Red Hat Satellite 6.4

Quick Start Guide

Installing, configuring, and provisioning physical and virtual hosts from Red Hat Satellite Servers.

Red Hat Satellite 6.4 Quick Start Guide

Installing, configuring, and provisioning physical and virtual hosts from Red Hat Satellite Servers.

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

Legal Notice

Abstract

The Red Hat Satellite Quick Start Guide is designed to help you quickly install and configure Red Hat Satellite Server, as well as provision physical and virtual hosts. This guide is aimed primarily at Satellite administrators with sound networking knowledge and skills.

Table of Contents

CHAPTER 1. SHOULD YOU USE THIS GUIDE?

The Quick Start Guide enables you to install and configure your Satellite Server based on a set of assumptions about your environment. This includes setting up the required topology and providing all of the configuration information needed to provision hosts on your network.

You should use this guide if the following assumptions apply to your environment:

- You are performing a fresh installation of Red Hat Satellite.

- You will have a single Satellite Server deployment.

- You can connect to the Red Hat Content Network to obtain packages and receive updates.

- You have an existing external DNS server that you want to connect to Satellite Server.

- Your environment does not use DHCP, or uses external DHCP services.

- You have an existing host that you want to register to Satellite to use for patching.

- You want to use PXE-less discovery to provision hosts.

- You use a kickstart-based deployment.

- Your deployment uses simple Content Views.

- You are using IPv4.

Some of the tasks in this guide use example information. You should ensure that you use values that are specific to your environment.

If your environment does not fit with these assumptions, you can use the following resources to assist you with installing Satellite Server:

- Installing Satellite Server from a Connected Network

- Installing Satellite Server from a Disconnected Network

- 10 Steps to Build an SOE: How Red Hat Satellite 6 Supports Setting Up a Standard Operating Environment

- Red Hat Satellite 6.4 Provisioning Guide

CHAPTER 2. INSTALLING SATELLITE SERVER

2.1. PREPARING FOR SATELLITE SERVER INSTALLATION

Ensure that you meet all the prerequisites before installing Satellite Server.

Prerequisites

- You must create a Red Hat Enterprise Linux 7 host before you can install and configure Satellite Server. Red Hat Enterprise Linux version 7.5 or later is supported. For more information about installing Red Hat Enterprise Linux 7, see the Red Hat Enterprise Linux 7 Installation Guide .

- Ensure that your environment meets the requirements for installation, including meeting storage requirements, and ensuring that network ports and firewalls are open and configured. For more information, see the Preparing your Environment for Installation in *Installing Satellite Server from a Connected Network*.

2.2. REGISTERING TO RED HAT SUBSCRIPTION MANAGEMENT

Registering the host to Red Hat Subscription Management 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 (RHSCL), and Red Hat Satellite.

1. Register your Satellite Server, using your user name and password.

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

2.3. IDENTIFYING AND ATTACHING THE SATELLITE SUBSCRIPTION TO THE HOST

After you have registered your host, you need to identify your Satellite subscription Pool ID. The Pool ID enables you to 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 (RHSCL), and Red Hat Satellite. This is the only subscription required.

1. To identify your Satellite subscription, enter the following command:

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

   The outputs displays something similar to the following:

   ```
   Subscription Name: Red Hat Satellite
   Provides:        Red Hat Satellite 6
                    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
```

2. Make a note of the Pool ID so that you can attach it to your Satellite host. Your Pool ID will be different than the example provided.

