Red Hat Satellite 6.0

Installation Guide

Installing and Configuring Satellite

Last Updated: 2017-09-19

Red Hat Satellite 6.0 Installation Guide

Installing and Configuring Satellite

Red Hat Satellite Documentation Team

Legal Notice

Abstract

This document describes how to install Satellite. It also steps through the basic configuration requirements to get Satellite running in your environment.

Table of Contents

CHAPTER 1. INTRODUCTION

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

Report a bug

1.1. RED HAT SATELLITE 6 SYSTEM ARCHITECTURE

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

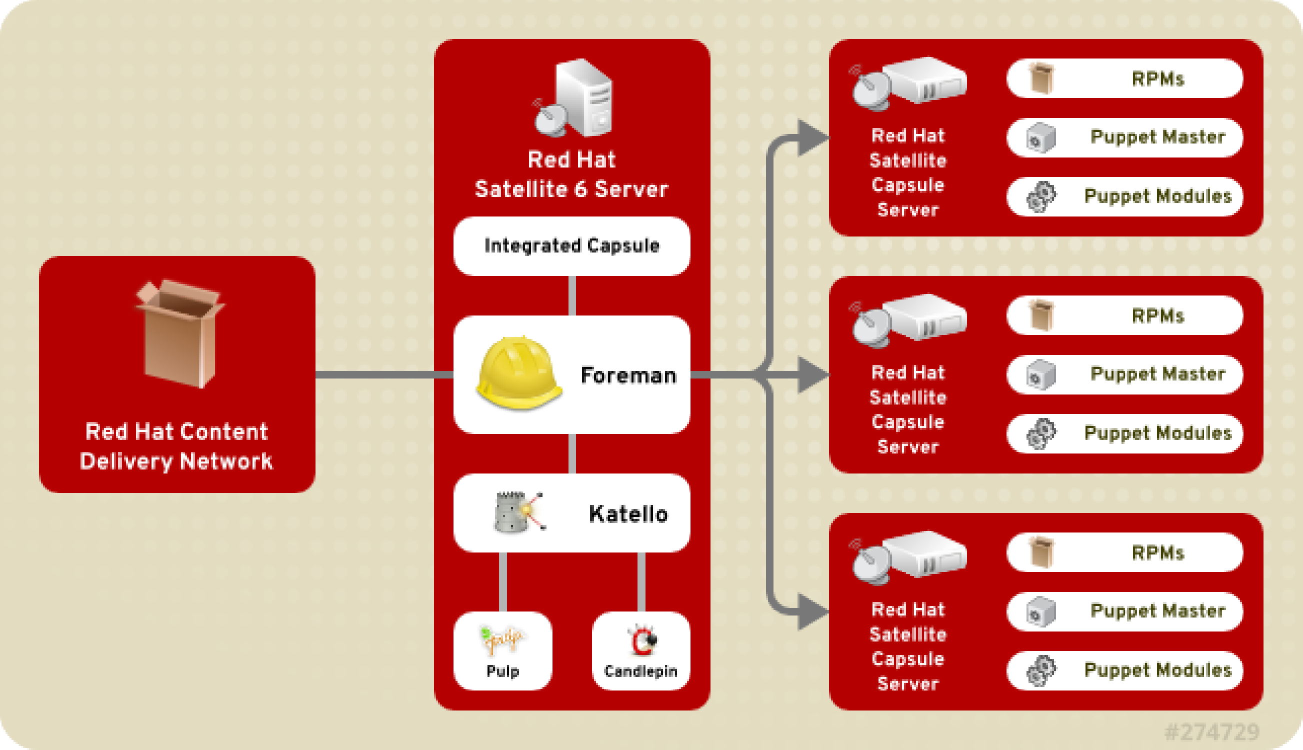

Red Hat Content Delivery Network
Red Hat Satellite 6 Server
Integrated Capsule
Foreman
Katello
Pulp
Candlepin
Red Hat Satellite Capsule Server
RPMs
Puppet Master
Puppet Modules
Red Hat Satellite Capsule Server
RPMs
Puppet Master
Puppet Modules
Red Hat Satellite Capsule Server
RPMs
Puppet Master
Puppet Modules
#274729

Figure 1.1. Red Hat Satellite 6 System Architecture

Foreman

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

Katello

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

Candlepin

Candlepin is a service within Katello that handles subscription management.

