the quantity of subscriptions to attach.
5. Click **Attach Selected**.



Note

It can take several minutes for all the subscriptions to attach. Refresh the screen every few minutes until you receive confirmation that the subscriptions are attached.

6. After the subscriptions have been attached, click **Download Manifest** and save the manifest file to a known location.

3.2. Uploading a Manifest to your Satellite Server

Procedure 3.2. To Upload a Manifest to your Satellite Server:

1. If you have not already selected the correct Organization, click **Any Context** → **Any Organization** → **Default_Organization**.
2. Click **Content** → **Red Hat Subscriptions**.
3. Click **Manage Manifest** to open the **Subscriptions** page.
4. Click **Browse** to select a suitable manifest, and then click **Open**.
5. Click **Upload** to upload the manifest to the Satellite server.

3.3. Enabling Red Hat Repositories

This section describes how to enable the required Red Hat repositories in order to support provisioning a Red Hat Enterprise Linux 6 host. Select the required release to suit your deployment. This procedure enables the following repositories:

- Red Hat Enterprise Linux 6 Server (Kickstart): This allows Satellite to kickstart a host. This is roughly analogous to a Satellite 5 "kickstart tree."
- Red Hat Enterprise Linux 6 Server (RPMs): This provides ongoing content and errata. This is roughly analogous to the Satellite 5 "rhel-x86_64-server-6" channel.
- Red Hat Enterprise Linux 6 Server - Satellite Tools (RPMs): This provides supporting software, such as Puppet and katello-agent. This is roughly analogous to the Satellite 5 "rhn-tools-rhel-x86_64-server-6" channel.

Procedure 3.3. To Enable Red Hat Repositories:

1. Select **Content** → **Red Hat Repositories**.
2. On the **Kickstarts** tab, navigate to **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server (Kickstart)** and select the Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.5 (or later) repository.



Important

Do NOT select the 6Server Kickstart repository; future versions of Satellite hide these unnecessary Kickstart repositories.

3. On the **RPMs** tab, navigate to **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server (RPMs)** and select the Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server repository.
4. On the **RPMs** tab, navigate to **Red Hat Enterprise Linux Server** → **Red Hat Enterprise Linux 6 Server - Satellite Tools (RPMs)** and select the Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64 repository.

3.4. Creating Custom Products and Repositories

This section describes how to create a custom product, reflecting the Puppet modules to deploy. You can also use this procedure to create custom repositories for both Puppet and Yum.

Procedure 3.4. To Create a Custom Product:

1. Click **Content** → **Products** and then click **New Product**.
2. Enter **Custom Products** in the **Name** field. The label is automatically generated. You do not need to enter a GPG key, synchronization plan or description.
3. Click **Save**.
4. After the screen refreshes, click **Create Repository**.
5. Enter **Puppet Modules** for the name. The label is automatically generated.
6. In the **Type** field, select **puppet**. Leave the **URL** field blank.
7. Click **Save**.

The next step is to upload a Puppet Module to the Puppet Module repository. You can also use the <https://forge.puppetlabs.com> as the URL to mirror Puppet Forge locally. This means that all of the content from Puppet Forge will be available on your Satellite. However, this requires downloading over 2700 modules and can take considerable time, depending on available bandwidth. This example uses the motd module because it is simple, and has no dependencies on other modules.

Procedure 3.5. To Upload a Puppet Module to the Repository:

1. Download the motd puppet module from <https://forge.puppetlabs.com/jeffmccune/motd>. The file that you download will have a **.tar.gz** extension.
2. Click **Content** → **Products** and then click **Custom Products** in the **Name** field.
3. On the **Repositories** tab, click **Puppet Modules** to modify the Puppet Modules repository.
4. In the **Upload Puppet Module** section, click **Browse**, and navigate to the motd module that you downloaded.
5. Click **Upload**.

3.5. Synchronizing Content

This section describes how to synchronize repositories from the Red Hat Content Delivery Network to your Satellite. This procedure also applies to synchronizing custom repositories (that is, Yum or Puppet) that contain a repository URL.

Procedure 3.6. To Synchronize Repositories to Your Satellite:

1. Click **Content** → **Sync Status** to display the list of available products.
2. Navigate to **Red Hat Enterprise Linux Server** → **6Server** → **x86_64**.
3. Select the following products:
 - ✦ Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server.
 - ✦ Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64.
4. Navigate to **Red Hat Enterprise Linux Server** → **6.6** → **x86_64** and select Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.6.
5. Click **Synchronize Now**.