3. To attach your subscription to your Satellite Server, enter the following command, using your Pool ID:

```
# subscription-manager attach --pool=8a85f9874152663c0541943739717d11
```

The outputs displays something similar to the following:

```
Successfully attached a subscription for: Red Hat Satellite
```

4. To verify that the subscriptions are successfully attached, enter the following command:

```
# subscription-manager list --consumed
```

The outputs displays something similar to the following:

```
+-------------------------------------------+
    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
                 Red Hat Satellite 6
                 Red Hat Software Collections (for RHEL Server)
                 Red Hat Satellite Capsule
                 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:          16538561912506999363
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
```

2.4. CONFIGURING REPOSITORIES

1. Disable all existing repositories.

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

2. Enable the Red Hat Satellite, Red Hat Enterprise Linux, and Red Hat Software Collections repositories.

   ```
   # subscription-manager repos --enable=rhel-7-server-rpms \
   --enable=rhel-server-rhscl-7-rpms \
   --enable=rhel-7-server-satellite-6.4-rpms \
   --enable=rhel-7-server-satellite-maintenance-6-rpms \
   --enable=rhel-7-server-ansible-2.6-rpms
   ```

3. Clear out any metadata.

   ```
   # yum clean all
   ```

4. Verify that the repositories have been enabled.

   ```
   # yum repolist enabled
   ```

2.5. INSTALLING SATELLITE SERVER

1. Update all packages:

   ```
   # yum update
   ```

2. Install the Satellite Server package:

   ```
   # yum install satellite
   ```

3. Install Satellite Server and perform the initial configuration:

   ```
   # satellite-installer --scenario satellite \
   --foreman-admin-username admin \
   --foreman-admin-password redhat \
   --foreman-proxy-puppetca true \
   --foreman-proxy-tftp true \
   --enable-foreman-plugin-discovery
   ```

2.6. CREATING A SUBSCRIPTION ALLOCATION IN CUSTOMER PORTAL

To create a manifest, you must first create an allocation and then export the manifest from the allocation.

You can access your subscription information on the Red Hat Customer Portal. You can also assign subscriptions for use in on-premise management applications, such as Red Hat Satellite, using subscription allocations.

1. Open https://access.redhat.com/ in your browser and log in to your Red Hat account.

2. Navigate to **Subscriptions** in the upper-left corner of the Customer Portal.

3. Navigate to **Subscription Allocations**.

4. Click **Create New subscription allocation**

5. In the **Name** field, enter a name.

6. From the **Type** list, select the type and version that corresponds to your Satellite Server.

7. Click **Create**.

2.6.1. Adding Subscriptions to an Allocation

The following procedure explains how to add subscriptions to an allocation.

1. Navigate to **Subscriptions Allocations**.

2. Select the name of the subscription you want to change.

3. Click the **Subscriptions** tab.

4. Click **Add Subscriptions**.

5. A list of your Red Hat product subscriptions appears. Enter the **Entitlement Quantity** for each product.

6. Click **Submit** to complete the assignment.

When you have added subscriptions to the allocation, export the manifest file.

2.7. EXPORTING A SUBSCRIPTION MANIFEST FROM THE CUSTOMER PORTAL

While viewing a subscription allocation that has at least one subscription, you can export a manifest in either of two places:

- From the **Details** tab, under the **Subscription** section, by clicking the **Export Manifest** button.

- From the **Subscriptions** tab, by clicking the **Export Manifest** button.

When the manifest is exported, the Customer Portal encodes the selected subscriptions certificates and creates a .zip archive. This is the Subscription Manifest, which can be uploaded into Satellite Server.

2.8. UPLOADING A MANIFEST TO YOUR SATELLITE SERVER

Uploading a Manifest Using the Web UI

1. Verify that you are in the correct organization.

2. Click **Content** > **Red Hat Subscriptions**.

3. Click **Manage Manifest** to open the Subscriptions page.

4. Click **Choose file** and select the manifest you created.

5. Click **Upload** to upload the manifest to Satellite Server.

Uploading a Manifest Using Hammer CLI

1. Upload a manifest to Satellite Server.

```
# hammer subscription upload \
--file ~/manifest_file.zip \
--organization "Default_Organization"
```

CHAPTER 3. ASSOCIATING OBJECTS WITH THE DEFAULT ORGANIZATION AND LOCATION

3.1. CREATING A NEW DOMAIN

Use this procedure to create a new domain.

