Red Hat Reference Architecture Series

Red Hat Cloud Infrastructure:

Deploying an On-Premise, Infrastructure-as-a Service, Private Cloud with Platform-as-a-Service Integration

Brett Thurber, Principal Software Engineer

RHCA, RHCVA

Version 1.1

September 2015

redhat.

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers.

Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:

https://www.facebook.com/rhrefarch

Follow us on Twitter:

https://twitter.com/RedHatRefArch

Plus us on Google+:

https://plus.google.com/u/0/b/114152126783830728030/

Table of Contents

1 Executive Summary

On-premise cloud deployments continue to increase in popularity and demand. As demand increases, IT leaders look for ways to reduce time to deploy, configure, and implement management capabilities for deployed environments. Red Hat Cloud Infrastructure 5 delivers on all fronts to include:

- Red Hat Satellite 6 for deployment, configuration, and lifecycle management
- Red Hat Enterprise Virtualization for hosting virtual infrastructure with enterprise applications
- Red Hat Enterprise Linux OpenStack Platform for creating repeatable, quickly deploy-able virtualized environments
- Red Hat CloudForms to provide best in class cloud management platform features such as lifecycle management, self-service capabilities, provider integration, chargeback reporting, and service automation

The focus of this reference architecture is to demonstrate deploying Red Hat Cloud Infrastructure technologies in an on-premise, Infrastructure-as-a-Service (IaaS) environment along with integrating Red Hat OpenShift Enterprise providing Platform-as-a-Service (PaaS) capabilities targeting the following detailed use cases:

- Configure Satellite 6 to support deploying RHCI components
- Deploy a self-hosted RHEV environment from Satellite 6
- Deploy a RHEL OSP environment from Satellite 6
- Deploy OpenShift Enterprise using OpenShift Origin puppet modules onto RHEV
- Deploy CloudForms and demonstrate discovery and management of RHCI on-premise cloud

2 Components Overview

2.1 Red Hat Cloud Infrastructure

Red Hat Cloud Infrastructure is a flexible private cloud solution that allows organizations to move to varying cloud infrastructure options on their timeline as their business and infrastructure needs dictate.

A single subscription product, Red Hat Cloud Infrastructure tightly integrates:

- Red Hat Enterprise Virtualization, a robust end-to-end enterprise virtualization solution
- Red Hat Enterprise Linux OpenStack Platform, a massively-scalable public cloud-like infrastructure based on the most popular open source cloud project ever
- Red Hat Satellite a robust systems management tool
- Red Hat CloudForms, the award-winning Cloud Management Platform that unifies operations for multiple hypervisor technologies and cloud technologies, enabling enterprise to deploy workloads to private clouds, public clouds, and the traditional datacenter as one cohesive environment.

In addition, Red Hat Cloud Infrastructure also includes Red Hat Enterprise Linux, both as the basis for Red Hat Enterprise Linux OpenStack Platform and Red Hat Enterprise Virtualization, as well as RHEL unlimited guests, should the customer select this option, for use with virtual machines.

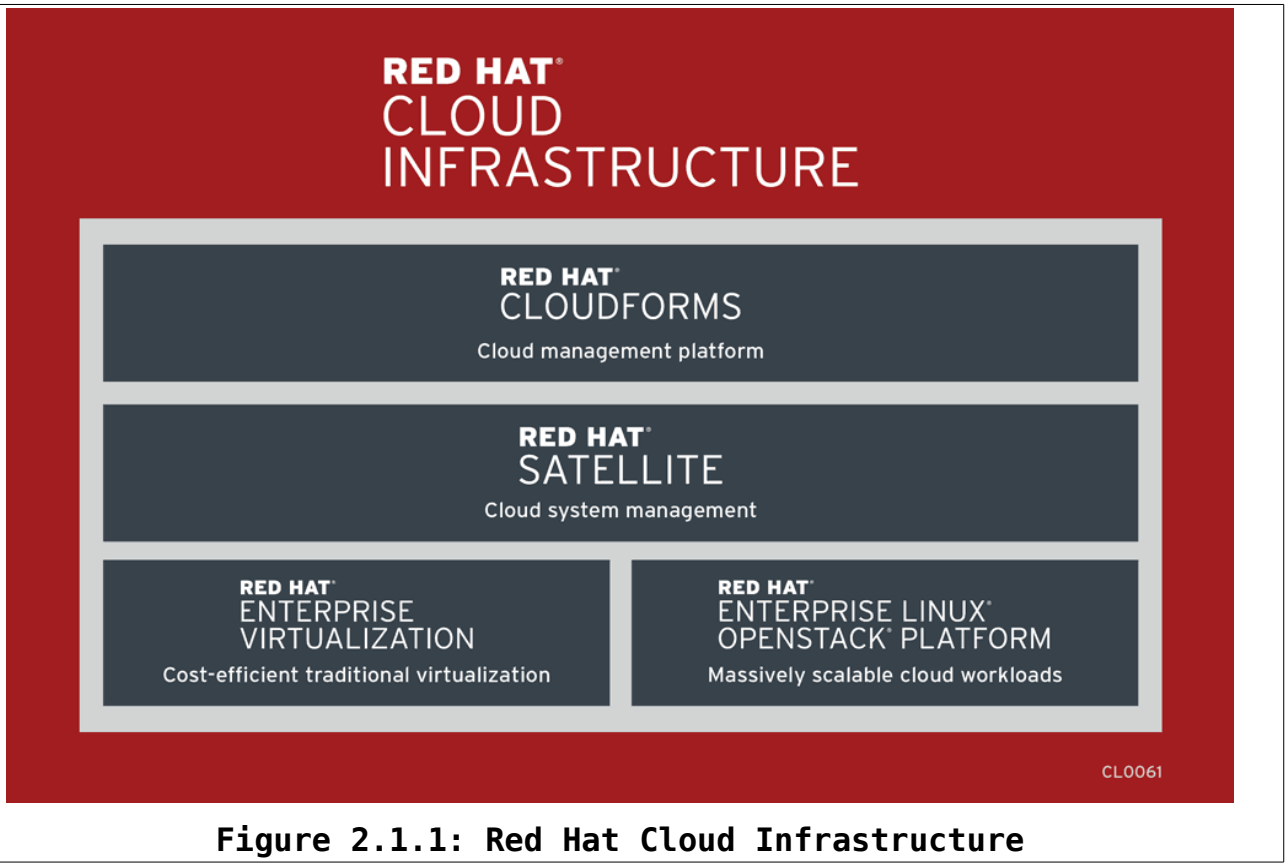


Figure 2.1.1: Red Hat Cloud Infrastructure

2.2 Red Hat Enterprise Virtualization

Red Hat Enterprise Virtualization is a complete virtualization management solution for virtualized servers and desktops. Created by the people who delivered Red Hat Enterprise Linux , Red Hat Enterprise Virtualization takes one beyond bare metal to meet critical business demands. It provides the performance advantages, competitive pricing, and the trusted, stable environment you expect from Red Hat. Red Hat Enterprise Virtualization provides common underlying services and management technologies for traditional virtualization workloads while also providing an on-ramp to high-level cloud functionality based on OpenStack (tech preview).

With Red Hat Enterprise Virtualization, one can:

- Take advantage of existing people skills and investments
- Decrease TCO and accelerate ROI
- Automate time-consuming and complicated manual tasks
- Standardize storage, infrastructure, and networking services on OpenStack