Wait for the repositories to synchronize; this could take several hours, depending on available bandwidth.

Chapter 4. Content Management and Promotion

This chapter describes how to set up Application Life Cycle Environments and Content Views, as well as how to add Red Hat Enterprise Linux repositories and Puppet modules. It also describes how to publish Content Views and how to create and edit Activation Keys.

The procedures in this chapter assume that you have successfully completed the steps in [Chapter 3, *Importing Subscriptions and Synchronizing Content*](#)

4.1. Creating Application Life Cycle Environments

An Application Life Cycle Environment represents a step, or stage, in a promotion path through the Software Development Life Cycle (SDLC). The first part of this example configures two Life Cycle Environments: Dev; and QA. The second part of the example creates a Content View for use with those environments.

Procedure 4.1. To Create an Application Life Cycle Environment:

1. Click **Content** → **Life Cycle Environments** to open the **Life Cycle Environment Paths** screen.
2. Click **Add New Environment** to display the **New Environment** page. The **Library** is the origin of all content that you can use in your environments.
3. Type **Dev** in the **Name** field; the label is automatically populated with the same name, but you can change it to suit your needs. You can add a description of your environment if desired.
4. Click **Save** to save the new environment and return to the previous page.
5. Click **Add New Environment** again and this time create an environment called **QA**.
6. Click **Save**.

4.2. Creating Content Views

A Content View is a managed selection of content, which contains one or more repositories (either yum or Puppet) with optional filtering. These filters can be either inclusive or exclusive, and tailor a host view of content for life cycle management. They are used to customize content to be made available to client hosts.

Procedure 4.2. To Create a Content View:

1. Click **Content** → **Content Views** and then click **Create New View**.
2. Type RHEL6 x86_64 in the **Name** field; the label is automatically populated.
3. Ensure the **Composite View** check box is cleared, and then click **Save**.

After you have successfully created the Content View, the **Repository Selection** screen displays automatically. Use this screen to add selected repositories and Puppet Modules to the Content View.

4.3. Adding Red Hat Enterprise Linux Repositories

The following procedure describes how to add Enterprise Linux repositories to the content view created in the previous step. You can use the same procedure to add any Red Hat or custom repository.

This example shows a simple use case where all content is published. You can also create filters to control the content that is included in or excluded from the published Content View.

Procedure 4.3. To Add the Red Hat Enterprise Linux RPM Repositories:

1. On the **Content Selection** screen, on the **Add** tab, select the check box next to each of the following repositories:
 - ✦ Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server
 - ✦ Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
 - ✦ Red Hat Enterprise Linux 6 Server - Satellite Tools RPMs x86_64
2. Click **Add Repositories**. After the page has refreshed, you can see the list of repositories in your Content View on the **List/Remove** tab.

4.4. Adding Puppet Modules

The following procedure describes how to add the motd Puppet Module that was uploaded in [Procedure 3.5, "To Upload a Puppet Module to the Repository:"](#) Puppet Modules are self-contained bundles of code and data that you can use to manage resources such as users, files, and services.

Procedure 4.4. To Add a Puppet Module:

1. Ensure you are still on the **Content Views** page; on the main menu, click **Content** → **Content Views**.
2. On the **Puppet Modules** tab, click **Add New Module** to display a list of available Puppet Modules. You can use the **Filter** field to help locate the required module.
3. Click **Select Version** to select the motd module.
4. Click **Select Version** next to the version of the module that you want to add.



Note

If you select "Use Latest" when you select which Puppet module version to use, it means that whenever a new version of the Content View is published, the latest version of that module is included in the published view.

4.5. Publishing a Content View

If you have successfully completed all of the preceding steps, your Red Hat Satellite now has one content view, which contains three Red Hat Enterprise Linux repositories, and one Puppet Module. The next step is to publish it to the Library.