For Web UI Users

1. Navigate to **Infrastructure** > **Domains** and click **Create Domain**.

2. In the **DNS Domain** field, enter the full DNS domain name.

3. In the **Fullname** field, enter the plain text name of the domain.

4. Click the **Parameters** tab.

5. Click **Add Parameter** and fill in the **Name** and **Value** fields.

6. Click the **Locations** tab, and add the location where the domain resides.

7. Click the **Organizations** tab, and add the organization that the domain belongs to.

8. Click **Submit** to save the changes.

For CLI Users

Use the **hammer domain create** command to create a domain:

```
# hammer domain create --name "domain_name.com" \
--description "My example domain" \
--dns-id 1 --locations "Default_Location" \
--organization "Default_Organization"
```

> **NOTE**
>
> In this example, the **--dns-id** option uses 1, which is the ID of the Satellite Server's Integrated Capsule.

3.2. ASSOCIATING THE DOMAIN WITH THE DEFAULT ORGANIZATION

Associating the Domain with the Default Organization Using the Web UI

1. On the Main Menu, click **Infrastructure → Domains**.

2. Select the domain in the **Description** column.

3. On the **Locations** tab, click **Default_Location** to add it to the Selected items list.

4. On the **Organizations** tab, click Default_Organization to add it to the Selected items list.

5. Click **Submit**.

Associating the Domain with the Default Organization Using Hammer CII

1. Associate the domain to your organization.

   ```
   $ hammer organization add-domain --name "Default_Organization" --domain
   "domain_name.com"
   ```

2. Associate the domain to your location.

   ```
   $ hammer location add-domain --name "Default_Location" --domain "domain-name.com"
   ```

3.3. CONFIGURING THE SUBNET

Configuring the Subnet Using the Web UI

1. On the Main Menu, click **Infrastructure → Subnets**.

2. Click **New Subnet** and enter the information specific to your environment.

   a. If you use DHCP, enter the following information:

   - 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

   - IPAM: None

   - Start of IP Range: 172.17.13.100

   - End of IP Range: 172.17.13.150

   - VLAN ID: Leave blank

   - Boot Mode: DHCP

   b. If you use static IP addresses, enter the following information:

   - 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

- IPAM: None

- Start of IP Range: 172.17.13.100

- End of IP Range: 172.17.13.150

- VLAN ID: Leave blank

- Boot Mode: Static

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 TFTP and Discovery Proxy capsule to reflect the host name of the Satellite Server.

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.

Configuring the Subnet Using Hammer CLI

1. Create a subnet.

   a. Create a subnet using DHCP.

      ```
      # hammer subnet create --name Provisioning_Net \
      --organization "Default_Organization" \
      --locations 'RDU' \
      --domain-ids 1 \
      --boot-mode DHCP \
      --network 172.17.13.0 \
      --mask 255.255.255.0 \
      --ipam None \
      --dns-primary 172.17.13.1 \
      --gateway 172.17.13.1 \
      --from 172.17.13.100 \
      --to 172.17.13.150 \
      --tftp-id 1
      ```

   b. Create a subnet using static IP addresses.

      ```
      # hammer subnet create --name Provisioning_Net \
      --organization "Default_Organization" \
      --locations 'RDU' \
      --domain-ids 1 \
      --boot-mode Static \
      --network 172.17.13.0 \
      --mask 255.255.255.0 \
      --ipam None \
      --dns-primary 172.17.13.1 \
      ```

```
--gateway 172.17.13.1 \
--from 172.17.13.100 \
--to 172.17.13.150 \
--tftp-id 1
```

> **NOTE**
>
> You cannot set the discovery proxy using Hammer CLI.

CHAPTER 4. IMPORTING SUBSCRIPTIONS AND SYNCHRONIZING CONTENT

4.1. ENABLING RED HAT REPOSITORIES

You must enable the required Red Hat repositories to support provisioning a Red Hat Enterprise Linux host.

> **NOTE**
>
> For a list of repositories required to install Red Hat Satellite 6.4, see Content Delivery Network Repositories in the *Release Notes*.

Select the appropriate release for your environment. You can use the examples in this topic to enable the following repositories:

- **Red Hat Enterprise Linux Server (Kickstart)**.
  Enable this repository to kickstart a host.