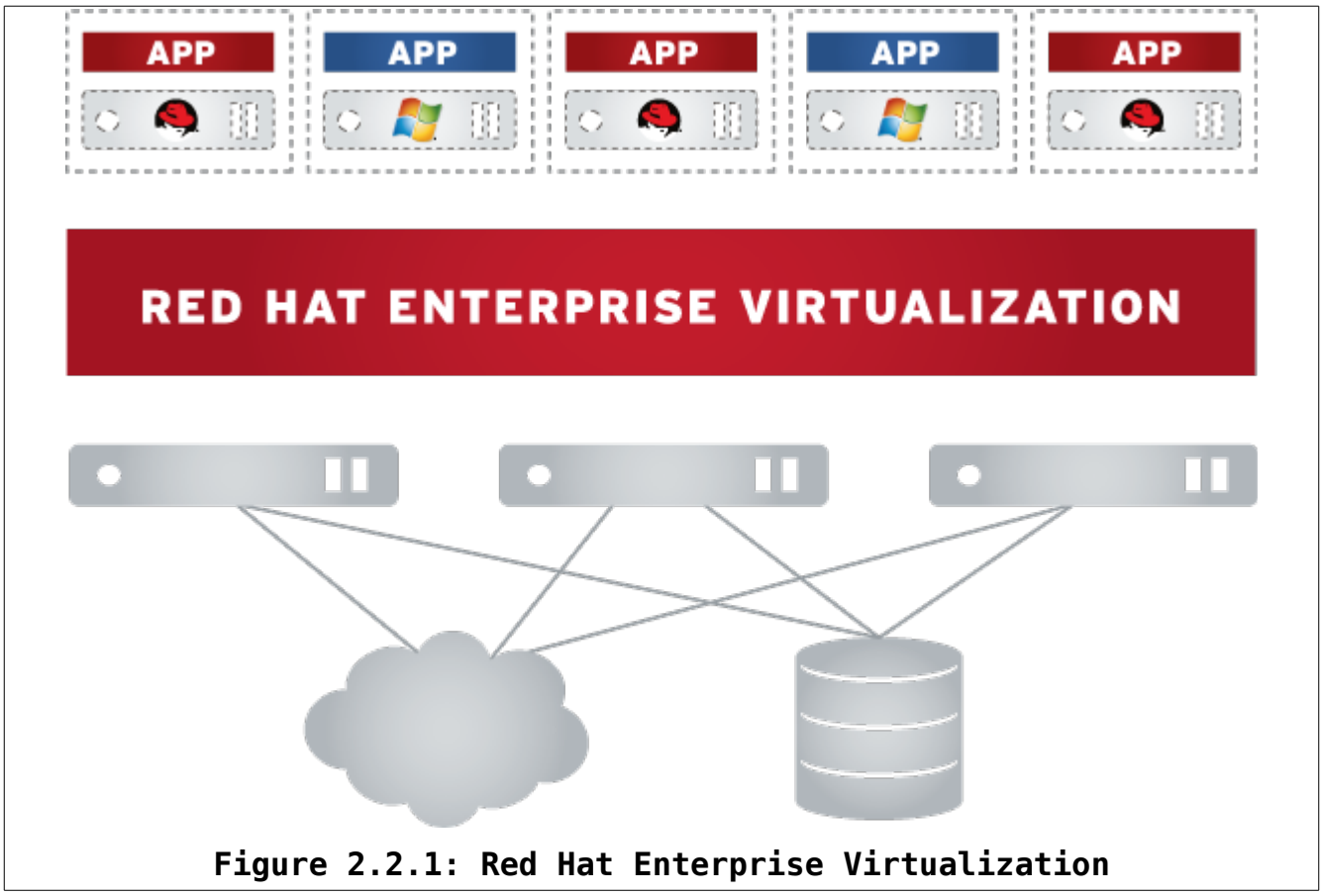


Figure 2.2.1: Red Hat Enterprise Virtualization

2.3 Red Hat Enterprise Linux OpenStack Platform

With Red Hat Enterprise Linux OpenStack Platform, each consumer receives all the benefits expected from Red Hat Enterprise Linux, plus the fastest-growing cloud infrastructure platform from OpenStack—both co-engineered to work seamlessly together because OpenStack is dependent on its underlying Linux operating system for everything from service operation and access, to hardware resources, to system performance, stability, and security.

Red Hat Enterprise Linux OpenStack Platform delivers the next-generation core IaaS and secondary IaaS+ infrastructure for a private or public cloud.

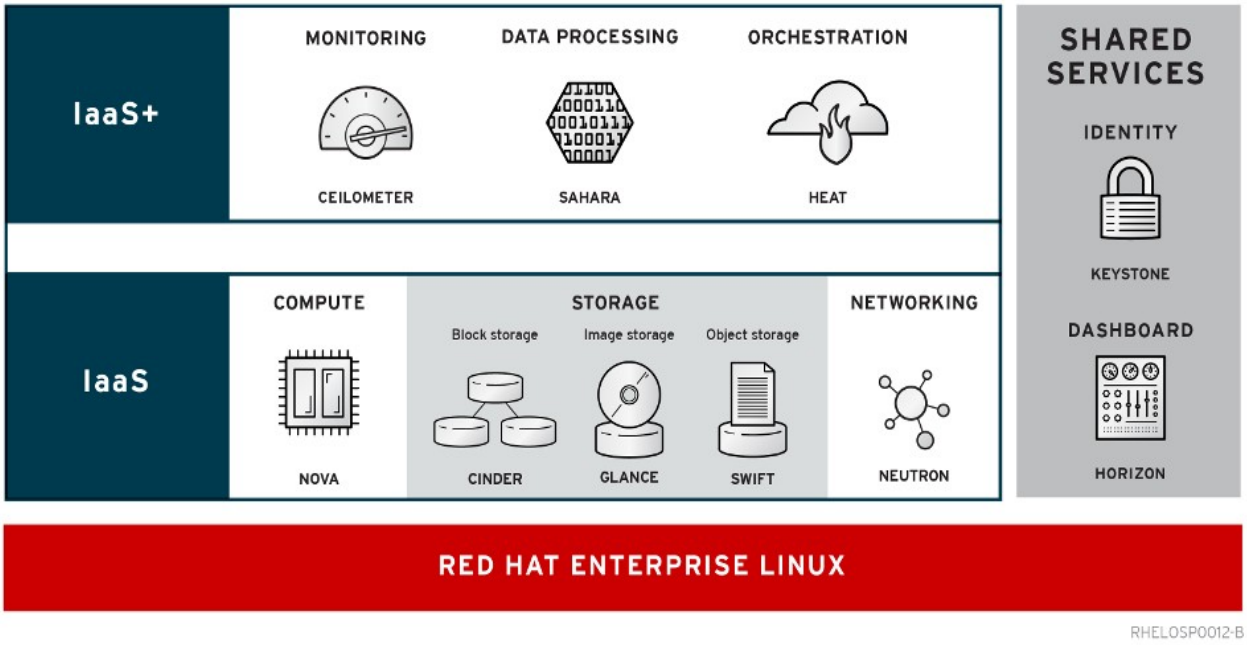


Figure 2.3.1: Red Hat Enterprise Linux OpenStack Platform

2.4 Red Hat Satellite

Red Hat Satellite is a systems management platform that makes Linux easier to deploy, scale, and manage. It provides lower total cost of ownership (TCO) in life cycle management, and scales to an IT environment as it grows. The latest release of Red Hat's systems management solution, Red Hat Satellite 6, delivers dramatic improvements across system provisioning, configuration management, content management, and overall scalability and security.

Red Hat Satellite improves the ability to deploy and update hosts and securely manage an environment. A flexible and scalable architecture means that a Satellite deployment can grow along with an organization.

Red Hat Satellite 6 introduces new capabilities across system provisioning, configuration management, and content and life cycle management, along with improved Red Hat subscription management.

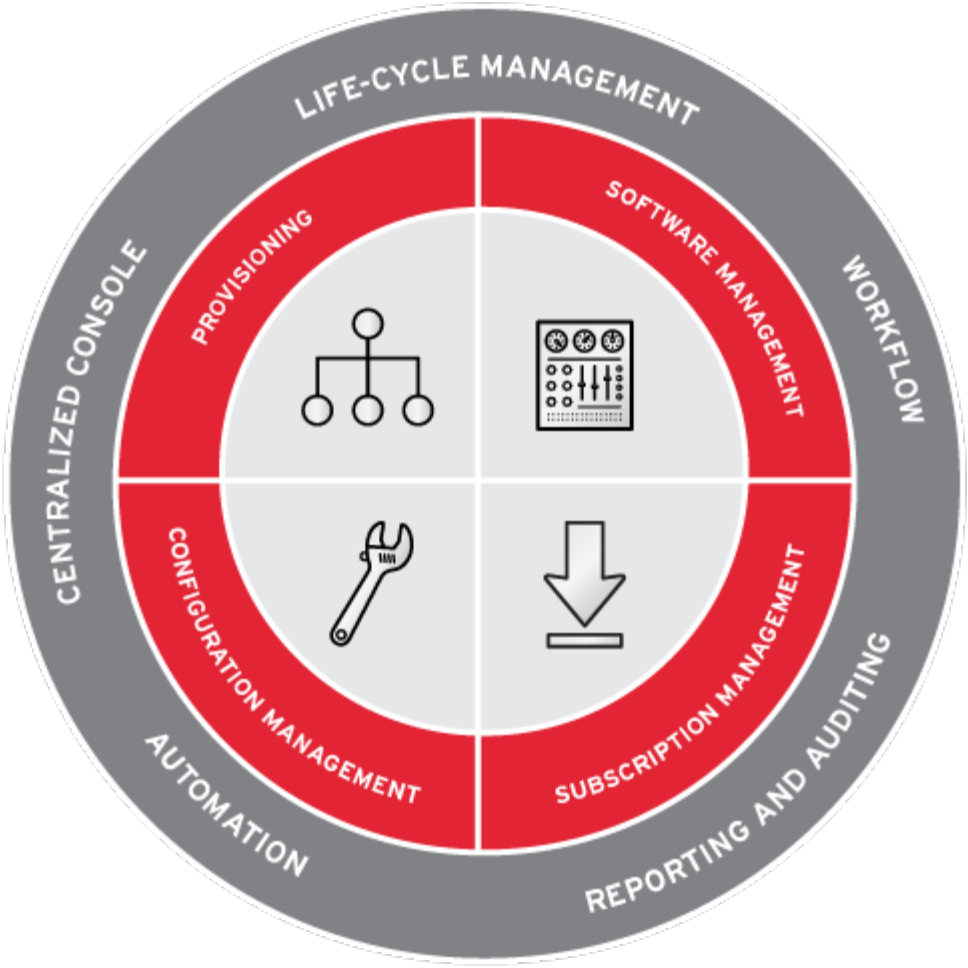


Figure 2.4.1: Red Hat Satellite

2.5 Red Hat CloudForms

Gain control of a virtualization environment, and build and manage a private or hybrid cloud. Red Hat CloudForms can do both with a comprehensive management platform. As needs change, CloudForms evolves, protecting investments and providing a continuum of capabilities as IT progresses toward Infrastructure-as-a-Service (IaaS) models.