Procedure 4.5. To Publish a Content View to the Library:

1. Click **Content** → **Content Views** to display the **Content Views** page.
2. Click the name of the Content View that you want to publish.
3. Click **Publish New Version** to display the **Publish New Version** page. This specifies the version and allows you to enter a comment to reflect any changes made to the content view.
4. Click **Save** to publish the Content View to the Library. You can monitor the publication progress on the screen that appears.
5. When the publishing process is complete, click **Promote** to display the list of available Promotion Paths (Library -> Dev -> QA).
6. Select the check box for the Dev environment, and then click **Promote Version**.

4.6. Creating and Editing Activation Keys

After you have successfully published a content view, you need to create an Activation Key. In a later procedure, the Activation Key is associated with a Host Group. This allows the provisioned host to be registered to the Satellite and associated with the selected Life Cycle Environment, Content View, Subscriptions, and so on.

Procedure 4.6. To Create an Activation Key:

1. On the main menu, click **Content** → **Activation Keys** and then click **New Activation Key**.
2. In the Name field, type **ak-Reg_to_Dev**.
3. For the purposes of this example, clear the **Content Host Limit** check box.

You can use this field to control how many times a given Activation Key is used. For example, if you associate the key with a subscription that has a limited quantity, you can set the limit on the Activation Key to eliminate exceeding that quantity.
4. Select the check box for the Dev environment.
5. In the **Content View** drop-down list, select the RHEL 6 x86_64 Content View, and then click **Save**.

After you have created the Activation Key, you can edit various parameters for that key.

Procedure 4.7. To Edit Activation Key Parameters:

1. On the **Activation Keys** page, click **Subscriptions** → **Add** to display the list of available subscriptions.
2. Select the check box next to each subscription that you want to attach to *each* host that uses this activation key.
3. Click **Add Selected**.

Chapter 5. Finalizing Provisioning Configuration

After you have successfully created a Content View and Activation Key, you need to set up the remaining items necessary to provision a host. This includes configuring provisioning templates and creating host groups.

5.1. Creating Provisioning Templates

This section describes how to set up a provisioning template that you can use to provision multiple hosts, each having the same configuration.

Procedure 5.1. To Create a Provisioning Template:

1. On the main menu, click **Hosts** → **Provisioning Templates**.
2. In the **Name** column, click **Satellite Kickstart Default** in the list of provisioning templates. This displays the configuration tabs where you can customize the template.
3. On the **Association** tab, select RHEL Server 6.5 from the list of applicable operating systems, and then click **Submit**.
4. In the **Name** column, click **Kickstart default PXELinux** in the list of provisioning templates.
5. On the **Association** tab, select RHEL Server 6.5 from the list of applicable operating systems, and then click **Submit**.
6. On the main menu, click **Hosts** → **Operating Systems** and then click RHEL Server 6.5. This displays the configuration tabs where you can customize the operating system.
7. On the **Partition Table** tab, select **Kickstart Default**.
8. On the Installation Media tab, ensure `Default_Organization/Library/Red_Hat_6_Server_Kickstart_x86_64_6Server` is visible and selected.
9. On the Templates tab, select **Kickstart default PXELinux** from the **PXELinux** drop-down list.
10. Select **Satellite Kickstart Default** from the **Provision** drop-down list, and then click **Submit**.

5.2. Creating Host Groups

This section describes how to create and configure a Host Group. A Host Group is effectively a host template that you can reuse to provision multiple hosts without the need to specify the same properties for each host.

Procedure 5.2. To Create a Host Group:

1. On the main menu, click **Configure** → **Host Groups**, and then click **New Host Group**.
2. On the **Host Group** tab, complete the following values:
 - » **Name:** RHEL6Server-x86_64

✧ **Lifecycle Environment:** Default_Organization/DEV

✧ **Content View:** RHEL_6_x86_64



Note

This field only appears after you enter a value in the **Life Cycle Environment** field.

✧ **Content Source:** The FQDN of your Capsule (which may be the Satellite Server).

✧ **Puppet CA:** The FQDN of your Satellite.

✧ **Puppet Master:** The FQDN of your Satellite.

3. On the **Puppet Classes** tab, select the `motd puppet` module from the list of available classes.

4. On the **Network** tab, select the following values:

✧ **Domain:** example.org

✧ **Subnet:** Provisioning_Net

✧ **Realm:** For the purposes of this example, leave this field blank. If you have configured realm management, for example IPA, select the appropriate realm here.