Pulp

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

Hammer

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

REST API

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

Capsule

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

Report a bug

1.2. RED HAT SATELLITE 6 LAYOUT AND TOPOLOGY

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

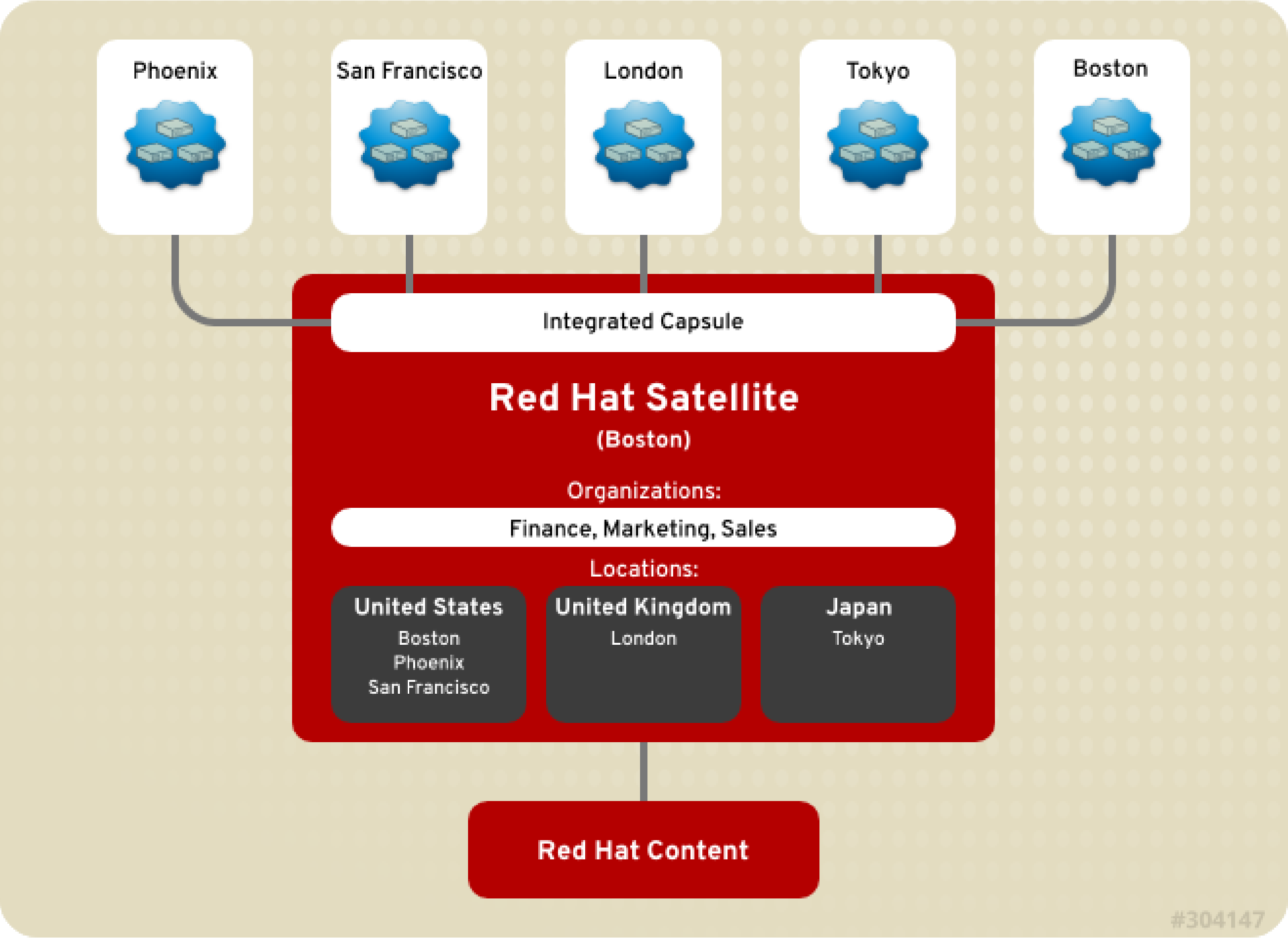


Figure 1.2. Single Satellite with Integrated Capsule

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

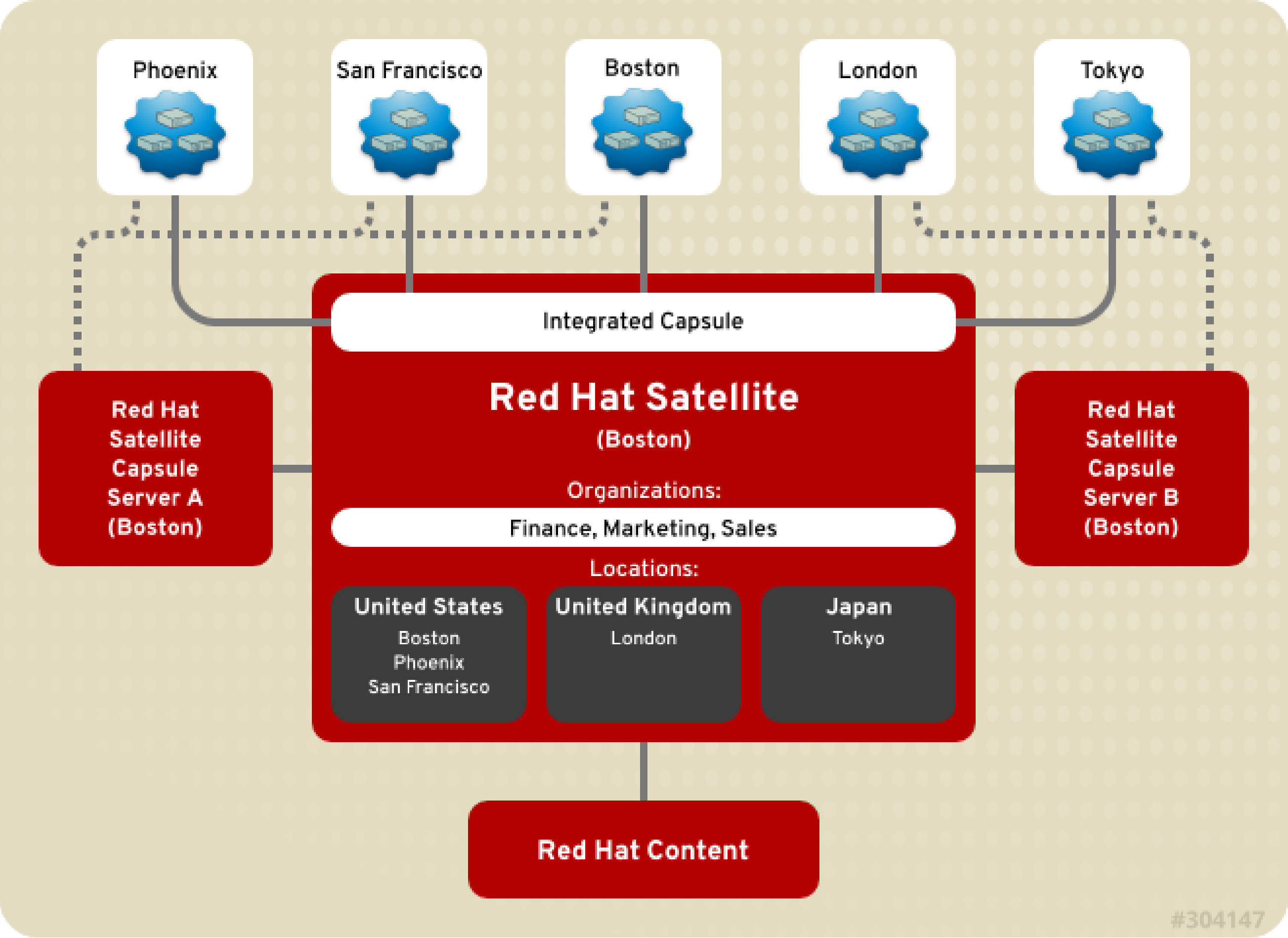


Figure 1.3. Single Satellite with Integrated Capsule and Backup Capsules

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

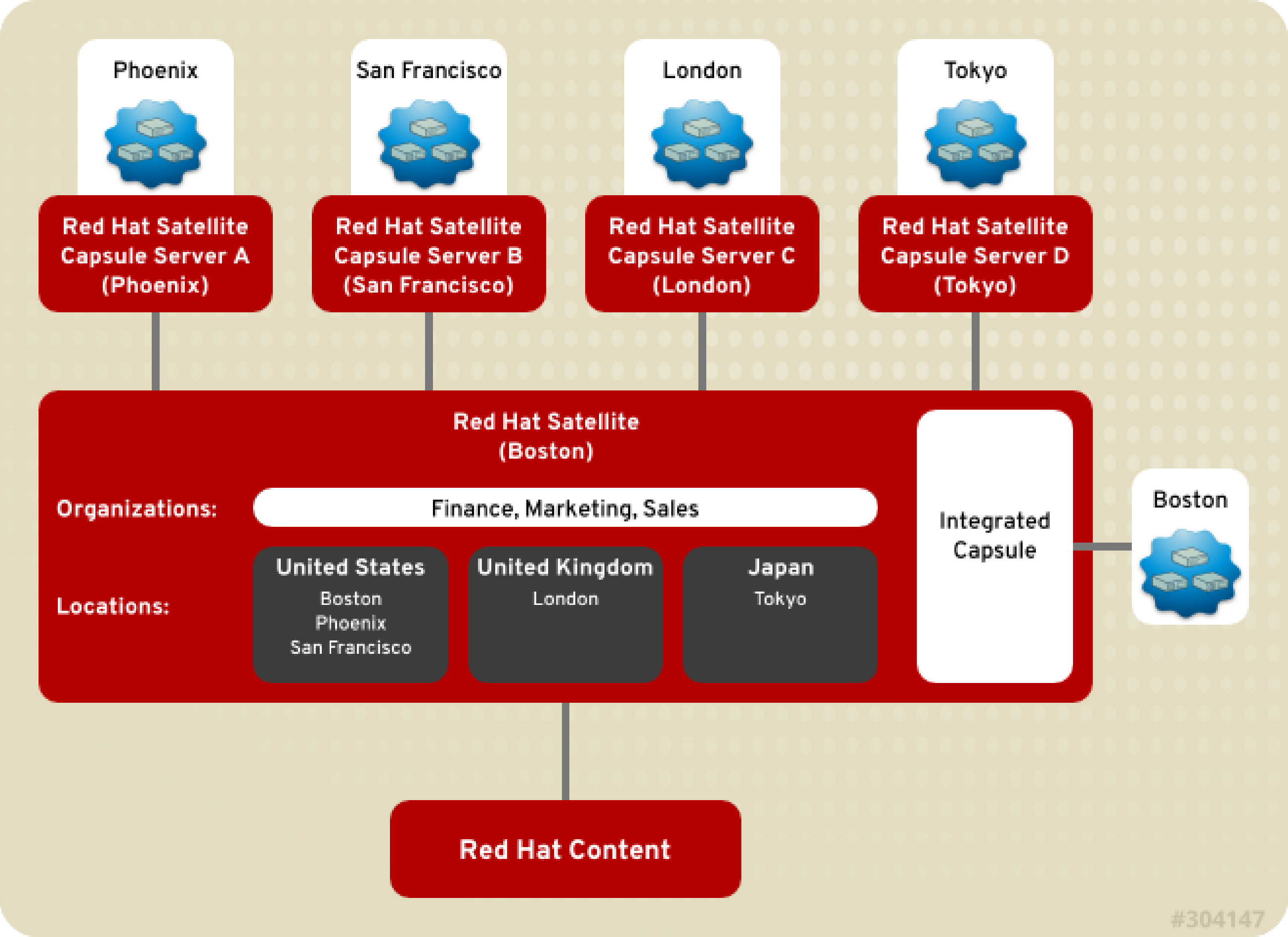

Phoenix
San Francisco
London
Tokyo
Red Hat Satellite Capsule Server A (Phoenix)
Red Hat Satellite Capsule Server B (San Francisco)
Red Hat Satellite Capsule Server C (London)
Red Hat Satellite Capsule Server D (Tokyo)
Red Hat Satellite (Boston)
Organizations:
Finance, Marketing, Sales
Locations:
United States
Boston
Phoenix
San Francisco
United Kingdom
London
Japan
Tokyo
Integrated Capsule
Boston
Red Hat Content
#304147

Figure 1.4. Remote Capsules Based on Location

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

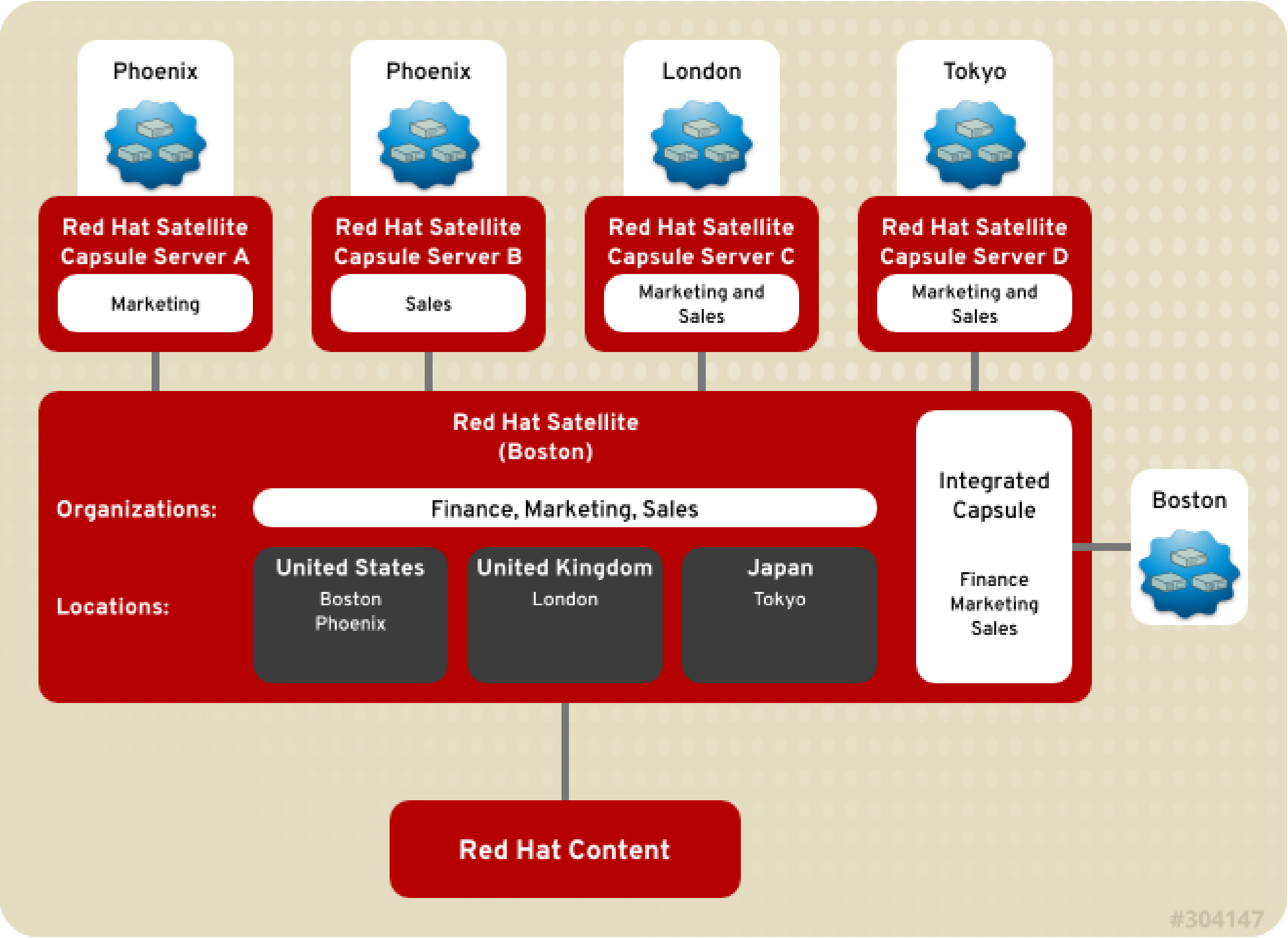


Figure 1.5. Remote Capsules Based on Location and Organization

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

Report a bug

1.3. RED HAT SATELLITE 6 SERVER INSTALLATION WORKFLOW

Installation

This installation workflow covers preparing the host system for the installation and the installation of the Red Hat Satellite Server, including setting up basic infrastructure to provision a host machine.

1. Install the host system with Red Hat Enterprise Linux 6.5 or later.

2. Prepare the host system for the Satellite Server installation by configuring network ports and SELinux settings, as well as checking for correct package versions. See Section 1.5, “Prerequisites” for more information.

3. Install the Satellite Server. There are two methods available:

    1. Section 2.1, “Installing Red Hat Satellite”

    2. Section 2.2, “Installing Red Hat Satellite with an ISO Image”

4. Configure the initial setup of the Satellite Server using `katello-installer`. See Section 2.3, “Configuring Red Hat Satellite Manually” for more information.

Initial Configuration

These are the initial configuration procedures to provide a basic Satellite Server and populating it with content from Red Hat subscriptions:

1. Create a manifest from the Red Hat Customer Portal. See Section 4.1.1.1, “Setting up a Manifest” for more information.

2. Log in to the Satellite Server. This requires the administration user and password. See Chapter 3, *Logging in to Red Hat Satellite* for more information.

3. Create an Organization. Satellite is pre-populated with a "Default Organization" and this organization can be used if needed. It is, however, highly recommended to create an organization that matches the ones in your own organizational structure. See Section 3.1.1, “Creating an Organization” for more information.

4. Remove the "Default Organization". See Section 3.1.3, “Removing an Organization” for more information.

5. Set your default Organization and Location context so that subsequent logins will place you in the right context within the Red Hat Satellite Server. See Section 3.2, “Changing Your Account Preferences” for more information.

6. Logout and re-login. Your default organization and location preferences should now be visible on the upper left corner of the main menu.

7. Upload the manifest in the Satellite Server web interface. This will propagate the subscription information into the Satellite Server. See Section 4.1.1.2, “Uploading a Subscription Manifest” for more information.

8. Once the manifest has been uploaded, the Red Hat Repositories available from valid Red Hat Subscriptions are imported into the Satellite Server. Choose which repositories are relevant to your organization. See Section 4.1.1.3, “Enabling Red Hat Repositories” for more information.

   A common basic set of subscriptions which contain repositories with the latest packages for Red Hat Enterprise Linux 6 would be:

   1. Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server Repository

   2. Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server Repository

   3. Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 Repository

9. Manually synchronize the repositories. This is a required step after enabling them. It is at this point that the local repository in the Satellite Server is populated by the required packages. See Section 4.1.3.1, “Synchronization Status” for more information.

10. With content synchronized locally to the Red Hat Satellite Server, set up lifecycle environments and environment paths according to your organization's requirements. For example, creating a Test or Development environment to simulate the lifecycle environment currently in your organization. See *Lifecycle Environments* from the *Red Hat Satellite 6 User Guide* to create environments and environment paths.