- **Red Hat Enterprise Linux Server (RPMs)**.
  Enable this repository for ongoing content and errata.

- **Red Hat Enterprise Linux Server - Satellite Tools (RPMs)**
  Enable this repository to support software, such as Puppet and katello-agent.

Enabling Red Hat Enterprise Linux 7 Server Repositories Using the Web UI

1. In the Satellite web UI, navigate to **Content** > **Red Hat Repositories**.

2. Use the Search field to enter the following repository name: **Red Hat Enterprise Linux 7 Server (RPMs)**.

3. In the Available Repositories pane, click on **Red Hat Enterprise Linux 7 Server (RPMs)** to expand the repository set.

4. For the **x86_64 7.5** entry, click the **Enable** icon to enable the repository.

5. Use the Search field to enter the following repository name: **Red Hat Satellite Tools 6.4 (for RHEL 7 Server) (RPMs)**.

6. In the Available Repositories pane, click on **Red Hat Satellite Tools 6.4 (for RHEL 7 Server) (RPMs)** to expand the repository set.

7. For the **x86_64** entry, click the **Enable** icon to enable the repository.

8. Use the Search field to enter the following repository name: **Red Hat Enterprise Linux Server 7 (Kickstart)**.

9. In the Available Repositories pane, click on **Red Hat Enterprise Linux 7 Server (Kickstart)** to expand the repository set.

10. For the **x86_64 7.5** entry, click the **Enable** icon to enable the repository.

Enabling Red Hat Enterprise Linux 7 Server Repositories Using Hammer CLI

1. Enable your Kickstart repository.

   ```
   # hammer repository-set enable --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --basearch='x86_64' \
   --releasever='7Server' \
   --name 'Red Hat Enterprise Linux 7 Server (Kickstart)'
   ```

2. Enable your RPMs repository.

   ```
   # hammer repository-set enable --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --basearch='x86_64' \
   --releasever='7Server' \
   --name 'Red Hat Enterprise Linux 7 Server (RPMs)'
   ```

3. Enable your Satellite Tools repository.

   ```
   # hammer repository-set enable --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --basearch='x86_64' \
   --name 'Red Hat Satellite Tools {ProductVersion} (for RHEL 7 Server) (RPMs)'
   ```

4.2. CREATING A CUSTOM PRODUCT

You can enable a repository with custom packages, but you must first manually create a Product for the repository.

Creating Custom Products Using the Web UI

1. Click **Content → Products** and then click **New Product**.

2. Enter a name for your custom Product.

3. Click **Save**.

4. After the screen refreshes, click **Create Repository**.

5. Enter **Puppet Modules** in the **Name** field.

6. In the **Type** field, **Puppet**. Leave the **URL** field blank.

7. Click **Save**.

Uploading Puppet Modules to the Repository Using the Web UI

You can now 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. All of the content from Puppet Forge is available on your Satellite Server. 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. If you want to upload packages with large files, use the Hammer CLI.

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 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 Package** section, click **Choose Files**, and navigate to the motd module that you downloaded.

5. Click **Upload**.

Creating Custom Products and Enabling Repositories Using Hammer CLI

1. Create a custom Product.

   ```
   $ hammer product create --name "product_name" --organization "Default_Organization"
   ```

2. Create a new repository under the custom Product.

   ```
   $ hammer repository create --name "repo_name" \
   --organization "Default_Organization" \
   --product "product_name" \
   --content-type cont_type \
   --publish-via-http true \
   --url "repo_url"
   ```

   For the **--content-type** option, specify whether you want **yum**, **puppet**, or **docker** content. The command accepts only lower-case text.

3. Upload packages to the custom repository.

   ```
   $ hammer repository upload-content --product "product_name" \
   --organization "Default_Organization" \
   --id "repo_id" \
   --path path_to_dir
   ```

4.3. SYNCHRONIZING CONTENT

You can synchronize repositories from the Red Hat Content Delivery Network to your Satellite, which also applies to synchronizing custom repositories (such as Yum or Puppet) that contain a repository URL.