5. On the **Operating System** tab, select the following values:

✧ **Architecture:** x86_64

✧ **Operating system:** RHEL Server 6.5

✧ **Media:** Default_Organization/Library Red Hat Server 6.5 x86_64. In this example, this is automatically populated.



Note

If this field is not automatically populated, it means the Organization and Location associations are not correctly configured. See [Procedure 2.5, “To Associate Installation Media with Organizations and Locations:”](#).

✧ **Partition table:** Kickstart default

✧ **Root password:** *changeme*

6. On the **Locations** tab, select Default_location.

7. On the **Organizations** tab, select Default_Organization.

8. On the **Activation Keys** tab, select ak-Reg_To_Dev from the **Activation Keys** list.

9. Click **Submit**.

Chapter 6. Provisioning Hosts

This chapter describes how to provision a new host using the Red Hat Satellite Server. The preceding chapters worked through installing and configuring everything that is required for provisioning; ensure that you have successfully completed all of the tasks in those chapters before you attempt to provision hosts.

Satellite provides two main approaches to provisioning hosts: PXE booting, which requires DHCP and TFTP services; and boot disk provisioning, which provides host provisioning when PXE services are not available.

6.1. Provisioning a Host Using PXE

The following procedure describes how to provision a host from your Satellite 6 Server.

Procedure 6.1. To Provision a Host:

1. On the main menu, click **Hosts** → **New Host** to open the **New Host** page.
2. On the **Host** tab, complete the following values:
 - ✦ **Name:** Choose a suitable name for your host. For example, *host1.example.org*.
 - ✦ **Host Group:** Select RHEL6Server-x86_64



Note

New hosts inherit the default values configured for the host group. This means you can quickly build a host without reentering those values.

- ✦ **Content Source:** The \$FQDN of your Satellite. This is automatically selected based on the Host Group.
3. On the **Network** tab, complete the following values:
 - ✦ **MAC Address:** The MAC address of the new host. The Satellite server reserves a DHCP address using this value. Ensure you enter it correctly.
 - ✦ **Subnet: Provisioning_Net** This value is automatically populated.
 - ✦ **IP Address:** This value is automatically populated.

Do not make any changes to the **Puppet Classes**, **Operating System**, **Parameters**, or **Additional Information** tabs.
 4. Click **Submit**.
 5. Power on your host (either physical host or virtual machine); it will PXE-boot and begin its installation process.

See <https://access.redhat.com/articles/1169613> for more information.

6.2. Provisioning a Host Using a Boot Disk

The Satellite network provisioning model is usually based on PXE, which requires DHCP and TFTP services. Because not all Satellite deployments have these services available, the boot disk provisioning feature provides host-specific, full host, and generic boot disk image types to enable provisioning in such deployments.

Each boot disk image type has its own advantages, but all are designed for environments without control of the network infrastructure; consequently, no DHCP reservations or TFTP settings are needed.

Boot images are written as hybrid ISO images (usable as ISO files or USB disks), and can be booted either from physical media or from a virtual disk or CD.

Table 6.1. Comparison of Boot Image Type Characteristics

Type	Generic	DHCP Required	DHCP Reservation	Preregister Host	Operating System Specific
Host-specific image	No	No	No	Yes	No
Full host image	No	Yes	No	Yes	Yes
Generic image	Yes	Yes	No	Yes	No

6.2.1. Prerequisites

All of the requisite packages for the Satellite boot disk feature are normally installed by default. Ensure you address the following conditions before proceeding:

- Regardless of image type you use, the host must be registered to Satellite before you boot from the image. Hosts are identified by their MAC or IP address to provide the correct provisioning template if the host is in build mode.
- For host-specific images, ensure the host IP addresses and subnets are populated, and the subnet's gateway, subnet mask, and DNS resolvers are correctly configured. Navigate to **Infrastructure** → **Subnets** to configure these values.
- To permit access to images for non-administrative users, add the "Boot disk access" role to a user or add the "download_bootdisk" permission to an existing role.
- Ensure that the appropriate iPXE provisioning templates are available. Red Hat Enterprise Linux uses the "Kickstart default iPXE" template.



Note

gPXE images are currently unsupported.

Host and generic image types are based on iPXE technology, which supports a different set of hardware drivers from PXELinux. See http://ipxe.org/appnote/hardware_drivers for the list of supported hardware.