CloudForms can transform existing virtual environments into private clouds, hybrid clouds, or both. Seamlessly add new infrastructure platforms to expand the cloud model and take advantage of better cost, performance, density, innovation, or whatever the need.

Dynamically and automatically ensure the most efficient use of resources, including the ability to:

- Discover and track resource changes
- Provision and de-provision resources based on policies and demand
- Identify the current condition of resources and the best fit for new workloads across compute, storage, and network resources
- Anticipate and plan for future resource requirements based on capacity, trending, data, and analytics

CloudForms also allows anticipation and planing for future resource requirements based on capacity, trending, data, and analytics.

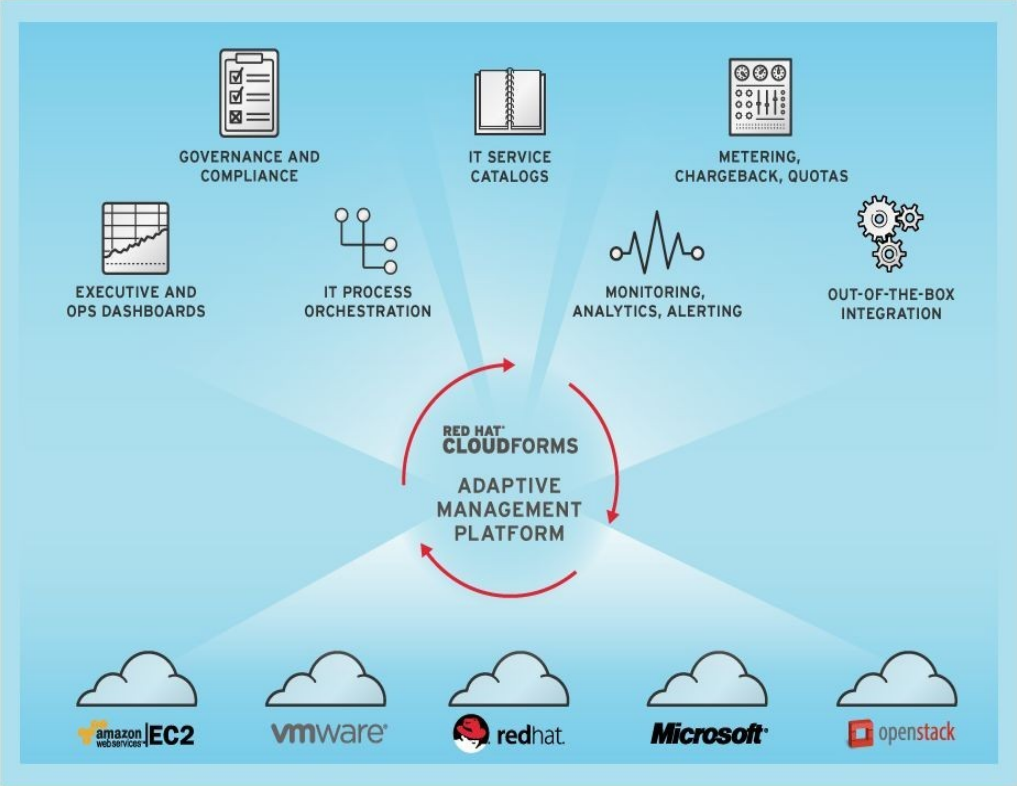


Figure 2.5.1: Red Hat CloudForms

2.6 Red Hat OpenShift Enterprise

OpenShift Enterprise by Red Hat is a cloud computing Platform-as-a-Service (PaaS) solution designed for on-premise or private cloud deployments.

While Infrastructure-as-a-Service (IaaS) provides on-demand access to raw compute resources, and Software-as-a-Service (SaaS) provides on-demand access to a complete application, OpenShift Enterprise PaaS enables on-demand access to a cloud-based application platform. This lets enterprises easily build the applications they need and run them

in a cloud architecture.

OpenShift Enterprise automates much of the provisioning and systems management of the application platform stack. This allows IT operations teams to more easily meet the growing demand for new application services coming from the business.

OpenShift Enterprise provides on-demand, elastic, scalable, and fully configured application development, testing, and hosting environment for application developers so they can focus on coding those new application services. Once installed in a datacenter or a private cloud, OpenShift Enterprise provides a self-service capability to developers. This allows developers to easily create scalable applications in the PaaS environment with their choice of programming languages and middleware, and begin coding applications from their favorite development environments

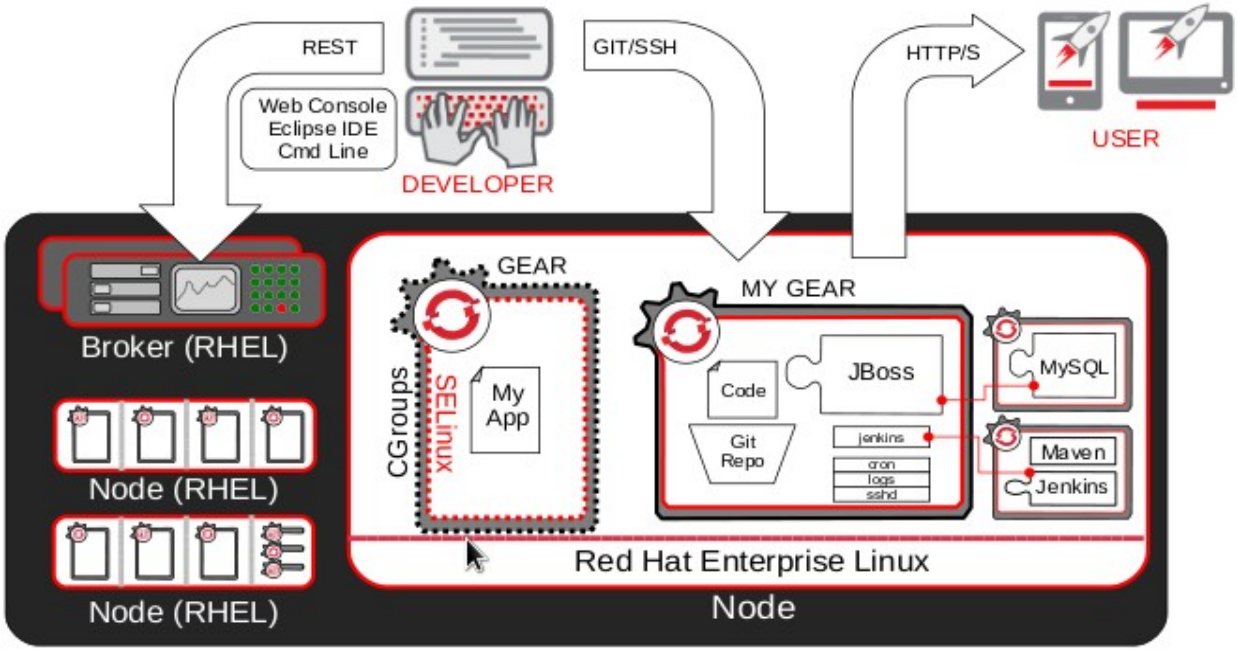


Figure 2.6.1: Red Hat OpenShift Enterprise

3 Environment

The reference environment consists of the following:

Red Hat Enterprise Virtualization (self-hosted):

- One NFS server configured to use iSCSI backed storage for the self-hosted Red Hat Enterprise Virtualization Manager virtual machine
- One Red Hat Enterprise Virtualization Hypervisor attached to iSCSI storage for the data domain

Red Hat Enterprise Virtualization (infrastructure):

- One Red Hat Enterprise Virtualization Manager hosted from the NFS server via KVM
- Two Red Hat Enterprise Virtualization Hypervisors attached to iSCSI storage for the data domain

Red Hat Enterprise Linux OpenStack Platform

- One Red Hat Enterprise Linux OpenStack Platform installer
- One Red hat Enterprise Linux OpenStack Platform controller node
- One Red Hat Enterprise Linux OpenStack Platform compute node with Nova networking

Red Hat Satellite

- One Red Hat Satellite 6 virtual machine hosted within a Red Hat Enterprise Virtualization environment with integrated capsule service DNS, DHCP, puppet, PXE and TFTP services

Red Hat Identity Management

- One Red Hat Identity Management virtual machine to provide single sign on capabilities hosted within a Red Hat Enterprise Linux environment

Red Hat CloudForms

- One Red Hat CloudForms appliance hosted within the self-hosted Red Hat Enterprise Virtualization environment

Red Hat OpenShift Enterprise

- One OpenShift Broker and several OpenShift Nodes hosed within the self-hosted Red Hat Enterprise Virtualization environment

The following diagram depicts the reference environment configuration.