Synchronizing Content Using Red Hat Enterprise Linux 7 Server Using the Web UI

1. Click **Content → Sync Status** to display the list of available Products.

2. Navigate to **Red Hat Enterprise Linux Server→ 7Server → x86_64**.

3. Select the following Products:

   - Red Hat Enterprise Linux 7 Server RPMs x86_64 6Server.

   - Red Hat Enterprise Linux 7 Server – Satellite Tools 6.4 RPMs x86_64.

4. Navigate to **Red Hat Enterprise Linux Server→ 7.5 → x86_64** and select **Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.5**.

5. Click **Synchronize Now**.
   The synchronization could take several hours, depending on available bandwidth.

Synchronizing Content Using Hammer CLI

1. Synchronize your Kickstart repository.

   ```
   $ hammer repository synchronize --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --name  'Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.5' \
   --async
   ```

2. Synchronize your RPMs repository.

   ```
   $ hammer repository synchronize --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --name 'Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server' \
   --async
   ```

3. Synchronize your Satellite Tools repository.

   ```
   $ hammer repository synchronize --organization "Default_Organization" \
   --product 'Red Hat Enterprise Linux Server' \
   --name 'Red Hat Satellite Tools 6.4 for RHEL 7 Server RPMs x86_64' \
   --async
   ```

CHAPTER 5. MANAGING AND PROMOTING CONTENT

5.1. 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).

1. Click **Content → Life Cycle Environments** to open the **Life Cycle Environment Paths** page.

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. Enter a name in the **Name** Field.
   The label is automatically populated with the same name, but you can change it to suit your needs. You can also add a description of your environment.

4. Click **Save**.

Creating Life Cycle Environments Using Hammer CLI

1. Create the life cycle environment.

   ```
   $ hammer lifecycle-environment create --name Development \
   --prior Library \
   --organization "Default_Organization"
   ```

5.2. CREATING SIMPLE 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. You can use Content Views to customize content that is made available to client hosts.

Creating Content Views Using the Web UI

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. Clear the **Composite View?** check box and click **Save**.

Creating Content Views Using Hammer CLI

1. Create the Content View.

   ```
   $ hammer content-view create --organization "Default_Organization" \
   --name 'RHEL7_Base' \
   --label rhel7_base \
   --description 'Core Build for RHEL 7'
   ```

5.3. ADDING RED HAT ENTERPRISE LINUX REPOSITORIES

You can add Red Hat Enterprise Linux repositories to a Content View. Optionally, you can use the procedure described in this topic to add Red Hat repositories, custom repositories, or both to a Content View.

In the following example, all content is included in the published Content View. Optionally, you can create filters to control whether content is included or excluded from the published Content View.

Adding Repositories Using Web UI

1. Navigate to **Content** > **Content Views**, and then select the Content View you want to edit.

2. In the **Content View** pane, click the **Yum Content** tab, and then select **Repositories**.

3. In the **Add** tab, select the following repositories from the **Repository Selection** list:

   - Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.5

   - Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server

   - Red Hat Satellite Tools 6.4 for RHEL 7 Server RPMs x86_64

4. Click **Add Repositories**.
   To see the list of repositories added to your Content View, click the **List/Remove** tab.

Adding Repositories Using Hammer CLI

1. Add the Kickstart repository.

   ```
   $ hammer content-view add-repository --organization "Default_Organization" \
   --name 'RHEL7_Base' \
   --product 'Red Hat Enterprise Linux Server' \
   --repository 'Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.5'
   ```

2. Add the RPMs repository.

   ```
   $ hammer content-view add-repository --organization "Default_Organization" \
   --name 'RHEL7_Base' \
   --product 'Red Hat Enterprise Linux Server' \
   --repository 'Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server'
   ```

3. Add the Satellite Tools repository.

   ```
   $ hammer content-view add-repository --organization "Default_Organization" \
   --name 'RHEL7_Base' \
   --product 'Red Hat Enterprise Linux Server' \
   --repository 'Red Hat Satellite Tools 6.4 for RHEL 7 Server RPMs x86_64'
   ```

5.4. ADDING PUPPET MODULES TO A CONTENT VIEW

Puppet Modules are self–contained bundles of code and data that you can use to manage resources such as users, files, and services.