If you encounter issues with iPXE, full host images contain built-in kernels and RAM disks and can load on any kind of network card, including those without PXE support.

- Ensure that the appropriate standard templates are available. Red Hat Enterprise Linux uses "Kickstart RHEL default" as the provisioning template.



Important

If you are not using the default Satellite kickstart provisioning templates, then ensure the templates you use provide the static IP details required to configure the operating system. For a kickstart file, you can use the following configuration:

```
network --bootproto <%= @static ? "static" : "dhcp" %> --hostname <%=
@host %> <%= " --ip=#{@host.ip} --netmask=#{@host.subnet.mask} --
gateway=#{@host.subnet.gateway} --nameserver=#
{@host.subnet.dns_primary},#{@host.subnet.dns_secondary}" if @static %>
```

6.2.2. Creating Boot Disk Images

This section describes how to create host-specific, full host, and generic boot disk images. You can use either the web UI or the command line to create the images; both methods are described.



Note

To create images using the command line, ensure the *ruby193-rubygem-foreman_bootdisk* package is installed. This is normally installed by default.

6.2.2.1. Creating Host-specific Images

You can use the host and subnet data in Satellite to create host-specific images with static networking. The behavior is dynamic; the image chain-loads from Satellite and consequently the current operating system and build state are provided by Satellite instead of being stored in the image.

Procedure 6.2. To Create a Host-specific Image Using the Web UI:

1. Navigate to **Hosts** → **All hosts** and click the appropriate host name.
2. Click **Boot disk** and then click **Host *hostname* image**.

To create a host-specific image using the **hammer** CLI tool, run the following command:

```
# hammer bootdisk host --host client.example.com
```

To create a host-specific image from the command line on the Satellite server, run the following command:

```
# foreman-rake bootdisk:generate:host NAME=client.example.com OUTPUT=/tmp
```

Set the value of **OUTPUT** to a suitable destination path, either a directory or a file. The **foreman** user must have write access to the specified destination.

6.2.2.2. Creating Full Host Images

Full host images are similar to host-specific images, but instead of chain loading from Satellite, these images contain the initial operating system boot loader. This is useful for hosts that fail to chain load, but the downside is that the image may become out of date if the host operating system, boot loader, or templates change, or if build tokens are required and they expire.

Procedure 6.3. To Create a Full Host Image Using the Web UI:

1. Navigate to **Hosts** → **All hosts** and click the appropriate host name.
2. Click **Boot disk** and then click **Full Host *hostname* image**.

Full host images take longer to create because the process downloads the operating system boot loaders, which can be quite large.

To create a full host image using the **hammer** CLI tool, run the following command:

```
# hammer bootdisk host --host client.example.com --full true
```

To create a full host image from the command line on the Satellite server, run the following command:

```
# foreman-rake bootdisk:generate:full_host NAME=client.example.com
```

6.2.2.3. Creating Generic Images

Generic images provide a single ISO file that can be used by all registered hosts. IP address details cannot be stored inside these images, however, which means that the network must provide a DHCP pool. You use the generic image to boot the host, which then contacts Satellite for the template of a registered host matching a MAC address or the IP the host was assigned by DHCP.

The installation can continue using either the DHCP-assigned or static IP address, depending on how the operating system iPXE template is configured. You can use the kickstart file to specify additional network configuration options.

Procedure 6.4. To Create a Generic Image from the Web UI:

1. Navigate to **Hosts** → **All hosts** and click the appropriate host name.
2. Click **Boot disk** and then click **Generic image**.

To create a generic image using the **hammer** CLI tool, run the following command:

```
# hammer bootdisk generic
```

To create a generic image from the command line on the Satellite server, run the following command:

```
# foreman-rake bootdisk:generate:generic
```

6.2.2.4. Creating USB Images

Whenever you create an ISO file it is also passed through the **isohybrid** command, which means that the resulting file is also bootable as a disk, and suitable for copying to a USB device.

To copy the ISO file to a USB device, run the following command. Ensure the device name and input file are correct for your environment:

```
# dd if=fqdn.iso of=/dev/sdX
```

6.3. Provisioning Hosts with Static IP Addresses

Red Hat Satellite 6 expects all systems to be configured for DHCP, because it reserves a DHCP record for a given MAC address. You can also provision hosts with static IP addresses, using either custom provisioning templates, host parameters, or based on subnet information.