11. Create a content view to add to the lifecycle environment. For more information on Content Views, see *Using Content Views*, as well as *Creating a Content View* from the *Red Hat Satellite 6 User Guide*.

12. After creating the content view, add repositories and puppet modules to the content view based on your organization's needs. *Adding Repositories to the Content View* from the *Red Hat Satellite 6 User Guide* will guide you through the steps.

13. Publish the content view to use it. See *Publishing Content View* from the *Red Hat Satellite 6 User Guide* for instructions. After the content view has been published, it can be promoted through the promotion paths in the lifecycle environments.

14. Now that the content and promotional paths have been set up, create an activation key to prepare for system provisioning. An activation key is used during the provisioning process to ensure that the system is associated with the correct Host Group. It also ensures that the system consumes the correct subscription. See the *Creating an Activation Key* section of the *Red Hat Satellite 6 User Guide* for more information.

15. Now that you have completed your basic setup, the Red Hat Satellite 6 Server is ready to manage content on existing hosts. Register an existing system in your organization to your Satellite Server using the created activation key. See *Registering a Host* in the *Red Hat Satellite 6 User Guide* for more details.

Red Hat Satellite Capsule Server Installation

The Red Hat Satellite Capsule Server is a Red Hat Satellite component that provides federated services to discover, provision, and configure hosts outside of the primary Satellite Server. Follow these steps to install and configure Satellite Capsule Server:

1. Install the host system with Red Hat Enterprise Linux 6.5 or Later.

2. Prepare the system for the Red Hat Satellite Capsule Server installation by fulfilling the requirements outlined in Section 5.2, “Red Hat Satellite Capsule Server Prerequisites”. This includes configuring your network ports, SELinux settings, etc.

3. Install the Red Hat Satellite Capsule Server. See Section 5.3, “Installing a Red Hat Satellite Capsule Server” for more information.

4. Configure the initial setup of the Satellite Capsule Server. See Section 5.4, “Configuring a Red Hat Satellite Capsule Server” for more information.

5. For Satellite Capsule Servers with content node functionality, the procedure outlined in Section 5.5, “Adding Lifecycle Environments to a Red Hat Satellite Capsule Server” is required to have been completed in order to synchronize content from the central Satellite Server to the Satellite Capsule Server.

6. Your Satellite Capsule Server is now installed, configured, and ready for use. Register hosts in the Satellite Capsule Server's location proximity. See Section 5.7, “Registering Host Systems to a Red Hat Satellite Capsule Server” for more information.

Report a bug

1.4. RED HAT SATELLITE 6 SUPPORTED USAGE

Each purchased Red Hat Satellite product includes one supported instance of Red Hat Enterprise Linux Server. Install Red Hat Satellite on a fresh installation of Enterprise Linux where Red Hat Satellite is the only application and service provided by the Operating System (OS). Using the Red Hat Enterprise Linux

OS included with Satellite to run other daemons, applications, or services within your environment is not supported.

NOTE

All Red Hat Satellite components and their usage are supported within the context of Red Hat Satellite only. Third party usage of any components falls beyond supported usage.

Red Hat Satellite components' supported use are as follows:

Puppet

Red Hat Satellite 6 includes supported puppet packages. The installation program allows users to install and configure Puppet Masters as a part of Red Hat Satellite Capsule Servers. The server installs hiera, which can be used to refine how puppet modules are applied. A puppet module, running on a puppet master on the Red Hat Satellite Server or Satellite Capsule Server, using hiera, is supported by Red Hat.

Red Hat supports many different scripting and other frameworks, including puppet modules. The support of these frameworks are supported based on this article "How does Red Hat support scripting frameworks?"

Installation and configuration of Puppet is left up to the Satellite administrator's discretion.

Pulp

Pulp is the content management subsystem within Red Hat Satellite 6. Pulp usage is only supported via the Satellite Server WebUI, CLI and API. Direct modification or interaction with Pulp's local API or database is not within Red Hat's scope of support.

Red Hat does not support direct modification with Pulp as this can cause irreparable damage to the Red Hat Satellite 6 databases.

Foreman

Foreman is an upstream component that makes up a large amount of Red Hat Satellite's core functionality including the WebUI container, users, organizations, security and other significant functions. Foreman can be extended using plug-ins. However, only Red Hat Satellite packaged plug-ins are supported. Red Hat does not support plug-ins in the Red Hat Satellite Optional channel.

Red Hat Satellite also includes components, configuration and functionality to provision and configure operating systems other than Red Hat Enterprise Linux. While these features are included and can be employed, Red Hat only provides support for their application towards Red Hat Enterprise Linux.

Katello

Katello is the upstream component for Red Hat Satellite 6's Content and Subscription user interface and API. The only supported method of using Katello is through the Red Hat Satellite 6 WebUI, API and database. Red Hat does not support direct interactions and modifications to Katello.

Candlepin

Candlepin is the subscription management subsystem within Red Hat Satellite 6. The only supported method of using Candlepin is through the Red Hat Satellite 6 WebUI, CLI and API only.

Red Hat does not support direct modification and interactions with Candlepin, its local API or database, as this can cause irreparable damage to the Red Hat Satellite 6 databases.

Embedded Tomcat Application Server

The only supported method of using the embedded tomcat application server is through the Red Hat Satellite 6 WebUI, API and database. Red Hat does not support direct interactions and modifications of the embedded tomcat application server's local API or database.

Report a bug

1.5. PREREQUISITES

The following conditions must be met before installing Red Hat Satellite 6:

Base Operating System

Red Hat Satellite is only supported on either:

- Red Hat Enterprise Linux 6 Server
- Red Hat Enterprise Linux 7 Server

Install the operating system from disc, local ISO image, kickstart, or any other method that Red Hat supports and update the system to the latest set of packages in Red Hat Enterprise Linux.

IMPORTANT

- Red Hat Satellite Server requires Red Hat Enterprise Linux installations with the @Base package group with no other package-set modifications, and without third-party configurations or software that is not directly necessary for the direct operation of the server. This restriction includes hardening or other non-Red Hat security software. If such software is required in your infrastructure, install and verify a complete working Satellite Server first, then create a backup of the system before adding any non-Red Hat software.
- Your subscription-manager 'Release' field must be set to 6Server or 7Server in order to receive the latest version of Red Hat Enterprise Linux and Red Hat Satellite during the installation. Set the field by using the command:

  ```
  subscription-manager release --set=Release
  ```

  Only release versions 6Server and 7Server are supported by Red Hat Satellite.

- There should be at least one networked host with the following minimum specifications:
  - 64-bit architecture
  - Red Hat Enterprise Linux 6.5 or later
  - A minimum of two CPU cores, but four CPU cores are recommended.
  - A minimum of 8 GB memory but ideally 12 GB of memory for each instance of Satellite. Use 4 GB of swap space where possible.
  - No Java virtual machine installed on the system, remove any if they exist.
  - No Puppet RPM files installed on the system.

  - No third-party unsupported yum repositories enabled. Third-party repositories may offer conflicting or unsupported package versions that may cause installation or configuration errors.

- A current Red Hat Network subscription.

- Administrative user (**`root`**) access.

- Full forward and reverse DNS resolution using a fully qualified domain name. Ensure that ***`hostname`*** and ***`localhost`*** resolve correctly, using the following commands:

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

IMPORTANT

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

Supported Browsers

Browser support is divided into 4 levels:

1. Level 1: Fully supported preferred browsers for ideal experience.

2. Level 2: Mostly supported. The interface functions but some design elements may not align correctly, UI controls and layout may be misaligned and there maybe degraded performance experienced.

3. Level 3: Design elements may not align correctly.

4. Level 4: Unsupported

The table below outlines the supported browsers and their level of support:

Table 1.1. Supported Browser Matrix

Browser	Version	Support Level
Firefox	3.6	L3
Firefox	17, 18, 19, 20	L4
Firefox	21	L2
Firefox	22, 23, 24	L1
Firefox	Latest	L1
Chrome	19, 20	L4

Browser	Version	Support Level
Chrome	21, 27	L2
Chrome	Latest	L1
Internet Explorer	7, 8	L4
Internet Explorer	9, 10, 11	L2
Safari	ALL	L4

NOTE

The web-based user interface and command-line interface for Satellite Server supports English, Portuguese, Simplified Chinese, Traditional Chinese, Korean, Japanese, Italian, Spanish, Russian, French, and German.

Storage

Satellite Server storage specifications are as follows:

- A minimum of 6 GB storage for base operating system installation of Red Hat Enterprise Linux
- A minimum of 400 MB storage for the Red Hat Satellite 6 software installation
- A minimum of 20 GB storage for each unique software repository. Packages that are duplicated in different channels are only stored once on the disk. Additional repositories containing duplicate packages will require less additional storage. The bulk of storage resides on the **/var/lib/mongodb** and **/var/lib/pulp** directories. These end points are not manually configurable. Make sure that storage is available on the /var file system to prevent storage issues.
- A minimum of 2 GB of available storage in **/var/lib/pgsql** with the ability to grow the partition containing this directory as data storage requirements grow.

NOTE

Most Satellite Server data is stored within the /var directory. It is strongly recommended to mount /var on LVM storage that the system can scale to meet data storage requirements.

Application Specifications

Satellite Server application installation specifications are as follows:

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

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

In Red Hat Enterprise Linux 7 **chrony** is the default time synchronizer. Run the following command to start the **`chronyd`** service and have it persist across restarts:

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

Required Network Ports

The following network ports need to be open and free before continuing with the installation:

Table 1.2. Required Network Ports

Port	Protocol	Service
443	tcp	HTTPS
5671	tcp	SSL
80	tcp	HTTP
8080	tcp	Tomcat6
8140	tcp	Puppet
9090	tcp	Foreman Smart Proxy

You can use either the **`iptables`** command to configure the firewall, or the **`system-config-firewall-tui`** command-line utility. To modify the firewall with the **`iptables`** command:

- Port 443 for HTTPS (secure WWW) must be open for incoming connections.
- Port 5671 must be open for SSL communication with managed systems.
- Port 80 for HTTP (WWW) must be open to download the bootstrap files.
- Port 8080 for TCP must be free for java connections.
- Port 8140 must be open for incoming Puppet connections with the managed systems.
- Port 9090 must be open for Foreman Smart Proxy connections with the managed systems.
- Run the following commands to configure the firewall with the **`iptables`** command and to make these rules persistent during reboots:

  NOTE

  Run the firewall commands as the root user.

  1. For Red Hat Enterprise Linux 6:

     ```
     iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
     ACCEPT \
     && iptables -I INPUT -m state --state NEW -p tcp --dport 5671 -j
     ```

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

2. For Red Hat Enterprise Linux 7:

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

Advanced Firewall Considerations

The following tables provide details about the network connections made to and from the Red Hat Satellite Server.