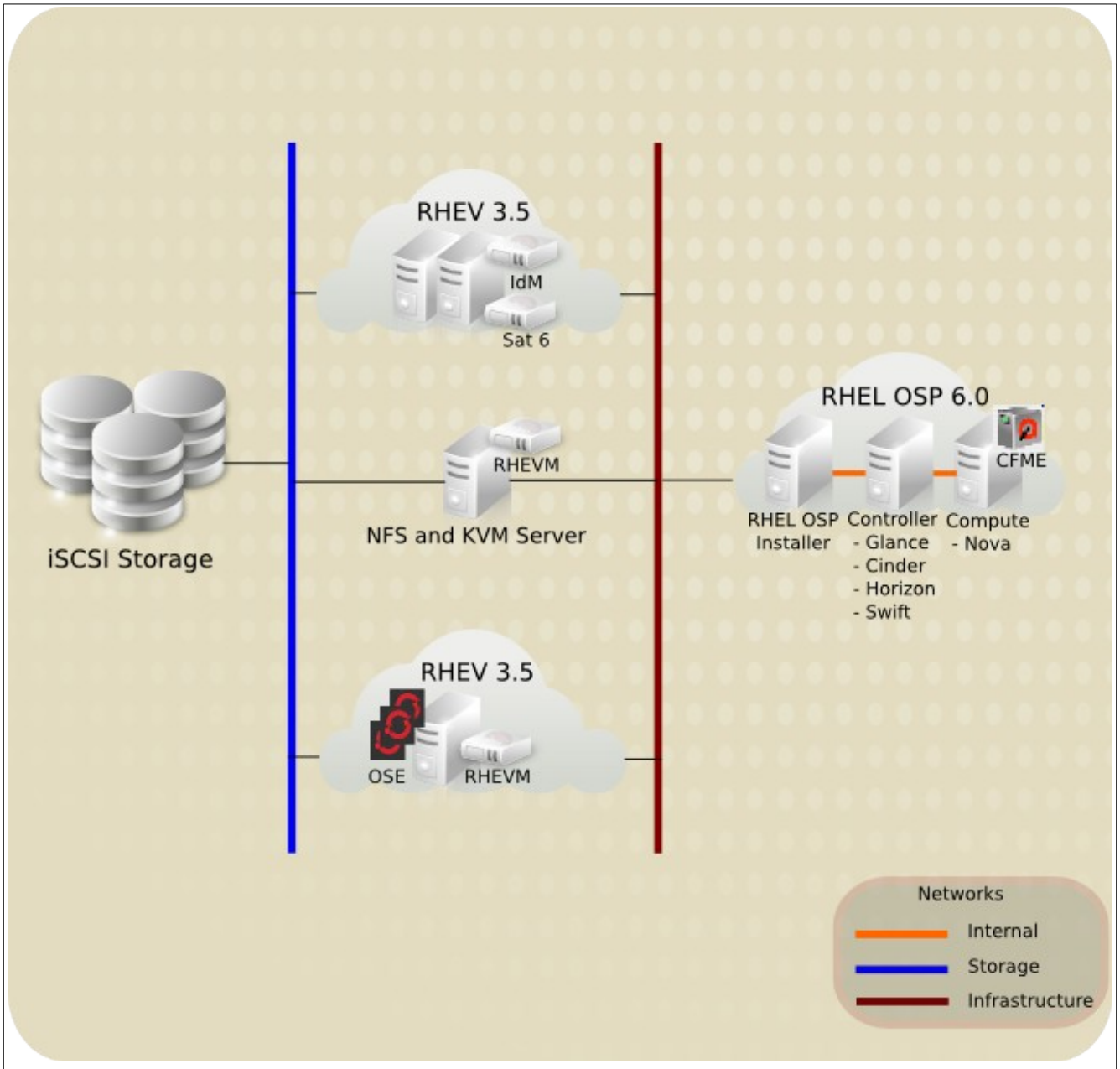


Figure 3.1: Reference Architecture Components

3.1 Software

The following section details the software versions used in the reference environment.

3.1.1 Red Hat Satellite

The following table lists the software version used for Red Hat Satellite Server.

System	Software Version	Role(s)
Satellite Server (rhci-sat6)	6.0.8	DNS, DHCP, Provisioning, Puppet

Table 3.1.1-1: Satellite Server – Software Versions

3.1.2 Red Hat Enterprise Virtualization

The reference environment utilizes two Red Hat Enterprise Virtualization 3.5 configurations. One for deployment from Satellite and hosting OpenShift Enterprise and a second for hosting infrastructure machines.

Systems	Configuration	Software Versions	Role(s)
Red Hat Enterprise Virtualization self-hosted (rhci-rhevm)	RHEV 3.5 environment • One RHEL Hypervisor	• RHEV Manager (3.5): • 3.5.0-0.32.el6ev • RHEL Hypervisor – 7.0 • libvirt-1.1.1-29.el7_0.7 • VDSM Version: vdsm-4.16.8.1-6.el7ev	OpenShift Enterprise hosting

Table 3.1.2-1: RHEV Environment: Self-Hosted – Software Versions

Systems	Configuration	Software Versions	Role(s)
Red Hat Enterprise Virtualization (cf-rhevm-32)	RHEV 3.5 environment • Two RHEV Hypervisors	• RHEV Manager (3.5): • 3.5.0-0.32.el6ev • RHEV Hypervisor - 6.6 - 20150128.0.el6ev • VDSM Version: vdsm-4.16.8.1-6.el6ev	Infrastructure hosting

Table 3.1.2-2: RHEV Environment: Infrastructure – Software Versions

3.1.3 Red Hat Enterprise Linux OpenStack Platform

The following table lists the software version used for RHEL OSP 6.0.

Systems	Software	Version
Controller (rhci-rhelosp-glance) Glance, Cinder, Horizon, Ceilometer	qpid-cpp-server	0.22-50
	openstack-keystone	2014.2.2-1
	openstack-nova-{api,cert,common,conductor,scheduler,console, novncproxy}	2014.2.2-19
	openstack-glance	2014.2.2-1
	openstack-cinder	2014.2.2-2
	openstack-dashboard	2014.2.2-2
	openstack-utils	2014.2-1
	openstack-selinux	0.6.27-1
	openstack-ceilometer-{common,collector,central,alarm,api}	2014.2.2-2
Compute (rhci-rhelosp-nova) Nova compute Nova networking	openstack-nova-{common,compute}	2014.2.2-19
	openstack-utils	2014.2-1
	openstack-selinux	0.6.27-1
	openstack-nova-network	2014.2.2-19
	openstack-ceilometer-compute	2014.2.2-2
Installer (rhci-rhelosp-inst)	foreman-proxy	1.6.0.30-5
	foreman-selinux	1.6.0.14-1
	foreman	1.6.0.49-6
	foreman-discovery-image	7.0-20140905.0.2
	puppet	3.6.2-2
	rhel-osp-installer-client	0.5.5-5
	rhel-osp-installer	0.5.5-5

Table 3.1.3-1: RHEL OSP – Software Versions

3.1.4 Red Hat OpenShift Enterprise

The following table lists the software versions used for OSE 2.2 to include the OpenShift Origin puppet modules.

Systems	Software	Versions
Red Hat OpenShift Enterprise Broker	openshift-origin-broker	1.16.2.2-1
	openshift-origin-broker-util	1.32.1.1-1
	rubygem-openshift-origin-dns-nsupdate	1.16.3.0-1
	rubygem-openshift-origin-msg-broker-mcollective	1.31.1.1-1
	rubygem-openshift-origin-auth-remote-user	1.21.1.0-1
Red Hat OpenShift Enterprise Node	openshift-origin-node-util	1.32.4.1-1
	openshift-origin-cartridge-php	1.29.1.0-1
	openshift-origin-msg-node-mcollective	1.27.1.1-1
	rubygem-openshift-origin-node	1.33.2.1-1
OpenShift Origin Puppet Module	openshift-openshift_origin	4.1.2

Table 3.1.4-1: OpenShift Enterprise – Software Versions

3.1.5 Red Hat CloudForms

The following table lists the software version used for Red Hat CloudForms.