6.3.1. Using Custom Templates to Assign Static IP Addresses

You can create a custom provisioning template that provides static IP support for all provisioned hosts. You can copy the PXE configuration template and associate it with a different operating system, for example "RHEL 7.1 static". This means that when you kickstart a system with this template it receives a static IP. This method is currently required because Satellite 6.0 and 6.1 do not support a choice of dynamic or static IP configuration in the existing "Create New Host" work flow.

This method requires that you edit your PXE template to enable static networking. As described in the following example, edit the PXE template and add `&static=yes` to the end of each instance of `foreman_url('provision')`.

Procedure 6.5. To Edit a PXE Template:

1. Navigate to **Hosts** → **All Hosts** and click the name of the host whose template you want to edit.
2. Click the **Templates** tab to display the list of available template types.
3. Click **Edit** for the **PXELinux Template** type. The template displays in the template editor.
4. Make the required changes and then click **Submit**. See [Example 6.1, "Editing the PXE Template to Assign Static IP Addresses"](#) for an example of how to update a PXE template.

Example 6.1. Editing the PXE Template to Assign Static IP Addresses

```
default linux
label linux
kernel <%= @kernel %>
<% if @host.operatingsystem.name == 'Fedora' and
@host.operatingsystem.major.to_i > 16 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%>&static=yes
ks.device=bootif network ks.sendmac
<% elsif @host.operatingsystem.name != 'Fedora' and
@host.operatingsystem.major.to_i >= 7 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%>&static=yes
network ks.sendmac
<% else -%>
```

```
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%>&static=yes
ksdevice=bootif network kssendmac
<% end -%>
IPAPPEND 2
```

6.3.2. Using Host Parameters to Assign Static IP Addresses

You can use host parameters to set up static IP addresses on a per-host basis. When you create a new host, navigate to **Hosts** → **New host** and enter the required parameters on the **Parameters** tab.

[Example 6.2, “Using Host Parameters to Assign Static IP Addresses”](#) demonstrates using an **if** statement in conjunction with host parameters to set a static IP address.

Example 6.2. Using Host Parameters to Assign Static IP Addresses

```
<% if @host.params['my_param_deploy_static'] == "true" -%>
  network --bootproto <%= "static --ip=#{@host.ip} --netmask=#
{@host.subnet.mask} --gateway=#{@host.subnet.gateway} --nameserver=#
[[@host.subnet.dns_primary,@host.subnet.dns_secondary].reject{|n|
n.blank?}.join(',')]" %> --hostname <%= @host %>
<% else -%>
  network --bootproto 'dhcp' --hostname <%= @host %>
<% end -%>
```

See the [Red Hat Satellite 6 User Guide](#) for more information.

6.3.3. Using Subnets to Set Static IP Addresses

You can configure Red Hat Satellite to provision hosts with a static IP address based on the host's subnet. When you set up subnets, you can specify either DHCP or static boot modes. The Red Hat Enterprise Linux installation program (Anaconda) uses this value to determine whether to assign a static IP address or an address from the DHCP pool. Specify "Static" to ensure that all hosts provisioned in this subnet receive static IP addresses.

Procedure 6.6. To Specify Static Boot Mode by Default:

1. Navigate to **Infrastructure** → **Subnets** and click the name of the subnet that you want to configure. To create a new subnet, follow the steps in [Procedure 2.4, “To Select the Default Subnet:”](#).
2. On the **Subnet** tab, select **Static** from the **Boot mode** drop-down list, and then click **Submit**.

Whenever you create a new host and assign it to this subnet, it uses a static IP address by default.

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 Host

A *Content Host* is the part of a host that manages tasks related to content and subscriptions.

Content View

A *Content View* is a definition of content that combines products, packages, and Puppet modules with capabilities for intelligent filtering and creating snapshots. Content Views are a refinement of the combination of channels and cloning from Red Hat Satellite 5.

External Node Classifier

An *External Node Classifier* is a Puppet construct that provides additional data for a Puppet Master to use when configuring Hosts. Red Hat Satellite 6 acts as an External Node Classifier to Puppet Masters in a Satellite deployment.