Adding Puppet Modules Using the Web UI

1. Go to the **Content Views** page.

2. Click **Content → Content Views**.

3. 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.

4. Click **Select Version** and select the 'motd' module.

5. Click **Select Version** next to the version of the module that you want to add.
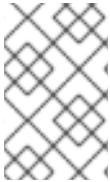
> **NOTE**
>
> If you select "Use Latest" when you select which Puppet module version to use, it means that whenever a new version of the Content View is published, the latest version of that module is included in the published view.

Adding Puppet Modules Using Hammer CLI

1. Add a Puppet module.

   ```
   $ hammer content-view puppet-module add \
   --content-view cv_name \
   --name module_name
   ```

5.5. PUBLISHING A CONTENT VIEW

After you have created Content Views and Puppet Modules, you can publish them to the Library.

Publishing a Content View Using the Web UI

1. Click **Content → Content Views**.

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**.
   The list of available promotion paths (Library → Dev → QA) displays.

6. Select the check box for the Dev environment, and then click **Promote Version**.

Publishing a Content View Using Hammer CLI

1. Publish your Content View.

```
$ hammer content-view publish --organization "Default_Organization" \
--name RHEL7_Base \
--description 'Initial Publishing'
```

2. Promote your Content View.

```
$ hammer content-view version promote \
--organization "Default_Organization" \
--content-view RHEL7_Base \
--to-lifecycle-environment Development
```

5.6. CREATING AND EDITING ACTIVATION KEYS

After you have successfully published a Content View, you need to create an Activation Key to associate with a Host Group.

Creating and Editing an Activation Key Using the Web UI

1. On the main menu, click **Content → Activation Keys → New Activation Key**.

2. In the Name field, enter a name.

3. If applicable, 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 your environment.

5. In the **Content View** drop-down list, select the RHEL 7 x86_64 Content View, and then click **Save**.

6. On the **Activation Keys** page, click the **Subscriptions** tab, then click the **Add** tab to display the list of available subscriptions.

7. From the list of available subscriptions, select the subscriptions you want to add.

8. Click **Add Selected**.

Creating and Editing an Activation Key Using Hammer CLI

1. Create the activation key.

```
$ hammer activation-key create --name ak-Reg_To_Dev_EL7 \
--organization "Default_Organization" \
--content-view RHEL7_Base \
--lifecycle-environment Development
```

2. Add a subscription.

```
$ hammer subscription list --organization "Default_Organization"
$ hammer activation-key add-subscription \
--name ak-Reg_To_Dev_EL7 \
```

```
--subscription-id 8 \
--organization "Default_Organization"
```

3. Set Product content to enable the repository.

```
$ hammer activation-key content-override --name ak-Reg_To_Dev_EL7 \
--organization "Default_Organization" \
--content-label rhel-7-server-satellite-tools-6.4-rpms \
--value 1
```

If you do not plan to use Satellite Server to patch your systems, continue to Creating Provisioning Templates.

CHAPTER 6. PATCHING YOUR SYSTEMS

To patch your systems through Satellite Server, you need to register your systems first, and then choose to install Katello agent or use remote execution. Remote execution is enabled by default. This chapter covers both methods. Choose the option that suits your environment

6.1. REGISTERING EXISTING HOSTS

This section explains how to register Red Hat Enterprise Linux Hosts and Atomic Hosts.

6.1.1. Registering an Existing Red Hat Enterprise Linux Host

1. On the host, clear any old data to ensure updated data is uploaded correctly.

   ```
   # subscription-manager clean
   ```

2. Install the katello-ca-consumer-latest RPM.

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

   This installs the proper certificates which allow communication between the client and the Satellite Server.

3. Register the host.

   ```
   # subscription-manager register --org="Default_Organization" \
   --activationkey=ak-Reg_To_Dev_EL7
   ```

6.1.2. Registering an Atomic Host

The following procedure explains how to register an Atomic Host with Subscription Manager.