Table 1.3. Advanced Firewall Considerations with a Red Hat Satellite Server

Flow	Action	Initiator	Initiator Detail	Endpoint	Endpoint Detail	Port	Protocol	SELinux Type
Capsule Registration	Capsule Registration (Content Host)	Capsule	subscription-manager	Satellite Server	Apache	443	https	http_port_t
	Capsule Registration (Smart Proxy)	Capsule	Smart Proxy	Satellite Server	Passenger	8443/9090	https	http_port_t, websm_port_t
	Broker Connections	Capsule	AMQP Broker	Satellite Server	AMQP Broker	5671	amqp	amqp_port_t
Content Promotion	Initiation	Server	AMQP Broker	Capsule	AMQP Broker	5671	amqp	amqp_port_t
	Content Mirroring	Capsule	Pulp Node	Server	Apache	443	https/http	http_port_t
Puppet Run	Manifest Request	Client	Puppet Agent	Capsule	Puppet Master	8140	https	puppet_port_t

Flow	Action	Initiator	Initiator Detail	Endpoint	Endpoint Detail	Port	Protocol	SELinux Type
	External Node Classification	Capsule	Puppet Master	Server	Passenger	443	https	http_port_t
	Run Report	Capsule	Puppet Master	Server	Passenger	443	https	http_port_t
Provisioning	IP/TFTP/Etc	Server	Passenger	Capsule	Smart Proxy	8140/8443/9090	https	puppet_port_t, http_port_t, websm_port_t
	Installation	Client	Anaconda/Yum	Capsule	Pulp Node	8443/80	https/http	http_port_t
	Installation Templates	Client	Anaconda	Server	Passenger	443/80	https/http	http_port_t
Orchestration	DNS records	Capsule	nsupdate	DNS server	bind	53	dns	dns_port_t
	DHCP records	Capsule	omapi	DHCP server	ISC dhcp	7911	omapi	dhcp_port_t
Yum Run	Yum Execution	Client	Yum	Capsule	Apache	8443/80	https/http	http_port_t
Power Management	BMC On/Off/Cycle/Status	Capsule	User Initiated API/UI	Client	IPMI Device	n/a	ipmi	n/a

Table 1.4. Advanced Firewall Considerations without a Red Hat Satellite Capsule Server

Flow	Action	Initiator	Initiator Detail	Endpoint	Endpoint Detail	Port	Protocol	SELinux Type

Flow	Action	Initiator	Initiator Detail	Endpoint	Endpoint Detail	Port	Protocol	SELinux Type
Subscription Management	Subscription-Manager - Execution	Client	subscription-manager	Satellite Server	Passenger	443	https	http_port_t
Action Initiation	Server	AMQP Broker	Client	katello-agent	5671	amqp	amqp_port_t	
	Installation/Removal/Update	Client	Anaconda/Yum	Capsule	Apache	8443/80	https/http	http_port_t
Provisioning	Installation Templates	Client	Anaconda	Server	Passenger	443/80	https/http	http_port_t
Compute Resources	Fog interactions (query/create destroy/etc)	Server	Fog	Compute Resource	Openstack	5000	http	commplex_main_port_t
					EC2	443	https	http_port_t
					oVirt	443	https	https_port_t
					Libvirt	22/16514	ssh/tls	ssh_port_t, virt_port_t

SELinux Policy

Red Hat recommends that the SELinux policy on Satellite 6 systems be set to enforcing.

IMPORTANT

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

Troubleshooting

Red Hat recommends that the sos package is installed on Satellite. The sos packages provides the `sosreport` command. The `sosreport` command is a tool that collects configuration and diagnostic information from a Red Hat Enterprise Linux system and is used to provide the initial analysis of a system required when opening a service request with Red Hat Technical Support.

To install the sos package run the following command:

```
# yum install sos
```

Report a bug

CHAPTER 2. RED HAT SATELLITE INSTALLATION

2.1. INSTALLING RED HAT SATELLITE

This section describes how to use Subscription Manager to install Red Hat Satellite Server from the repository.

Procedure 2.1. To Install Satellite Server on a Certificate-managed System:

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

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

   This command displays output similar to the following:

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

   Subscription Name: Red Hat Satellite Subscription
   Provides:          Red Hat
                      Red Hat Satellite Capsule 6
                      Red Hat Enterprise Linux 7
                      Red Hat Satellite 6
   SKU:               SKU123456
   Pool ID:           e1730d1f4eaa448397bfd30c8c7f3d334bd8b
   Available:         6
   Suggested:         1
   Service Level:     Self-Support
   Service Type:      L1-L3
   Multi-Entitlement: No
   Ends:              01/01/2022
   System Type:       Physical
   ```

   NOTE

   The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type. Take note of the pool IDs of Red Hat Satellite 6, Red Hat Enterprise Linux and Red Hat Software collections that correspond to your system version and product type.

2. Attach a subscription to the registered system:

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

3. Disable all existing repositories:

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

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

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

   NOTE

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

5. Install the katello package:

   ```
   # yum install katello
   ```

Result:

Satellite Server is installed on your host system.

IMPORTANT

You need to configure your Satellite Server before use. See either Section 2.3, “Configuring Red Hat Satellite Manually” or Section 2.4, “Configuring Red Hat Satellite with an Answer File” for more information.

Report a bug

2.2. INSTALLING RED HAT SATELLITE WITH AN ISO IMAGE

Prerequisite

ISO installations require imported Red Hat GPG keys before installation. Run the following command as root prior to running the installation script:

```
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

The following procedure details how to install Satellite Server on a host through ISO.

1. Download the ISO image from the Red Hat Satellite 6 Channel in the Red Hat Customer Portal.

2. As the root user, mount the ISO image to a directory:

   ```
   # mkdir /media/iso
   # mount -o loop [iso_filename] /media/iso
   ```

3. Change directory to **`/media/iso`**.

4. Run the installer script in the mounted directory:

```
# ./install_packages
```

Result:

Satellite Server is installed on your host system.

IMPORTANT

You need to configure your Satellite Server before use. See either Section 2.3, “Configuring Red Hat Satellite Manually” or Section 2.4, “Configuring Red Hat Satellite with an Answer File” for more information.

Report a bug

2.3. CONFIGURING RED HAT SATELLITE MANUALLY

Satellite Server has an automatic initial configuration that prepares the Satellite Server for use. The **katello-installer** script supports the ability to override various default settings within the different components of Satellite Server. For example, for organizations that have an existing HTTP proxy, additional configuration options need to be passed to the Satellite Server installer. See Section 2.3.1, “Configuring Red Hat Satellite Manually with an HTTP Proxy” for HTTP proxy options and Section 2.3.2, “Configuring Red Hat Satellite with a Custom Server Certificate” for custom Certificate Authority (CA) certificates.

Procedure 2.2. Running the Installer Script

1. As the root user, configure Red Hat Satellite Server manually by running the **katello-installer** script:

   ```
   katello-installer --foreman-admin-username desired_username --
   foreman-admin-password desired_password
   ```

   Where:

   1. desired_username is the username that will replace the default "admin" user.

   2. desired_password is the password that will replace the default password generated.

   Additional configuration options can be passed to the **katello-installer** command to adjust various aspects of the configuration. View a complete list of options using the command:

   ```
   # katello-installer --help
   ```

   This script can be run multiple times without any issues.

   IMPORTANT

   The default username is ***admin***. The default organization name is ***Default Organization***. It is strongly recommended that you override these default settings once you log in to the Satellite Server interface and navigate to **Administer** → **Users** and **Administer** → **Organizations**.

When the configuration script has completed successfully, it displays:

```
# katello-installer
Installing             Done
   [100%] [.......................................]
   Success!
   * Katello is running at https://sat6.example.com
       Default credentials are 'admin:changeme'
   * Capsule is running at
https://sat6.example.com: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
```

2. After configuration, run the following commands to configure the firewall to limit **elasticsearch** to the **`foreman`**, **`katello`** and root users and make these rules persistent during reboots:

    - For Red Hat Enterprise Linux 6:

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

    - For Red Hat Enterprise Linux 7:

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

```
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP
```

The Red Hat Satellite Server is initially populated with an initial organization and location named "Default Organization" and "Default Location", respectively. These organizations can be used within the Satellite Server as they are. However, additional organizations and locations can be created in the Satellite Server after initial configuration. The default location and organization can be deleted after new organizations and locations have been created.

Report a bug

2.3.1. Configuring Red Hat Satellite Manually with an HTTP Proxy

For networks that go through an HTTP Proxy, the following katello-installer options should be used in order for the Satellite Server to successfully complete the configuration:

```
katello-installer --katello-proxy-url=http://myproxy.example.com --
katello-proxy-port=8080 --katello-proxy-username=proxy_username --katello-
proxy-password=proxy_password
```

Where:

- **`--katello-proxy-url`** - the URL of the HTTP proxy server.
- **`--katello-proxy-port`** - the port the HTTP proxy server is listening on.
- **`--katello-proxy-username`** - (optional) the HTTP proxy username for authentication. If your HTTP proxy server does not require a username, you are not required to specify the username.
- **`--katello-proxy-password`** - (optional) the HTTP proxy password for authentication. If your HTTP proxy server does not require a password, you are not required to specify the password.

After configuring the Satellite Server to go through the HTTP Proxy, make sure that **yum** or **subscription-manager** can connect to the Red Hat Content Delivery Network (CDN) and that the Satellite Server can synchronize its repositories to the CDN by following these steps:

Procedure 2.3. Configuring the Satellite Server to Allow Red Hat Subscription Manager Access to the CDN

1. On the network gateway and the HTTP Proxy, open the following hostnames, ports and protocols:

   Table 2.1. Required Hostnames, Ports and Protocols

Hostname	Port	Protocol
subscription.rhn.redhat.com	443	https
cdn.redhat.com	443	https
*.akamaiedge.net	443	https

2. On the Satellite Server, fill in the following details in the **`/etc/rhsm/rhsm.conf`** file:

```
# an http proxy server to use (enter server FQDN)
proxy_hostname = http_proxy.example.com

# port for http proxy server
proxy_port = 3128

# user name for authenticating to an http proxy, if needed
proxy_user =

# password for basic http proxy auth, if needed
proxy_password =
```

Report a bug

2.3.2. Configuring Red Hat Satellite with a Custom Server Certificate

`katello-installer` comes with a default CA used both for the server ssl certificates as well as the client certificates used for authentication of the subservices. These certificates can be replaced with custom ones.

There are two instances wherein you can configure the Satellite Server to use a custom CA certificate:

1. When **`katello-installer`** is run the first time

2. After **`katello-installer`** has already been run

Procedure 2.4. Setting a Custom Server Certificate while running katello-installer for the first time

- Run this command on the Red Hat Satellite Server:

  ```
  katello-installer --certs-server-cert ~/path/to/server.crt\
                    --certs-server-cert-req ~/path/to/server.crt.req\
                    --certs-server-key ~/path/to/server.crt.key\
                    --certs-server-ca-cert ~/path/to/cacert.crt
  ```

  Where:

  - **`certs-server-cert`** is the path to your certificate, signed by your certificate authority (or self signed)

  - **`certs-server-cert-req`** is the path to your certificate signing request file that was used to create the certificate.

  - **`certs-server-key`** the private key used to sign the certificate

  - **`certs-server-ca-cert`** ~/path/to/cacert.crt the path to the CA certificate on this system.

Procedure 2.5. Setting a Custom Server Certificate after running katello-installer

1. The initial run of **katello-installer** uses the default CA for both server and client certificates. To enforce custom certificates deployment, set the ***--certs-update-server*** parameter and the ***--certs-update-server-ca*** parameter to update the CA certificate:

   ```
   katello-installer --certs-server-cert ~/path/to/server.crt\ --certs-
   server-cert-req ~/path/to/server.crt.req\ --certs-server-key
   ~/path/to/server.crt.key\ --certs-server-ca-cert
   ~/path/to/cacert.crt\ --certs-update-server --certs-update-server-ca
   ```

   This will regenerate the katello-ca-consumer package and the server CA certificate.

2. After the server CA changes, install the new version of the consumer-ca-consumer package on the client systems:

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

IMPORTANT

Use the same custom server certificate on both the Red Hat Satellite Server and the Red Hat Satellite Capsule Server to ensure that the trusted relationship between the two hosts is maintained.

Report a bug

2.4. CONFIGURING RED HAT SATELLITE WITH AN ANSWER FILE

The following task shows how to configure Red Hat Satellite Server with an answer file.