System	Software Version	Role(s)
CloudForms Management Engine (rhci-cfme)	5.3.3.2.20150217120931_a465215	Orchestration and Management

Table 3.1.5-1: CloudForms Management Engine – Software Versions

3.1.6 Red Hat Identity Management

The following table lists the software version used for Red Hat IdM.

System	Software Version	Role(s)
Identity Management (rhci-idm)	ipa-server-3.3.3-28.el7_0.3	Authentication

Table 3.1.6-1: Identity Management – Software Versions

3.2 Systems

The following describes the physical and virtual machine configurations used in the reference environment.

3.2.1 Server Hardware

All seven physical systems use the same hardware platform type:

Component	Details
Blade Chassis	IBM BladeCenter H - 8852HC1
Blade Server	IBM BladeServer – HS22 - 70870
CPU	(2) Intel Xeon X5680 (6 core @3.33 GHz)
Memory	52 GB
Network	(2) Broadcom Corporation NetXtreme II BCM5709S Gigabit Ethernet (2) Emulex Virtual Fabric Adapter (CFFh) 10GB Ethernet
Disk	2 x 146 GB SAS

Table 3.2.1-1: Server Hardware Configuration

3.2.2 Red Hat Enterprise Virtualization Virtual Machines

The following virtual machines provide infrastructure resources.

CloudForms Management Engine

Component	Details
CPU	4
Memory	6144 MB
Network	1 bridged virtIO
Disk	Disk 1 – 40 GB (OS) Disk 2 – 20 GB (CFME database)

Table 3.2.2-1: CFME – Virtual Machine Configuration

Red Hat Satellite Server

Component	Details
CPU	2
Memory	8192 MB
Network	1 bridged virtIO
Disk	Disk 1 – 120GB (OS and Satellite)

Table 3.2.2-2: Satellite Server – Virtual Machine Configuration

Red Hat IdM Server

Component	Details
CPU	1
Memory	4096 MB
Network	1 bridged virtIO
Disk	Disk 1 – 15 GB (OS)

Table 3.2.2-3: IdM Server – Virtual Machine Configuration

RHEV Manager – Self-Hosted

Component	Details
CPU	1
Memory	8192 MB
Network	1 bridged virtIO
Disk	Disk 1 – 15 GB (OS)

Table 3.2.2-4: RHEV-M – Virtual Machine Configuration

Red Hat OpenShift Enterprise – Broker and Nodes

Component	Details
CPU	1
Memory	4096 MB
Network	1 bridged virtIO
Disk	Disk 1 – 15 GB (OS)

Table 3.2.2-5: OSE Broker and Nodes – Virtual Machine Configuration

3.3 Storage

Non-local storage is provided by an EMC Celerra NS-120.

System	Disk Size
RHEV Infrastructure	1.15 TB
RHEV self-hosted	500 GB
NFS for self-hosted RHEVM	100 GB

Table 3.3-1: Storage Configuration

4 Red Hat Satellite 6 Configuration

Red Hat Satellite 6 is the foundation for deploying Red Hat Cloud Infrastructure. This section describes the details and steps needed in order to successfully deploy the RHCI components to also include Red Hat OpenShift Enterprise. However, the assumption made is that Red Hat Satellite 6 is already deployed whereas those steps are not covered.

Red Hat Satellite 6 installation and Deployment steps are found in the *Red Hat Satellite 6.0 Installation Guide*[1].

> **Note:** The reference environment configuration and settings are meant to serve for guidance only. There are many ways to configure Satellite 6 depending upon environment requirements.

4.1 Capsule Settings

For the reference environment the built-in capsule is configured to provide DHCP, DNS, TFTP, PXE, Pulp, and puppet services. Refer to **Appendix C: Satellite 6 Install Settings**, for Satellite 6 install options used via *katello-installer*.

4.1.1 Capsule

Login to the Satellite 6 portal using the *admin* account. Navigate to **Infrastructure -> Capsules** and verify the capsule is present with the desired features.

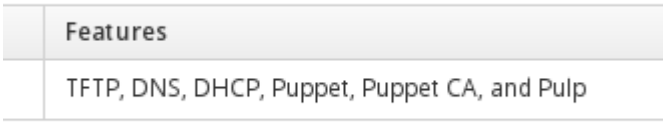


Figure 4.1.1.1: Capsule Settings

1 https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.0/html/Installation_Guide/index.html

4.1.2 Domain

Next navigate to **Infrastructure -> Domains** and click **New Domain**. Under the **Domain** tab provide **DNS Domain** and on the drop down for **DNS Capsule**, select the FQDN of the Satellite server.

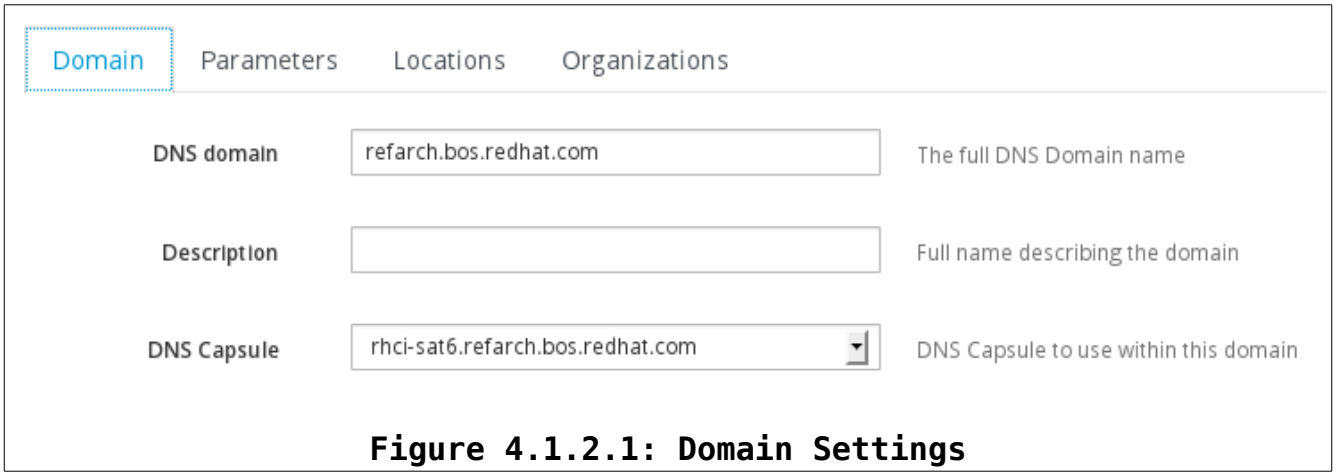


Figure 4.1.2.1: Domain Settings

Click **Submit** to save changes.

4.1.3 Subnet

Navigate to **Infrastructure -> Subnets** and click **New Subnet**. Under the **Subnet** tab provide desired settings for **Name**, **Network address**, **Network mask**, **Gateway address**, **Primary DNS server**, **Secondary DNS server**, **Start of IP range**, **End of IP range**, and **VLAN ID**. For the reference environment the following settings are used:

Setting	Value
Name	syseng
Network address	10.19.10.0
Network mask	255.255.254.0
Gateway address	10.19.11.254
Primary DNS server	10.19.11.51 (IP address of Satellite Server)
Start of IP range	10.19.11.100
End of IP range	10.19.11.200

Table 4.1.3-1: Subnet Settings

Select the **Domain** tab and and place a check mark next to the **Domain** created in a previous step.

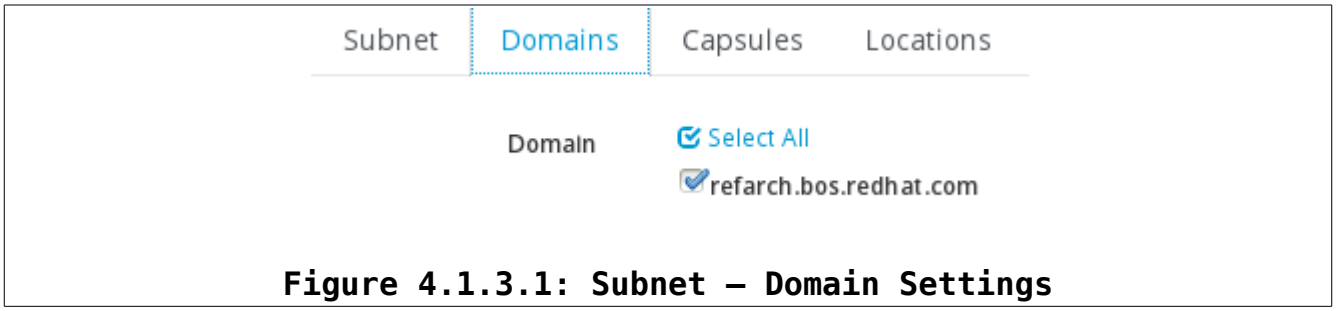


Figure 4.1.3.1: Subnet – Domain Settings

Select the **Capsules** tab and for **DHCP Capsule**, **TFTP Capsule**, and **DNS Capsule** drop downs, select the FQDN of the Satellite server for each.