1. Retrieve **katello-rhsm-consumer** from the Satellite Server:

   ```
   [root@atomic_client ~]# wget http://satellite.example.com/pub/katello-rhsm-consumer
   ```

2. Change the mode of **katello-rhsm-consumer** in order to make it executable:

   ```
   [root@atomic_client ~]# chmod +x katello-rhsm-consumer
   ```

3. Run **katello-rhsm-consumer**:

   ```
   [root@atomic_client ~]# ./katello-rhsm-consumer
   ```

4. Register with Red Hat Subscription Manager:

   ```
   [root@atomic_client ~]# subscription-manager register
   ```

NOTE

Because Atomic is functionally an appliance, we do not recommend that you try to install **katello-agent** on it.

6.2. PATCHING YOUR SYSTEM USING KATELLO AGENT

6.2.1. Installing Katello Agent

On the content host, install the **katello-agent** RPM package.

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

The goferd service must be running so that the Red Hat Satellite Server or Capsule Server can provide information about errata that are applicable for content hosts.

Ensure **goferd** is running:

- On Red Hat Enterprise Linux 6, run the following command:

  ```
  # service goferd start
  ```

- On Red Hat Enterprise Linux 7, run the following command:

  ```
  # systemctl start goferd
  ```

6.2.2. Applying Errata to Content Hosts

Applying Errata to Content Hosts Using the Web UI

1. Go to **Hosts → Content Hosts** and click on **auth01.example.com**.

2. Select the **Errata** tab to view the list of errata applicable to the content host.

3. From the list, select an errata.

4. Click **Apply Selected**.

5. A confirmation message appears. Click **Apply**.

6. Verify that the errata has been applied to the client.

   ```
   [root@client ~]# yum list-sec
   ```

Applying Errata to Content Hosts Using Hammer CLI

1. List the errata that apply to the **auth01.example.com** host.

   ```
   # hammer host errata list --host auth01.example.com \
   --organization "Default_Organization"
   ```

2. Apply the errata required by the host.

   -

```
# hammer host errata apply --host auth01.example.com \
--organization "Default_Organization" \
--errata-ids errata_id.
```

3. Verify that the errata has been applied to the client.

```
[root@client ~]# yum list-sec
```

6.3. PATCHING YOUR SYSTEM USING REMOTE EXECUTION

6.3.1. Enabling Remote Execution on a Host

During Satellite Server installation, an internal Capsule Server is automatically installed along with a public SSH key. The internal Capsule Server loads the SSH key from **/usr/share/foreman-proxy/.ssh/id_rsa_foreman_proxy**. You can enable remote execution by distributing the public SSH key to a host.

On the Satellite Server, distribute the key to the host.

```
# ssh-copy-id -i ~foreman-proxy/.ssh/id_rsa_foreman_proxy.pub root@auth01.example.com
```

6.3.2. Installing Errata on Your Host

Satellite provides default job templates for executing remote jobs, one of which is for installing errata.

1. Go to **Hosts → All hosts** and click the check box next to the host's name.

2. Click **Select Action** and select **Schedule Remote Job** from the drop-down menu.

3. In the **Job category** drop-down menu, select **Katello**.

4. In the **Job template** drop-down menu, select **Install Errata-Katello SSH Default**

5. In the **errata** field, enter the errata ID.

6. Select **Execute now** and click **Submit**.

CHAPTER 7. FINALIZING THE PROVISIONING CONFIGURATION

7.1. CREATING CUSTOM PROVISIONING TEMPLATES

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

Creating a Provisioning Template Using the Web UI

1. On the main menu, click **Hosts → Provisioning Templates**.

2. In the **Name** column, select **Kickstart Default iPXE** or **PXELinux** in the list of provisioning templates.
   This displays the configuration tabs where you can customize the template.

3. On the **Association** tab, select the appropriate operating system and click **Submit**.

4. In the **Name** column, select the appropriate provisioning template.
   If you use PXE-less discovery, you should select the **Discover Red Hat kexec**provisioning template.