Procedure 2.6. Configuration Using an Answer File

Answer files are used for automated installations with customized options. To configure and use an answer file for installation:

1. Copy the default answer file located at **/etc/katello-installer/answers.katello-installer.yaml** to a location on your local filesystem:

   ```
   # cp /etc/katello-installer/answers.katello-installer.yaml
   /etc/katello-installer/my-answer-file.yaml
   ```

   NOTE

   The initial answer file will be sparsely populated. However, once **katello-installer** is run for the first time, the answer file is populated with the standard parameter values for installation.

2. Open your copy of the answer file in your preferred text editor and edit the values to suit your environment. Save your answer file once you have finished editing it.

> **NOTE**
>
> The parameters for each module are specified in the module's **`params.pp`** file. Available modules with parameter files can be viewed by executing the command:
>
> ```
> # rpm -ql katello-installer | grep params.pp
> ```

3. Open the **`/etc/katello-installer/katello-installer.yaml`** file in your preferred text editor and edit the answer file entry to point to your custom answer file:

   ```
   :answer_file: /etc/katello-installer/my-answerfile.yaml
   ```

4. Run the **`katello-installer`** command.

   ```
   # katello-installer
   ```

Result:

Satellite is configured on your host system with your desired configuration.

Report a bug

CHAPTER 3. LOGGING IN TO RED HAT SATELLITE

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

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

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

   `https://`*`HOSTNAME`*`/`

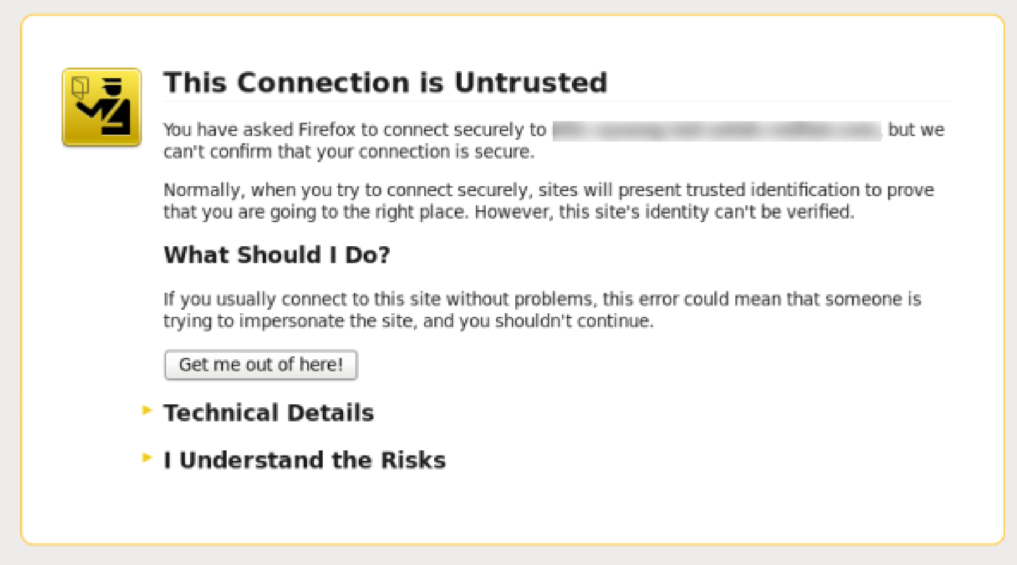
   To identify your hostname, use the **`hostname`** command at the prompt:

   ```
   # hostname
   ```

   ### IMPORTANT

   An untrusted connection warning appears on your web browser when accessing **Satellite** for the first time. Accept the self-signed certificate and add the **Satellite** URL as a security exception to override the settings. This procedure might differ depending on the browser being used.

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

   This Connection is Untrusted

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

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

   What Should I Do?

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

   Figure 3.1. Untrusted Connection Warning

2. Enter the user name and password created during the configuration process. If a user was not created during the configuration process, the default user name is *admin*.

Result

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

The main navigation tabs are as follows:

Table 3.1. Navigation Tabs

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

NOTE

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

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

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

Report a bug

3.1. ORGANIZATIONS

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

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

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

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

IMPORTANT

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

Report a bug

3.1.1. Creating an Organization

These steps show how to create a new organization.

Procedure 3.1. Creating an Organization

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

Result:

A new organization is created.

Report a bug

3.1.2. Editing an Organization

Procedure 3.2. Editing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Click the name of the organization to be edited.
3. Select the resource to edit from the list on the left.

4. Click the name of the desired items to add them to the `Selected Items` list.

5. Click the **`Submit`** button.

Result

The organization is updated and saved.

Report a bug

3.1.3. Removing an Organization

Procedure 3.3. Removing an Organization

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

2. Select **Delete** from the drop down menu to the right of the name of the organization you want to remove.

3. An alert box appears:

   ```
   Delete Organization Name?
   ```

4. Click the **`OK`** button.

Result

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

Report a bug

3.2. CHANGING YOUR ACCOUNT PREFERENCES

Setting up default account preferences ensures that subsequent logins will enable the correct context within the Red Hat Satellite Server for a specific user. It also allows changes in user preferences.

The following preferences can be changed:

1. **User** - Change personal data about your login name, as well as your password and default location/organization.

   1. First Name

   2. Surname

   3. Email Address

   4. Default Location

   5. Default Organization

   6. Password

2. **Locations** - Add or remove locations on your account based on the locations created within the Red Hat Satellite Server.

3. **Organizations** - Add or remove organizations on your user account based on the organizations created within the Red Hat Satellite Server.

4. **Roles** - Add or remove roles on your user account based on a set of roles created within the Red Hat Satellite Server.

Procedure 3.4. Changing your Account Preferences

To change these preferences:

1. At the upper right corner, hover your mouse over the **admin** user and on the drop-down menu that appears, click on **My Account**.

2. Choose the subtab of the preference you wish to change and click on the subtab.

3. Change the preferences you wish to change and click on **`Submit`**.

NOTE

Set your default location/organization in the **User** subtab after your initial login. This will make sure that subsequent logins will set you in the correct context for your user.

Report a bug

3.3. CONFIGURING LDAP AUTHENTICATION FOR RED HAT SATELLITE

Red Hat Satellite includes the option to use a Lightweight Directory Access Protocol (LDAP) service for user information and authentication, using one or more LDAP directories.

Procedure 3.5. To Configure LDAP Authentication:

1. Log in as the Satellite administrator.

2. Click **Administer** → **LDAP Authentication** on the upper right of the page.

3. Click **`New LDAP source`** on the upper right of the page.

4. Fill in the information in the following tabs:

   a. **LDAP Server** Tab. For the LDAP server tab, enter the following information:

      - Name - LDAP server's name.
      - Server - the LDAP server's hostname.
      - Port - the LDAP port. This defaults to port 389.
      - TLS - Enables Transport Layer Security. Tick the check box to enable encryption.

   b. **Account** Tab. For the Account Tab, enter the following information:

      - Account Username - an LDAP user who has read access to the LDAP server. This field cannot remain blank. Use the full path to the user's object. For example:

        -

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

- Account password - the LDAP password for the user defined in the Account username field. This field can remain blank if the Account username is using the "$login" variable.
- Base DN - the top level domain name of your LDAP directory. For example:

  ```
  cn=users,cn=accounts,dc=redhat,dc=com
  ```

- LDAP filter
- Automatically create accounts in Foreman - creates Satellite accounts automatically for LDAP users who log in for the first time in Satellite. Tick the check box to enable this feature.

c. **Attribute mappings** tab. LDAP attributes such as login, name, and email addresses, need to be mapped to Satellite attributes for LDAP to work on Satellite. The following attributes can be mapped to Satellite:

- Login name attribute
- First Name attribute
- Surname attribute
- Email address attribute
- Photo attribute

5. Click **`Submit`**.

Result:

LDAP authentication is now configured on the Satellite.

Report a bug

CHAPTER 4. POPULATING RED HAT SATELLITE WITH CONTENT

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

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

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

Report a bug

4.1. CONNECTED SATELLITE

4.1.1. Using Red Hat Content Providers

4.1.1.1. Setting up a Manifest

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

Prerequisites

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

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

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

1. Log in to the Customer Portal.
2. Click **Subscriptions**. On the Red Hat Subscription Management Page, scroll down and click on Subscription Management Applications, then, click **Satellite**.
3. On the upper right corner of the Subscriptions Management Applications page, click **`Register a Satellite`**.
4. Create a name to distinguish your Satellite from the other Satellite systems in your account.
5. Select **Satellite 6.0** from the drop-down menu as the **`Satellite version`**. It is important to select the correct version as each version requires a certain subset of packages.
6. Click **`Register`**.
7. Click **`Attach a subscription`**, add the subscriptions required for Red Hat Satellite, and then click **Attach Selected**. See How to generate a certificate for more information.

The minimum requirements for generating a manifest are:

- A valid Red Hat Satellite subscription on your Customer Portal account
- At least one Red Hat Enterprise Linux subscription to attach to the manifest

> **WARNING**
>
> If a valid Red Hat Satellite subscription is not in the account, the **`Download Manifest`** will not be clickable.

8. Click **Download manifest** to generate an archive in .zip format that contains the manifest for Red Hat Satellite.

Result:

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

Report a bug

4.1.1.2. Uploading a Subscription Manifest

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

Procedure 4.1. To Upload a Subscription Manifest

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

Report a bug

4.1.1.3. Enabling Red Hat Repositories

The Red Hat manifest file gives access to Red Hat products and repositories, however, since most products have several architectures and product versions, Red Hat Satellite Server allows the Satellite administrators to choose which repositories are required by their organizations. The repositories need to be enabled in the Red Hat Satellite Server to prepare it for synchronization.

These steps show how to enable a Red Hat repository:

Procedure 4.2. Enabling Repositories

1. Click **Content** → **Red Hat Repositories**.
2. Click on the tab of the type of content to be enabled. The tabs are: **`RPMs`**, **`Source RPMs`**, **`Debug RPMs`**, **`Beta`**, **`ISOs`**, **`Other`**.
3. Expand each Red Hat product to examine the different repository sets available by clicking on the arrow by the product name.
4. Choose which Red Hat repository sets you wish to add. Choosing it will automatically enable that repository for your Red Hat Satellite server.

   For example, a common basic set of subscriptions which contain repositories with the latest packages for Red Hat Enterprise Linux 6 would be:

   1. Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server Repository
   2. Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server Repository
   3. Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 Repository

   NOTE

   The Red Hat Common repository must be enabled as it provides client systems registered to the Satellite Server katello-agent and puppet packages for both katello-agent and puppet.

Result:

Repositories have been enabled and are ready to be synchronized.

NOTE

Products in the "Enabling Red Hat Repositories" page are dependent on manifest content. If a required Red Hat repository is not available on this page, consult the manifest or the Red Hat Customer Portal Subscriptions page.

Report a bug

4.1.2. Using Products

4.1.2.1. Creating a Product

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

Procedure 4.3. Creating a Product

1. Click **Content** → **Products**.
2. Click the **`+ New Product`** link.
3. Type in the name of the new product in the **`Name`** field.

4. Type in label for the new product in the **`Label`** field.

5. Select a GPG key from the **`GPG Key`** drop down menu.

6. Select a synchronization plan from the **`Sync Plan`** drop down menu. Alternatively select the **`+ New Sync Plan`** link to create a new synchronization plan.

7. Type in a description of the new product in the **`Description`** field.

8. Click the **`Save`** button to save your new product.

Result:

A new product is created.

Report a bug

4.1.2.2. Adding Repositories to a Product

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

Procedure 4.4. Adding Repositories to a Product

1. Click **Content** → **Products**.

2. Click the product you wish to add a repository to.

3. Click the **`Repositories`** subtab.

4. Click the **`Create Repository`** button.

5. Type in the name of the new repository in the **`Name`** field.

6. Type in a label for the new repository in the **`Label`** field.

7. Select the type of the repository from the **`Type`** drop down menu.

8. Type in the URL of the repository in the **`URL`** field.

9. Choose whether to publish the repository via HTTP by clicking the **`Publish via HTTP`** checkbox.

10. Select a GPG key for the repository from the **`GPG Key`** drop down menu.

11. Click the **`Create`** button to save your new repository.

Result:

A new repository is added to your product.