Facter

Facter is a program that provides information (facts) about the system on which it is run; for example, Facter can report total memory, operating system version, architecture, and more. Puppet modules enable specific configurations based on host data gathered by Facter.

Hammer

Hammer is a command line tool for Red Hat Satellite 6. Use Hammer to manage Red Hat Satellite 6 as a standard CLI, for scripts, and also through an interactive shell.

Hiera

Hiera is a key/value look-up tool for configuration data which allows keeping site-specific data out of puppet manifests.

Host

A *Host* refers to any system, either physical or virtual, that Red Hat Satellite 6 manages.

Host Collection

A *Host Collection* is equivalent to a Satellite 5 *System Group*, that is, a user defined group of one or more Hosts.

Host Group

A *Host Group* is a template for building a Host. This includes the content view (which defines the available RPM files and Puppet modules) and the Puppet classes to apply (which ultimately determines the software and configuration).

Location

A *Location* is collection of default settings that represent a physical place. These can be nested so that you can set up an hierarchical collection of locations. For example, you can set up defaults for "Middle East", which are refined by "Tel Aviv", which are further refined by "Data Center East", and then finally by "Rack 22".

Library

The *Library* contains every version, including the latest synchronized version, of the software that the user will ever deploy. For an Information Technology Infrastructure Library (ITIL) ^[1] organization or department, this is the Definitive Media Library ^[2] (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](#) ^[3]
- [The Structure of Satellite Certificates \(Classic Style of Certificates\)](#) ^[4]

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.

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

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

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

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

Appendix B. Revision History

Revision 1.2-10	Tue Jul 14 2015	David O'Brien
Remove beta and draft status. Publish for technical review.		
Revision 1.2-9	Fri Jun 19 2015	David O'Brien
Add Beta web label and book title.		
Revision 1.2-8	Sun Jun 14 2015	David O'Brien
Public Beta release.		
Revision 1.2-7	Sun Jun 14 2015	David O'Brien
Add section on how to use static IP.		
Revision 1.2-6	Fri May 15 2015	David O'Brien
Rebuild with updated product version.		
Revision 1.2-5	Mon May 11 2015	David O'Brien
Build for technical review.		
Revision 1.2-4	Thu Apr 16 2015	David O'Brien
Document how to use static IP with Satellite 6.1.		
Revision 1.2-3	Wed Apr 8 2015	Megan Lewis
Updated brand. Removed Preface.		
Revision 1.2-2	Fri Apr 3 2015	David O'Brien
Update procedure for creating a manifest to match new portal design.		
Revision 1.2-1	Tue Mar 31 2015	David O'Brien
BZ 1207429: Fix error in procedure "Creating Host Groups"; content source should refer to Capsule. Clarify confusing reference to "PTR zone file". BZ 1203878: Update "RH Common" repository names to "Satellite Tools". BZ 1168240: Clarify section on "domain" vs "zone". BZ 1176466: Clarify required options to katello-installer command. Clarify procedure on how to create host groups. BZ 1183657: Add "puppet module" and "catalog" to Glossary. Fix entities in "Report a bug" links.		
Revision 1.2-0	Thu Feb 22 2015	David O'Brien
BZ 1184671: Fix error in katello-installer option.		
Revision 1.1-1	Mon Dec 22 2014	David O'Brien
Update section on creating a host for Satellite to reference correct versions. Fix relative link in section "Creating Custom Products and Repositories".		
Revision 1.1-0	Wed Dec 10 2014	David O'Brien
Update section on creating manifests to reference more detailed information.		

Revision 1.0-0	Wed Nov 26 2014	David O'Brien
Implement updates from QA review. First public release.		
Revision 0.1-1	Thu Nov 20 2014	David O'Brien
Add Glossary of terms. Added definition of Content View as part of Creating Content Views in Content Management and Promotion. Added xref to referenced procedures. Updated section on initial configuration to include administrator user name and password. Updated chapter on Provisioning Hosts. Added Abstract. Draft chapter on Configuring Provisioning Templates. Draft section on Content Management and Promotion. Revised sections on importing subscriptions and synchronizing content.		
Revision 0.1-0	Fri Oct 17 2014	David O'Brien
Added first draft of chapter "Configuring Services". Added rough draft of chapter "Introduction to Provisioning". Added skeleton TOC with basic sections. Initial creation by publican.		