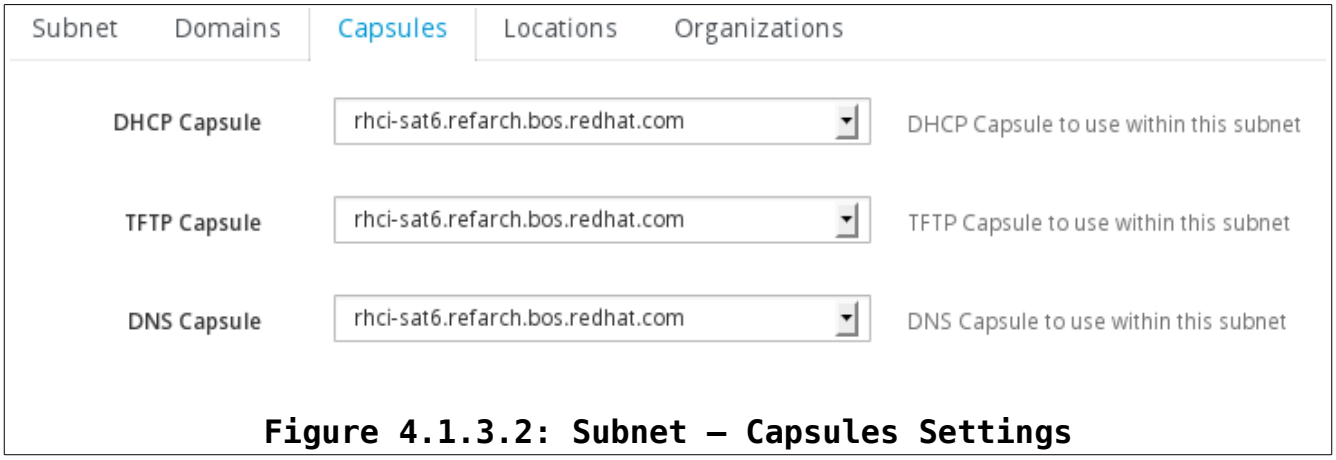


Figure 4.1.3.2: Subnet – Capsules Settings

Observe that no selections are made for **Locations** or **Organizations**.

Click **Submit** to save the changes.

4.2 IdM Integration (optional)

Satellite 6 supports integration with Lightweight Directory Access Protocol[2] (LDAP) server to provide single-sign on capabilities. This integration allows for user and group mappings into built-in Satellite 6 role-based access control (RBAC) along with realm support.

For the reference environment Red Hat Identity Management is configured as an LDAP provider.

4.2.1 LDAP Provider

To add an LDAP provider, as the a*dmin* user, navigate to **Administer -> LDAP Authentication**. Click **New LDAP Source**.

2 https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.0/html/User_Guide/chap-Users_and_Roles.html

Under the **LDAP server** tab provide a **Name**, **Server**, **Port**, and chose TLS if configured.

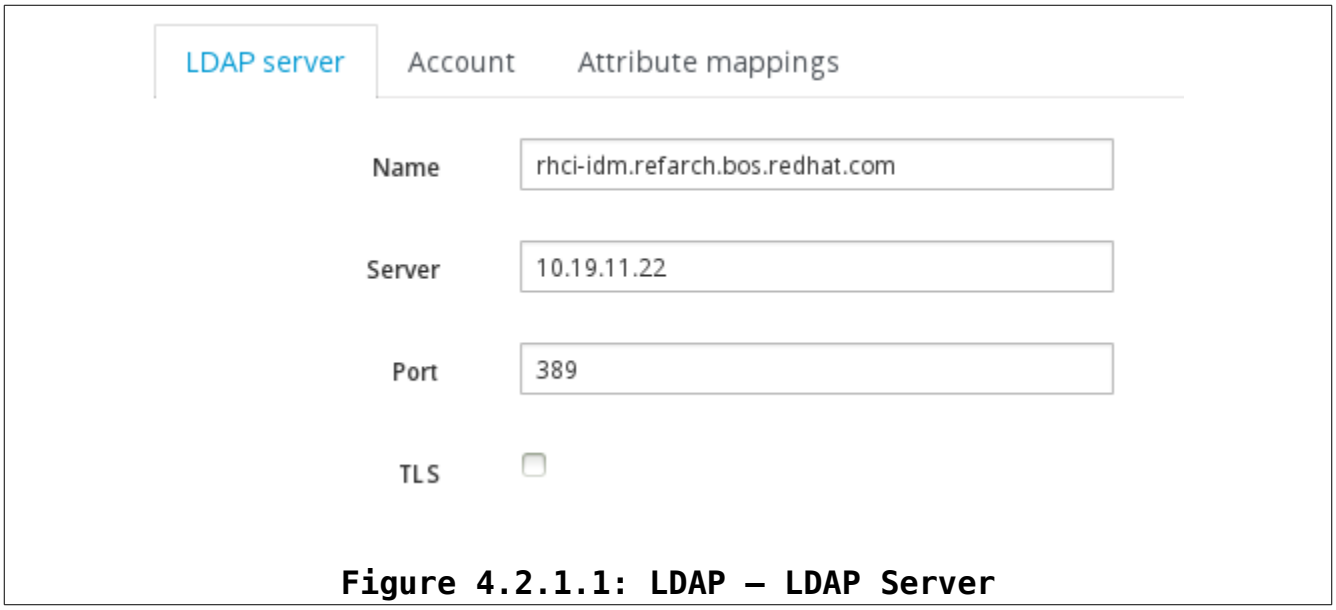


Figure 4.2.1.1: LDAP – LDAP Server

On the **Account** tab provide **Account username**, **Account password**, **Base DN**, **LDAP filter**, and choose to **Automatically create accounts in Foreman**. For the reference environment the following settings are used:

Setting	Value
Account username	uid=admin,cn=users,cn=accounts,dc=refarch,dc=bos,dc=redhat,dc=com
Account password	<REDACTED>
Base DN	cn=users,cn=accounts,dc=refarch,dc=bos,dc=redhat,dc=com
Automatically create accounts in Foreman	checked

Table 4.2.1-1: LDAP – Account Settings

On the **Attribute mappings** tab default settings are used.

Click **Submit** to save the changes.

4.2.2 LDAP Users and Roles

With an LDAP provider configured, use a LDAP provided user account to login to the Satellite 6 user interface. For the reference environment, an LDAP user named *test* is used.

Upon logging in with the *test* user, a permission denied message is shown as this user has not been assigned to any roles, organizations, or locations.

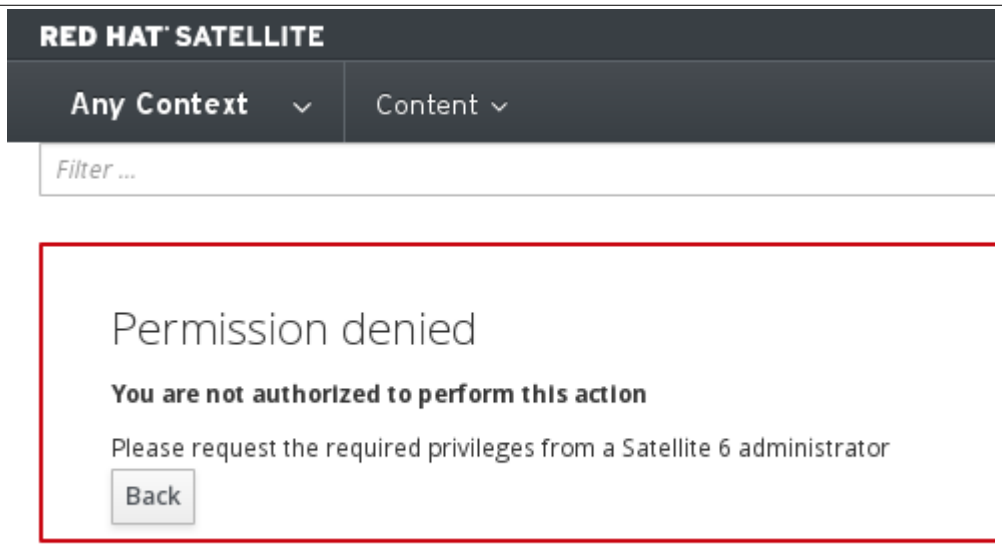


Figure 4.2.2.1: New LDAP User

As the a*dmin* user, login to the Satellite 6 user interface and navigate to **Administer -> Users**. The *test* LDAP user account appears under **Users**.