Report a bug

4.1.2.3. Using Bulk Actions for Products

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

Procedure 4.5. To Perform Tasks on Multiple Products:

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

Updating Synchronization Plans

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

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

Report a bug

4.1.2.4. Using Repository Discovery

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

Procedure 4.6. Using Repository Discovery

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

   ii. Enter the **`Name`** and **`Label`** for the new product and select a **`GPG Key`** from the drop down menu.

9. Select the **`Serve via HTTP`** check box to serve the repository via HTTP.

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

11. Click the **`Create`** button.

Result:

The repositories have been discovered and added to a product.

Report a bug

4.1.2.5. Removing a Product

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

Procedure 4.7. To Remove a Product from Satellite:

1. Click **Content** → **Products**.

2. Select the check box next to the products you want to remove.

3. Click **`Bulk Actions`** and then click **`Remove Products`**.

4. Click **`Remove`** to confirm that you want to remove the products.

Report a bug

4.1.3. Synchronizing Content from Source Repositories to Red Hat Satellite

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

Constant, scheduled synchronization will result in:

- Data integrity between packages

- Updated packages, security fixes, and errata

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

Report a bug

4.1.3.1. Synchronization Status

IMPORTANT

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

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

Procedure 4.8. Synchronize Products

1. Click **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product channels available for synchronization is displayed.

2. Click the arrow next to the product name to see available content.

3. Select the content you wish to synchronize.

4. Click the **`Synchronize Now`** button to starting synchronizing. The status of the synchronization process will appear in the **`Result`** column. If synchronization is successful, **`Sync complete`** will appear in the **`Result`** column. If synchronization failed, **`Error syncing`** will appear.

Result:

A product is synchronized.

NOTE

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

Report a bug

4.1.3.2. Creating a Synchronization Plan

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

Procedure 4.9. To Create a Synchronization Plan:

1. Click **Content** → **Sync Plans**.

2. Click the **`New Sync Plan`** link to create a new synchronization plan.

3. Enter the **`Name`**, **`Description`** and other details for the plan.

4. Click **`Save`** to create the synchronization plan.

Report a bug

4.1.3.3. Applying a Synchronization Schedule

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

Procedure 4.10. To Create a Synchronization Schedule:

1. Click **Content** → **Sync Plans** and select the synchronization plan you want to implement.
2. Click **Products** → **Add** in the synchronization plan main page.
3. Select the check box of the product to associate with the synchronization plan.
4. Click **`Add Selected`**.

Report a bug

4.2. DISCONNECTED SATELLITE

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

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

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

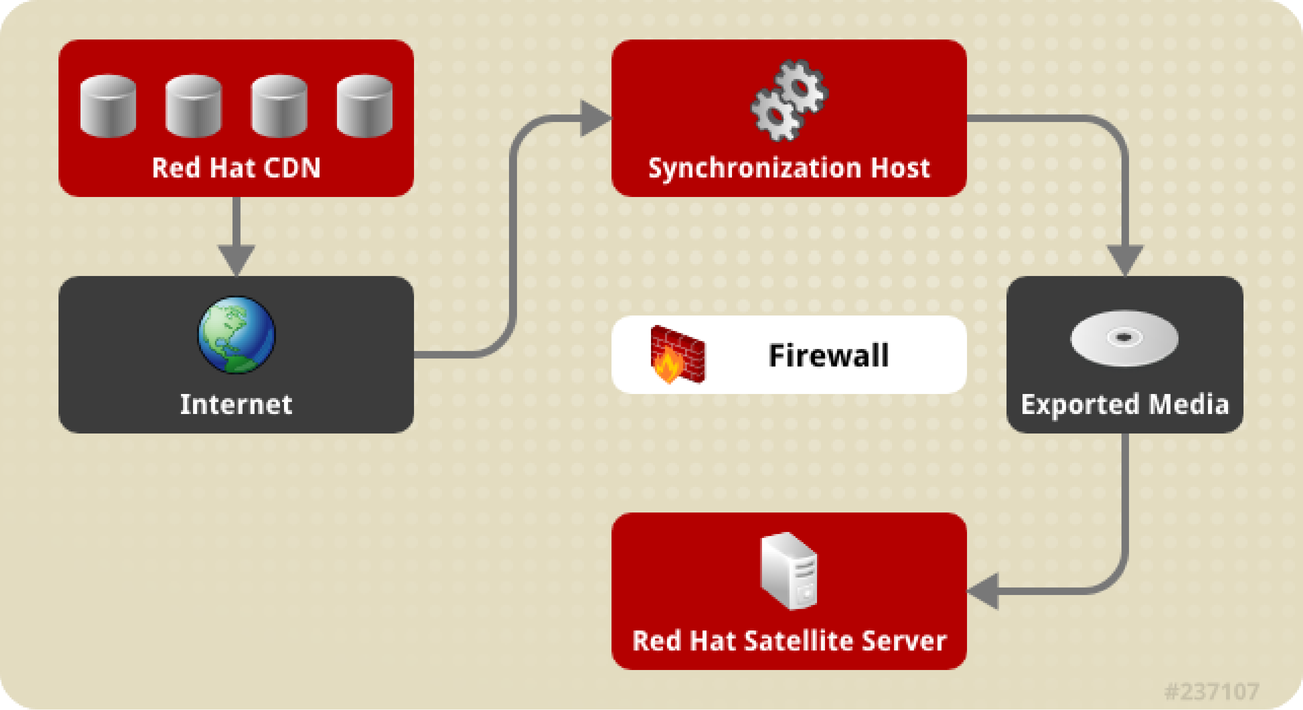


Figure 4.1. Disconnected Satellite

Report a bug

4.2.1. Configuring the Synchronization Host

Prerequisites

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

- An Internet connection
- Valid Red Hat Network subscriptions
- A valid manifest (See Section 4.1.1.1, “Setting up a Manifest” for instructions on how to obtain one.)

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

1. Use Red Hat Subscription Manager to register the synchronization host to RHN.

2. List all the available subscriptions to find the correct Red Hat Satellite product to allocate to your system:

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

   This command displays output similar to the following:

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


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

   NOTE

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

3. Subscribe to the pool using the following command:

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

4. Disable all existing repositories:

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

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

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

NOTE

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

6. Install katello-utils and associated RPM files:

```
# yum install python-qpid-qmf python-qpid  qpid-cpp-server-store
katello-utils
```

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

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

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

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

```
[oauth]
enabled: true
oauth_key: katello
oauth_secret: v8SeYqvS5QUfmg0dIrJOBG58lAHDRZnN
```

9. Disable authentication in **/etc/qpid/qpidd.conf**:

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

auth=no
```

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

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

```
# katello-disconnected setup --oauth-key=katello --oauth-
secret=v8SeYqvS5QUfmg0dIrJOBG58lAHDRZnN
```

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

11. Configure Pulp on the synchronization server:

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

12. Import the manifest to set up the list of available repositories to synchronize based on the selected subscriptions:

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

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

Report a bug

4.2.2. Synchronizing Content

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

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

1. Disable all repositories:

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

katello-disconnected enables all repositories by default.

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

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

```
rhel-5-server-debug-rpms-5_8-ia64
rhel-5-server-debug-rpms-5_8-x86_64
rhel-5-server-debug-rpms-5_9-i386
rhel-5-server-debug-rpms-5_9-ia64
rhel-5-server-debug-rpms-5_9-x86_64
rhel-5-server-isos-5Server-i386
```

3. Enable the chosen repositories for synchronization:

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

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

```
# katello-disconnected configure
```

NOTE

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

5. Synchronize the repositories:

```
# katello-disconnected sync
```

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

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

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


Watching finished
```

Content is now synchronized.

Report a bug

4.2.3. Exporting Content

Prerequisites

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

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

1. Export the synchronized repositories:

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

   The output will look similar to:

   ```

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

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

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

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

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

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

   NOTE

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

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

Report a bug

4.2.4. Importing Content to a Disconnected Satellite Server

Prerequisites

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

1. Copy the exported contents of the external media to a directory on the disconnected Satellite.

NOTE

If the exported content was placed in different external media devices, ensure that you copy all the contents in *one* directory on the disconnected Satellite.

2. Run the **expand_export.sh** script copied along with your exported content:

```
# ./expand_export.sh
*** Done expanding archives. ***
```

WARNING

The disconnected Satellite needs to have equal to the same space as the exported archive before you expand the exported content. The script will fail otherwise.

The script will expand the archive into a directory.

3. Move the expanded directory tree's content to **/var/www/html** on the Satellite Server and restore the SELinux contexts on the moved files:

```
# mv var/tmp/export/* /var/www/html/
# cd /var/www/html
# restorecon -r *
```

NOTE

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

4. Add the CDN address to the Satellite web interface:

   a. Log in to the Satellite web interface.

   b. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest**.

   c. On the **Subscription Manifest** information screen, scroll to **Red Hat Provider Details**. Click the edit icon on the **Repository URL** entry and change the entry to the CDN's repository URL.

   d. Click **Browse** to choose the manifest file.

   e. Click **Upload** to import your manifest.

5. Enable the repositories from the local CDN:

   a. Click **Content** → **Red Hat Repositories**

   b. Enable the repositories that were enabled and synchronized in the Synchronizing Content section.

6. Click **Content** → **Sync Status**.

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

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

Report a bug

CHAPTER 5. RED HAT SATELLITE CAPSULE SERVERS

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

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

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

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

Report a bug

5.1. RED HAT SATELLITE CAPSULE SERVER SCALABILITY

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit but has been tested on a Satellite Server with a Red Hat Enterprise Linux 6.5 and 7 hostsystems. Currently, running fourteen capsules with two vCPUs have been tested to run without issues.

Report a bug

5.1.1. Capsule Scalability with Puppet Clients

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

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

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

For example, assuming that the puppet agent runs are evenly distributed with less than 100 concurrent puppet agents running at any single point during a run-interval, a Capsule Server with four CPUs can expect a maximum of 1250-1600 puppet clients with a moderate workload of 10 puppet classes assigned to each puppet client. Depending on the number of puppet clients required, the Satellite installation can scale out the number of Capsule Servers to support them.

Based on the following assumptions:

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

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

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

NOTE

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

Report a bug

5.2. RED HAT SATELLITE CAPSULE SERVER PREREQUISITES

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

Base Operating System

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

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

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

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

    NOTE

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

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

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

- Available subscriptions on the Red Hat Satellite Server.

IMPORTANT

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

Application Specifications

Satellite application installation specifications are as follows:

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

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

Required Network Ports

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

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

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

1. For Red Hat Enterprise Linux 6:

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

2. For Red Hat Enterprise Linux 7:

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

Red Hat Satellite Server

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

IMPORTANT

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

Report a bug

5.3. INSTALLING A RED HAT SATELLITE CAPSULE SERVER

Prerequisites

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

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

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

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

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

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

Procedure 5.1. To Install a Satellite Capsule Server on a Certificate-managed System:

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

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

   The screen displays:

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


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

   NOTE

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

2. Subscribe to the pool using the following command:

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

3. Disable all existing repositories:

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

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

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

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

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

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

Result:

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

Report a bug

5.4. CONFIGURING A RED HAT SATELLITE CAPSULE SERVER

Prerequisite

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

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

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

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

To configure a Satellite Capsule Server:

1. On the Satellite Server:

   a. Generate a Satellite Capsule Server certificate:

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

   Where:

   - **capsule_FQDN** is the Satellite Capsule Server's fully qualified domain name. (REQUIRED)

   - **certs-tar** is the name of the tar file to be generated that contains the certificate to be used by the Satellite Capsule installer.

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

   ```
       To finish the installation, follow these steps:
     1. Ensure that the capsule-installer is available on the
   system.
        The capsule-installer comes from the katello-installer
   package and
        should be acquired through the means that are appropriate to
   your deployment.
     2. Copy ~/capsule.example.com-certs.tar to the capsule system
   capsule.example.com
     3. Run the following commands on the capsule (possibly with the
   customized
        parameters, see capsule-installer --help and
        documentation for more info on setting up additional
   services):
     rpm -Uvh http://master.com/pub/katello-ca-consumer-
   latest.noarch.rpm
     subscription-manager register --org "Default Organization"
     capsule-installer --parent-fqdn          "sat6.example.com"\
                       --register-in-foreman  "true"\
                       --foreman-oauth-key
   "xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                       --foreman-oauth-secret
   "w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                       --pulp-oauth-secret
   "doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                       --certs-tar
   "~/capsule.example.com-certs.tar"\
                       --puppet               "true"\
                       --puppetca             "true"\
                       --pulp                 "true"
   ```

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

2. On the Satellite Capsule Server:

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

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

NOTE

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

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

b. Depending on the desired Satellite Capsule Server type, choose one of the following options:

i. Option 1: Satellite Capsule Server with Smart Proxy: This installs a Satellite Capsule Server with Smart Proxy features (DHCP, DNS, Puppet). Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn
"satellite.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet               "true"\
                    --puppetca             "true"\
                    --pulp                 "true"\
                    --tftp                 "true"\
                    --dhcp                 "true"\
                    --dhcp-interface       "virbr1"\
                   --dns                  "true"\
                   --dns-forwarders       "8.8.8.8"\
                   --dns-forwarders       "8.8.4.4"\
                   --dns-interface        "virbr1"\
                   --dns-zone             "example.com"
```