5. On the **Association** tab, select the appropriate operating system and click **Submit**.

6. On the main menu, click **Hosts → Operating Systems** and select the appropriate operating system.
   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 **Katello Kickstart Default** from the provision drop-down list.

10. Select **Discovery Red Hat kexec**from the kexec drop-down list and click **Submit**.

Creating a Provisioning Template Using Hammer CLI

1. Create a provisioning template.

   ```
   $ hammer template create --name template_name --file path_to_template_file
   ```

   If you use PXE-less discovery, you should use the kexec template type.

2. Add the operating system.

   ```
   $ hammer template add-operatingsystem id
   ```

7.2. CREATING HOST GROUPS

You can create and configure a host group, which enables you to provision multiple hosts without the

need to specify the same properties for each host. You can associate an activation key with a host group, which allows the provisioned host to be registered to the Satellite Server and associated with the selected life cycle environment, Content View, subscriptions, and so on.

Creating Host Groups Using the Web UI

1. On the main menu, click **Configure → Host Groups**, and then click **New Host Group**.

2. On the **Host Group** tab, input the following values:

   - Name: RHEL6Server-x86_64

   - Lifecycle Environment: Default_Organization/DEV

   - Content View: RHEL_6_x86_64

   - Content Source: The FQDN of your Capsule (in this case 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. If this field is not automatically populated, it means the Organization and Location associations are not correctly configured.

   - 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 the appropriate key.

9. Click **Submit**.

Creating Host Groups Using Hammer CLI

1. Create the host group.

```
$ hammer hostgroup create --name "hostgroup_name" \
--environment "environment_name" \
--architecture "architecture_name" \
--domain domain_name \
--subnet subnet_name \
--puppet-proxy proxy_name \
--puppet-ca-proxy ca-proxy_name \
--operatingsystem "os_name" \
--partition-table "table_name" \
--medium "medium_name" \
--organization-ids org_ID1,org_ID2... \
--location-ids loc_ID1,loc_ID2...
```

2. Add an activation key.

```
$ hammer hostgroup set-parameter --hostgroup "hostgroup_name" \
--name "kt_activation_keys" \
--value key_name
```

CHAPTER 8. PROVISIONING HOSTS USING A PXE-LESS DISCOVERY IMAGE

Satellite provides two main approaches to provisioning hosts, PXE booting and boot disk provisioning. Boot disk provisioning, or PXE-less booting, provides host provisioning when PXE services are not available. This guide covers PXE-less host provisioning.

You can use the Discovery image ISO to boot either bare metal hardware or virtual systems and attach them to the Satellite Server. Both DHCP and static IP networking are supported. The Discovery Plug-ins are installed by default and are a part of the subscribed repository.

1. Verify that your Satellite Server has a compatible version of the discovery plug-ins.

   - Discovery plug-in - tfm-rubygem-foreman_discovery-5.0.0.4-1 or later

   - Discovery image - foreman-discovery-image-3.4.4-5 or later

   - Smart proxy plug-in - rubygem-smart_proxy_discovery-1.0.3-2 or later
     If you need to upgrade the plug-ins, you must reboot.

2. For physical systems, transfer the ISO image onto a USB stick or CDROM/DVD. For virtual systems, attach the ISO image to a virtual CD-ROM.

3. Boot the ISO image on your system.
   A text-based user interface appears.

4. Select either **Manual network setup** or **Discover with DHCP**, depending on your environment.

5. If the host has multiple NICs, choose the primary network interface and click **Select**.

6. If applicable, enter your network credentials and click **Next**.

7. Enter the full Satellite Server URL and select the **Server** connection type. Click **Next**.

8. Enter custom facts to trigger auto-provisioning rules and click **Confirm**.
   The host is discovered and sends a "discovery_kexec" fact to Satellite Server that prevents a reboot during provisioning.

9. In the web UI, go to **Hosts → Discovered Hosts** to view the host.

10. Click **Provision**, enter the information for your host and click **Submit**.