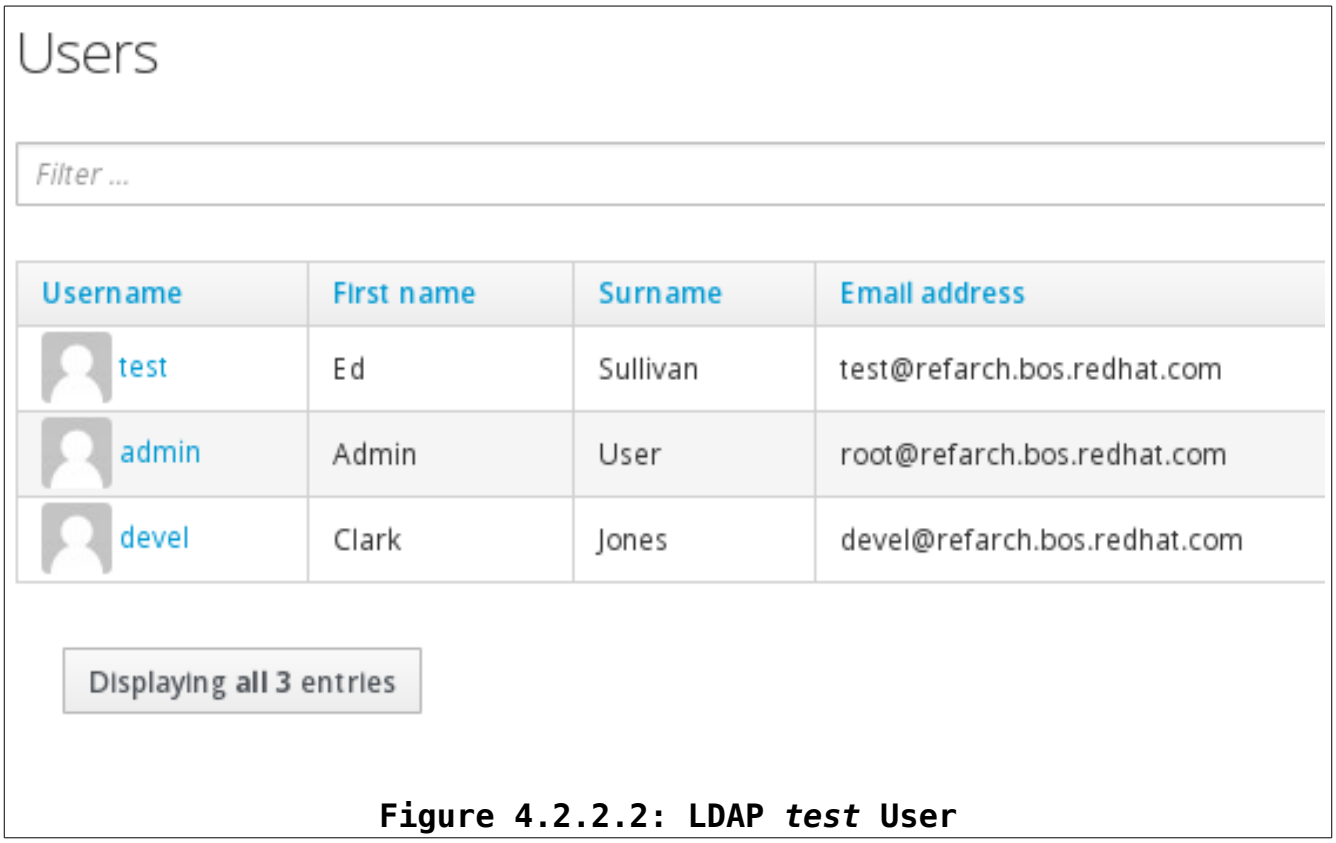


Figure 4.2.2.2: LDAP *test* User

Select the *test* account and verify the settings under the **User** tab. Attributes should be mapped over from the LDAP provider. **Authorized by** should automatically be populated with the LDAP provider.

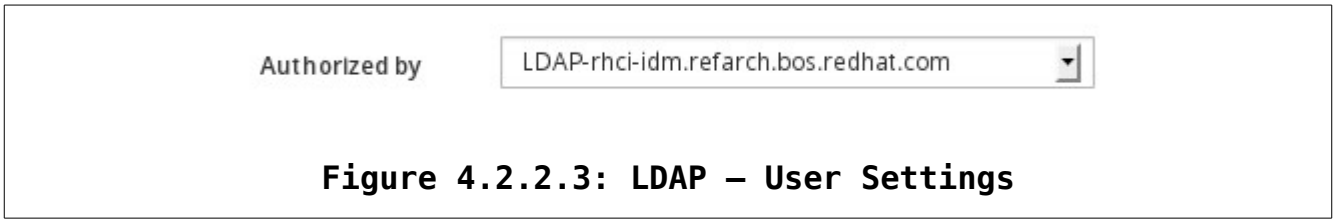


Figure 4.2.2.3: LDAP – User Settings

Under the **Locations** tab select the location desired. For the reference environment *Boston* is selected.

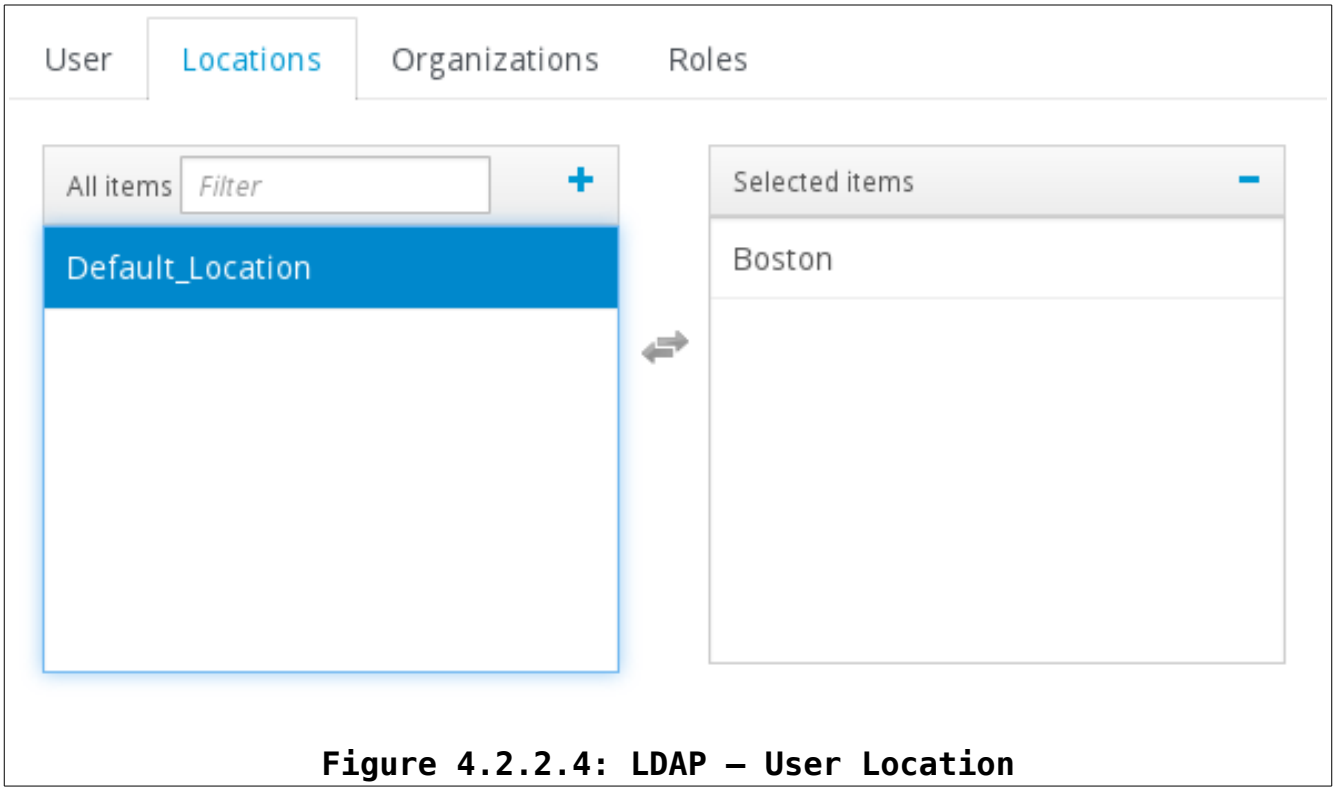


Figure 4.2.2.4: LDAP – User Location

Under the **Organizations** tab select the desired organization. For the reference environment *Systems Engineering* is selected.