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

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

```
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet                "true"\
                    --puppetca              "true"\
                    --pulp                  "true"\
                    --tftp                  "true"\
                    --dhcp                  "true"\
                    --dhcp-interface        "virbr1"\
                   --dns                   "true"\
                   --dns-forwarders        "8.8.8.8"\
                   --dns-forwarders        "8.8.4.4"\
                   --dns-interface         "virbr1"\
                   --dns-zone              "example.com"
```

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

   - For Red Hat Enterprise Linux 6:

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

   - For Red Hat Enterprise Linux 7:

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

```
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP
```

NOTE

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

```
# echo $?
```

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

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

Result:

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

Report a bug

5.4.1. Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate

`katello-installer` comes with a default CA used both for the server ssl certificates as well as the client certificates used for authentication of the subservices. These certificates can be replaced with custom ones.

There are two instances wherein you can configure the Satellite Capsule Server to use a custom CA certificate:

1. When **`capsule-certs-generate`** is run the first time

2. After **`capsule-certs-generate`** has already been run

Procedure 5.2. Setting a Custom Server Certificate while running capsule-certs-generate for the first time

1. Run this command on the Red Hat Satellite Server:

```
capsule-certs-generate --capsule-fqdn "$CAPSULE"\
                       --certs-tar "~/$CAPSULE-certs.tar"\
                       --server-cert ~/path/to/server.crt\
                       --server-cert-req ~/path/to/server.crt.req\
                       --server-key ~/path/to/server.key\
                       --server-ca-cert ~/cacert.crt
```

Where:

- **`capsule_FQDN`** is the Satellite Capsule Server's fully qualified domain name. (REQUIRED)

- **`certs-tar`** is the name of the tar file to be generated that contains the certificate to be used by the Satellite Capsule installer.
- **`server-cert`** is the path to your certificate, signed by your certificate authority (or self signed)
- **`server-cert-req`** is the path to your certificate signing request file that was used to create the certificate.
- **`server-key`** is the private key used to sign the certificate
- **`server-ca-cert`** ~/path/to/cacert.crt the path to the CA certificate on this system.

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

3. On the Satellite Capsule Server:

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

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

NOTE

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

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

   b. Depending on the desired Satellite Capsule Server type, choose one of the following options:

      i. Option 1: Satellite Capsule Server with Smart Proxy: This installs a Satellite Capsule Server with Smart Proxy features (DHCP, DNS, Puppet). Run the following commands as the root user on the Satellite Capsule Server:

```

# capsule-installer --parent-fqdn
"satellite.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet               "true"\
                    --puppetca             "true"\
                    --pulp                 "true"\
                    --tftp                 "true"\
                    --dhcp                 "true"\
                    --dhcp-interface       "virbr1"\
                   --dns                  "true"\
                   --dns-forwarders       "8.8.8.8"\
                   --dns-forwarders       "8.8.4.4"\
                   --dns-interface        "virbr1"\
                   --dns-zone             "example.com"
```

ii. Option 2 - Satellite Capsule Server as a Content Node: This installs a Satellite Capsule Server with content management features and a Puppet Master. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn          "sat6.example.com"\
                    --register-in-foreman  "true"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"~/capsule.example.com-certs.tar"\
                    --puppet               "false"\
                    --puppetca             "false"\
                    --pulp                 "true"
```

iii. Option 3 - Satellite Capsule Server as a Content Node with Smart Proxy: This installs a Satellite Capsule Server with all the features. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn          "sat6.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet               "true"\
                    --puppetca             "true"\
                    --pulp                 "true"\
                    --tftp                 "true"\
                    --dhcp                 "true"\
                    --dhcp-interface       "virbr1"\
                   --dns                  "true"\
                   --dns-forwarders       "8.8.8.8"\
                   --dns-forwarders       "8.8.4.4"\
                   --dns-interface        "virbr1"\
                   --dns-zone             "example.com"
```

Procedure 5.3. Setting a Custom Server Certificate after running capsule-certs-generate

Using custom server certificates for the Satellite Server means that the same custom server certificates need to be deployed in the Satellite Capsule Servers. Each Satellite Capsule Server will require the following steps:

1. Generate a new certificate based on your custom server certificate:

```
capsule-certs-generate --capsule-fqdn "satcapsule.example.com"\
                       --certs-tar "~/$CAPSULE-certs.tar"\
                       --server-cert ~/path/to/server.crt\
                       --server-cert-req ~/path/to/server.crt.req\
                       --server-key ~/path/to/server.key\
                       --server-ca-cert ~/cacert.crt\
                       --certs-update-server --certs-update-server-
ca
```

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

3. On the Satellite Capsule Server, re-run the capsule-installer command to refresh the certificates. Depending on the desired Satellite Capsule Server type, choose one of the following options:

    a. Option 1: Satellite Capsule Server with Smart Proxy: This installs a Satellite Capsule Server with Smart Proxy features (DHCP, DNS, Puppet). Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn
"satellite.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet               "true"\
                    --puppetca             "true"\
                    --pulp                 "true"\
                    --tftp                 "true"\
                    --dhcp                 "true"\
                    --dhcp-interface       "virbr1"\
                   --dns                  "true"\
                   --dns-forwarders       "8.8.8.8"\
                   --dns-forwarders       "8.8.4.4"\
                   --dns-interface        "virbr1"\
                   --dns-zone             "example.com"
```

    b. Option 2 - Satellite Capsule Server as a Content Node: This installs a Satellite Capsule Server with content management features and a Puppet Master. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn          "sat6.example.com"\
                    --register-in-foreman  "true"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"~/capsule.example.com-certs.tar"\
                    --puppet               "false"\
                    --puppetca             "false"\
                    --pulp                 "true"
```

c. Option 3 - Satellite Capsule Server as a Content Node with Smart Proxy: This installs a Satellite Capsule Server with all the features. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn          "sat6.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf"\
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet               "true"\
                    --puppetca             "true"\
                    --pulp                 "true"\
                    --tftp                 "true"\
                    --dhcp                 "true"\
                    --dhcp-interface       "virbr1"\
                   --dns                  "true"\
                   --dns-forwarders       "8.8.8.8"\
                   --dns-forwarders       "8.8.4.4"\
                   --dns-interface        "virbr1"\
                   --dns-zone             "example.com"
```

IMPORTANT

Use the same custom server certificate on both the Red Hat Satellite Server and the Red Hat Satellite Capsule Server to ensure that the trusted relationship between the two hosts is maintained.

Report a bug

5.5. ADDING LIFECYCLE ENVIRONMENTS TO A RED HAT SATELLITE CAPSULE SERVER

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

IMPORTANT

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

To add environments to your Satellite Capsule Server:

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

   ```
   # hammer capsule list
   ```

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

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

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

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

   Where:

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

4. Add the lifecycle environment to the Satellite Capsule Server:

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

   Where:

   - **`--id`** is the Satellite Capsule Server's identification number.
   - **`--lifecycle-environment-id`** is the lifecycle environment's identification number.

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

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

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

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

   ```
     # hammer capsule content synchronize --id
   external_capsule_id_number --environment-id environment_id_number
   ```

Result:

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

Report a bug

5.6. REMOVING LIFECYCLE ENVIRONMENTS FROM THE RED HAT SATELLITE CAPSULE SERVER

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

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

To remove a lifecycle environment from the Satellite Capsule Server:

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

   ```
   # hammer capsule list
   ```

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

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

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

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

4. Remove the lifecycle environment from the Satellite Capsule Server:

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

   Where:

   - **`--id`** is the Satellite Capsule Server's identification number.
   - **`--environment-id`** is the lifecycle environment's identification number.

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

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

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

Result:

The chosen environments are removed from the Satellite Capsule Server.

Report a bug

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

Prerequisite

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

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

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

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

Report a bug

CHAPTER 6. UNINSTALLING RED HAT SATELLITE SERVER AND CAPSULE SERVER

WARNING

This procedure will erase all applications that are used with Red Hat Satellite Server or Red Hat Satellite Capsule Server on the target system. If you are using any of these applications or application data for any other purposes than Red Hat Satellite, backup the information before running this script.

The command to uninstall Red Hat Satellite Server or Red Hat Satellite Capsule Server is **`katello-remove`**. The uninstall script will issue a warning twice, requiring confirmation before it removes all packages and configuration files in the system. Below is a sample output of the command:

```
# katello-remove
WARNING: This script will erase many packages and config files.
Important packages such as the following will be removed:
  * elasticsearch
  * httpd (apache)
  * mongodb
  * tomcat6
  * puppet
  * ruby
  * rubygems
  * All Katello and Foreman Packages
Once these packages and configuration files are removed there is no going
back.
If you use this system for anything other than Katello and Foreman you
probably
do not want to execute this script.
Read the source for a list of what is removed.  Are you sure(Y/N)? y
ARE YOU SURE?: This script peramently deletes data and configuration.
Read the source for a list of what is removed.  Type [remove] to continue?
remove
Shutting down Katello services...
...
```

Result

All Red Hat Satellite packages and configuration files are removed from the system.

Report a bug

CHAPTER 7. NEXT STEPS

The content of the Installation Guide takes you through installing Red Hat Satellite Server, Capsule Server and to set up the repositories so that client host systems can update from the Satellite Server. There are other configuration steps you will need to take to take full advantage of your Red Hat Satellite Server and Capsule Server. The *Red Hat Satellite User Guide* can assist in configuring lifecycle environments, products, organizations, locations, and other components while the *Red Hat Satellite Provisioning Guide* can assist with setting up a working provisioning environment for your Red Hat Satellite Server.

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).

Compute Profile

Compute Profiles specify default attributes for new virtual machines on a compute resource.

Compute Resource

A *Compute Resource* is virtual or cloud infrastructure, which Red Hat Satellite 6 uses for deployment of hosts and systems. Examples include Red Hat Enterprise Virtualization Manager, OpenStack, EC2, and VMWare.

Content

Content includes software packages (RPM files) and Puppet modules. These are synchronized into the Library and then promoted into Life Cycle Environments using Content Views so that they can be consumed by Hosts.

Content Delivery Network (CDN)

The *Content Delivery Network (CDN)* is the mechanism used to deliver Red Hat content in a geographically co-located fashion. For example, content that is synchronized by a Satellite in Europe pulls content from a source in Europe.

Content View

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

External Node Classifier

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

Facter

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

Hammer

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

Hiera

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

Host

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

Host Collection

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

Host Group

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

Location

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

Library

The *Library* contains *every* version, including the latest synchronized version, of the software that the user will ever deploy. For an Information Technology Infrastructure Library (ITIL) [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.

Permission

The ability to perform an action.

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.

Repository

A *Repository* provides storage for a collection of content. For example, a YUM repository or a Puppet repository.

Role

A *Role* specifies a collection of permissions that are applied to a set of resources, such as Hosts.

Smart Proxy

A *Smart Proxy* is a Capsule Server component that can integrate with external services, such as **`DNS`** or **`DHCP`**.

Smart Variable

A *Smart Variable* is a configuration value that controls how a Puppet Class behaves. This can be set on a Host, a Host Group, an Organization, or a Location.

Standard Operating Environment (SOE)

A *Standard Operating Environment (SOE)* is a controlled version of the operating system on which applications are deployed.

Subscription

Subscriptions are the means by which you receive content and service from Red Hat.

Synchronizing

Synchronizing refers to mirroring content from external resources into the Red Hat Satellite 6 Library.

Synchronization Plans

Synchronization Plans provide scheduled execution of content synchronization.

User Group

A *User Group* is a collection of roles which can be assigned to a collection of users. This is similar to a Role in Red Hat Satellite 5.

User

A user is anyone registered to use Red Hat Satellite. Authentication and authorization is possible through built-in logic, through external LDAP resources, or with Kerberos.

Report a bug

[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#subscr-legacy

APPENDIX B. REVISION HISTORY

Revision 1-11 — Fri Jan 23 2015 — Athene Chan

BZ#1184589 Emphasize what base operating system variants is required for Red Hat Satellite.

Revision 1-10 — Fri Jan 23 2015 — Megan Lewis

BZ#1178176 Corrected 40G to 40GB in 4.2.4. Importing Content to a Disconnected Satellite Server.

BZ#1179022 Corrected errors in examples in 5.4. Configuring a Red Hat Satellite Capsule Server.

Revision 1-9 — Fri Jan 23 2015 — Athene Chan

BZ#1177568 Replaced the "service" and "chkconfig" command for chronyd to the recommended "systemctl" command instead.

Revision 1-8 — Wed Jan 21 2015 — David O'Brien

BZ 1184306 - Make the requirement for a Base install more obvious.

Revision 1-7 — Thu Dec 18 2014 — Megan Lewis

BZ#1168273 Corrected package name for installing puppet agent.

BZ#1169499 Clarified supported Red Hat Enterprise Linux variants in Prerequisites.

BZ#1164251 Corrected example in Adding Lifecycle Environments to a Red Hat Satellite Capsule Server.

BZ#1167904 Added chrony and sos into the prerequisites for install.

Revision 1-6.2 — Thu Nov 19 2014 — Athene Chan

Added additional admin and password options to the katello-installer.

Removed hashes on the firewall requirements.

Included references to support for scripting frameworks in the Puppet Supported Usage paragraph.

Revision 1-6.1 — Friday Nov 14 2014 — Athene Chan

BZ#1153567 Added a "Capsule Scalability" section.

Revision 1-6 — Thu Nov 13 2014 — Athene Chan

BZ#1153564 Added a "Next Steps" chapter.

BZ#1153772 Added firewall configuration and additional steps to ensure that the Satellite Server can go through the HTTP Proxy without issues.

BZ#1146574 Changed the gpg filename.

Revision 1-5 — Tue Nov 11 2014 — Athene Chan

BZ#1132840 Added two advanced firewall consideration tables in the prerequisites.

BZ#1152630 Edited incorrect reference to Red Hat Enterprise Linux 7.

BZ#1150412 Added "--complete-reload" to the firewall-cmd firewall commands.

BZ#1143746 Changed incorrect certs-server-key in procedure 2.4.

Revision 1-4 — Mon Nov 10 2014 — Athene Chan

BZ#1152630 Added RHEL7 firewall-cmd command examples for the firewall requirements.

Revision 1-3 — Fri Nov 7 2014 — Athene Chan

BZ#1161254 Added a new firewall rule to the list of firewall rules to allow katello-installer to run after initial install. Moved the firewall rules to the "Configuring Red Hat Satellite" sections to prevent errors.

Revision 1-2.02 — Fri Oct 3 2014 — Athene Chan

Various edits from translators' feedback.

BZ#1147673 Removed MS DHCP from supported DHCP features.

BZ#1140520 Changed all "ACME_Corporation" entries to the correct default organization entry "Default Organization".

BZ#1139806 Added a note in the Prerequisites sections for Red Hat Satellite Server and Red Hat Satellite Capsule Server that the host system has to be updated before installing Red Hat Satellite. BZ#1138430 Changed "yum-config-manager" to "subscription-manager" to match the procedure description to the command block.

BZ#1141954 Added example repositories to the "Enabling Red Hat Repositories" section and a note to enable RH Common repositories for client systems.

BZ#1140722 Added note to highlight that the command needs to change if the repository is different from the example command.

Revision 1-2.01 Fri Sep 12 2014 Athene Chan

BZ#1140875 Added firewall rules after the Satellite Server and Capsule Server installation.

Revision 1-2 Thu Sep 11 2014 Athene Chan

BZ#1140422 Changed the repository names for Red Hat Satellite Server and Red Hat Satellite Capsule Server.

Revision 1-1 Wed Sep 10 2014 Athene Chan

Added additional ports in the Prerequisites section.

Revision 1-0 Tue Sep 9 2014 Athene Chan

Red Hat Satellite 6.0 GA Release

Revision 0-34 Thu Aug 21 2014 Athene Chan

BZ#1131360 Replaced an option on the command to reflect the correct one.

Revision 0-33 Tue Aug 12 2014 Athene Chan

BZ#1130208 Added "Red Hat Software Collections" as a channel to enable.

BZ#1129104 Add requirement to make port 8080 available for katello installation. Update how to configure iptables accordingly.

BZ#1125241 Added a note that default location and default organization can be changed after initial configuration.

BZ#1044558 Added chapter on http proxy configuration options in katello-installer.

BZ#1120492 Added a note in "Red Hat Satellite Server Supported Usage" about embedded tomcat deployments.

BZ#1125299 Added references to "next steps" sections in the "Installing Red Hat Satellite" chapter.

BZ#1125357 Removed the deprecated repository directories.

BZ#1121814 Corrected the Satellite Capsule Server installer option.

BZ#1089086 Included filesize recommendations in the Prerequisites.

BZ#1119866 Added the Red Hat Software Collections package as a required package for the Satellite Capsule Server installation.

BZ#1118406 Added a table of ports, protocols and services in the Prerequisites section.

BZ#1120855 Various corrections on filenames and commands.

BZ#1121676 Added a note that all hammer commands are ran on the Satellite Server.

BZ#1113811 Created the section "Red Hat Satellite 6 Supported Usage".

BZ#1128922 Added a "Results" subsection.

BZ#754728 Added sections "Configuring Red Hat Satellite with a Custom Server Certificate" and "Configuring Red Hat Satellite Capsule Server with a Custom Server Certificate"

BZ#1122183 Changed the entry on Account Username and added an example for Base DN.

BZ#1129498 Group iptables commands for better readability.

Revision 0-32 Fri Jul 11 2014 Athene Chan

BZ#1157545, BZ#115047, BZ#1116471, BZ#1117052, BZ#1117052, BZ#1115065 Minor edits, spelling errors and revisions to text.

Revision 0-31 Mon Jun 30 2014 Athene Chan

Book published for Beta Release.

Revision 0-30 Tue Jun 24 2014 Dan Macpherson

Second test brewing for Beta.

Revision 0-29 **Tue Jun 24 2014** **Dan Macpherson**

Test brewing for Beta.

Revision 0-28 **Mon Nov 11 2013** **Dan Macpherson**

Fixing minor error.

Revision 0-27 **Mon 11 Nov 2013** **Dan Macpherson**

Preparation for MDP2.

Revision 0-26 **Mon 11 Nov 2013** **Athene Chan**

BZ#1024530, 1027466 Additional edits to steps for Satellite nodes.

Revision 0-25 **Thu 7 Nov 2013** **Megan Lewis**

BZ#1027461 Added steps to create activation key and retrieve oauth secret. Added note to verify nodes exist.

Revision 0-24 **Thu 7 Nov 2013** **Athene Chan**

BZ#1027466 Added a small seciton on using Satellite nodes. Added synchronization step.

Revision 0-23 **Wed 30 Oct 2013** **Athene Chan**

BZ#1024438 changed procedures to accommodate yum-utils installation.

BZ#1024529 removed katello.yml instructions as this is not preferred way of LDAP configuration.

BZ#1024559 Added foreman-libvirt to the yum install command.

BZ#1024530 Added new information to the section on Satellite Nodes.

Revision 0-22 **Tue 29 Oct 2013** **Athene Chan**

BZ#1024094 yum-utils command updated.

Revision 0-21 **Wed 09 Oct 2013** **Dan Macpherson**

Finalizing QE review implementation

Revision 0-20 **Wed 2 Oct 2013** **Athene Chan**

BZ#1014402 Installation requirements updated.

Revision 0-19 **Wed 2 Oct 2013** **Athene Chan**

BZ#1014402 Prerequisites for installation updated.

Revision 0-18 **Tue 1 Oct 2013** **Athene Chan**

BZ#1009719, 971944 Minor spelling and grammar edits.

Revision 0-17 **Thu 19 Sep 2013** **Athene Chan**

BZ#1009719 Updated the Prerequisites and the install instructions.

Revision 0-16 **Tue 17 Sep 2013** **Athene Chan**

BZ#971944 Added storage requirements for Satellite.

Revision 0-15 **Wed 11 Sep 2013** **Megan Lewis**

Integrating QE feedback.

Revision 0-14 **Mon 12 Aug 2013** **Dan Macpherson**

Removing draft watermark.

Revision 0-13 **Mon 12 Aug 2013** **Dan Macpherson**

Preparing documentation for technical review.

Revision 0-09 **Thu 20 June 2013** **Dan Macpherson**

Correction to repo label for installation.

Revision 0-08	**Thu 20 June 2013**	**Dan Macpherson**

Added MDP1 status.

Revision 0-07	**Wed 19 June 2013**	**Athene Chan**

Revised channel for installation.

Revision 0-06	**Thu 13 June 2013**	**Athene Chan**

Edited book for grammatical errors and sentence structure.

Revision 0-05	**Tue 11 June 2013**	**Athene Chan**

Added Chapters for manifests and for synchronization.

Edited sections based on technical review feedback.

Revision 0-04	**Fri 31 May 2013**	**Athene Chan**

Changed field names in the Satellite:Provisioning LDAP section.

Revision 0-03	**Thu 30 May 2013**	**Athene Chan**

Renamed all web application components to the rebranded names of "Red Hat Satellite: Content and Entitlement" and "Red Hat Satellite: Provisioning and Configuration".

Revision 0-02	**Tue 28 May 2013**	**Athene Chan**

Incorporated technical review edits.

Updated commands for installing Red Hat Satellite.

Standardized tagging of components.

Revision 0-01	**Fri 17 May 2013**	**Athene Chan**

Initial book creation