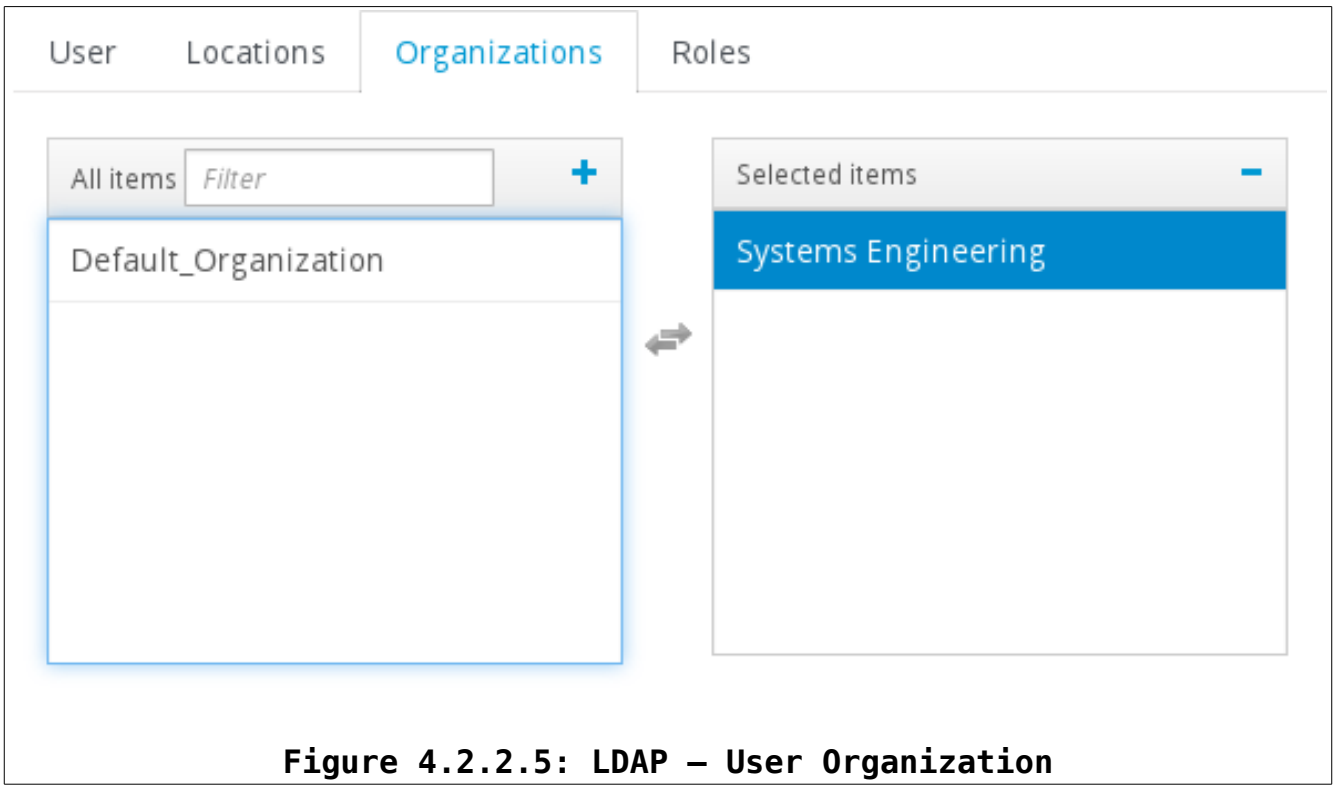


Figure 4.2.2.5: LDAP – User Organization

Under the **Roles** tab, select the desired roles. For the reference environment *Manager* is selected.

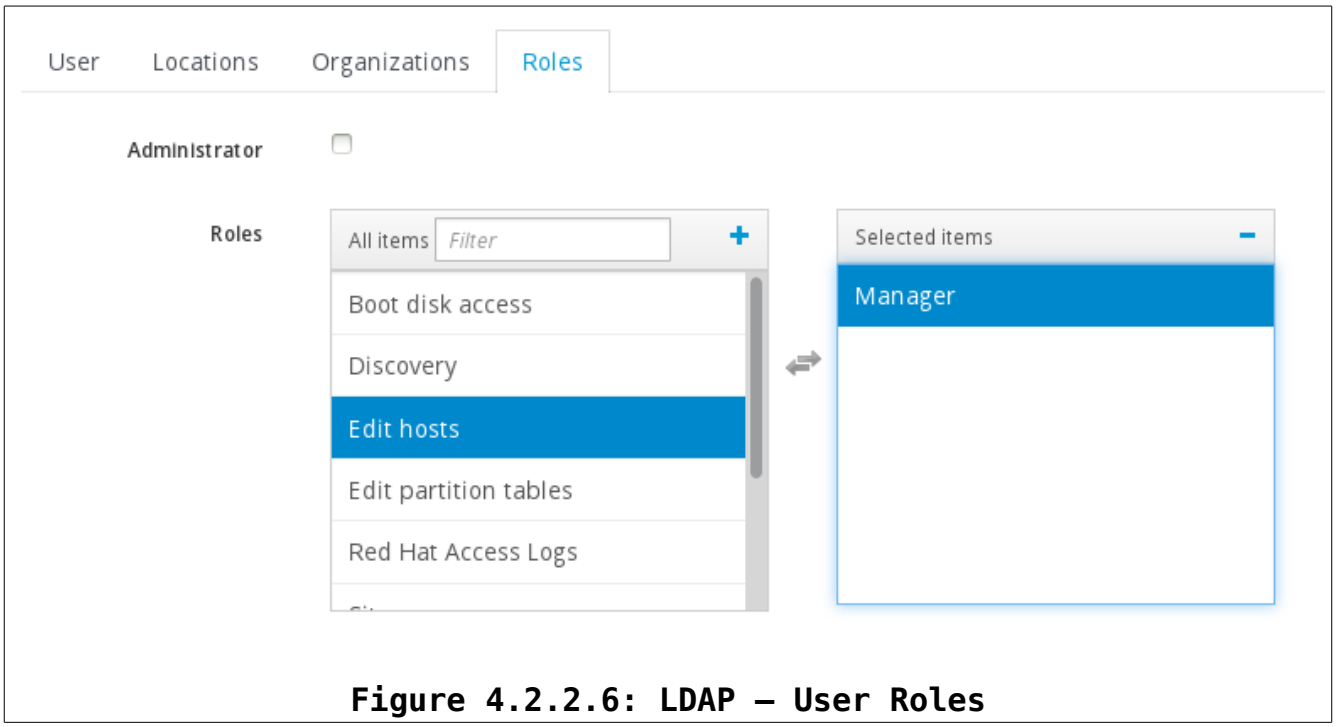


Figure 4.2.2.6: LDAP – User Roles

Click **Submit** to save the changes.

Permissions for the default roles can be modified by navigating to **Administer -> Roles**. For the *Manager* role, the following permissions are assigned under the **Filters** tab:

Resource Type	Permission
Miscellaneous	access_dashboard
	configuration
Host Group	view_hostgroup
	create_hostgroup
	edit_hostgroup
	destroy_hostgroup
Content Host	view_content_hosts
	create_content_hosts
	edit_content_hosts
	destroy_content_hosts
Content Views	view_content_views
	create_content_views
	edit_content_views
	destroy_content_views
	publish_content_views
	promote_or_remove_content_views

Table 4.2.2-1: Manager Role Permissions

Under the **Organizations** *Systems Engineering* is selected.

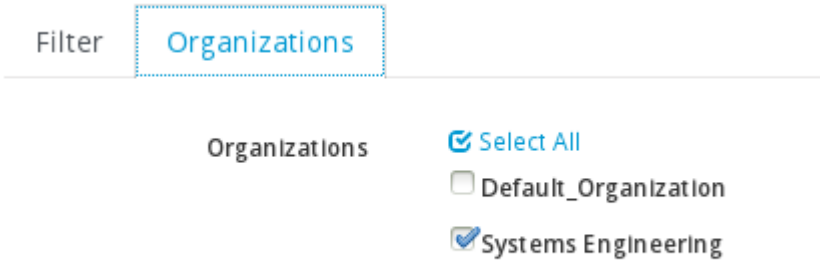


Figure 4.2.2.7: Manager Role Organizations

Click **Submit** to save changes.

Once the *test* user and role are configured, login to the Satellite 6 user interface with the *test* user to verify the permissions. Hover over the **Content** tab and note only **Content Views** and **Content Search** are available.

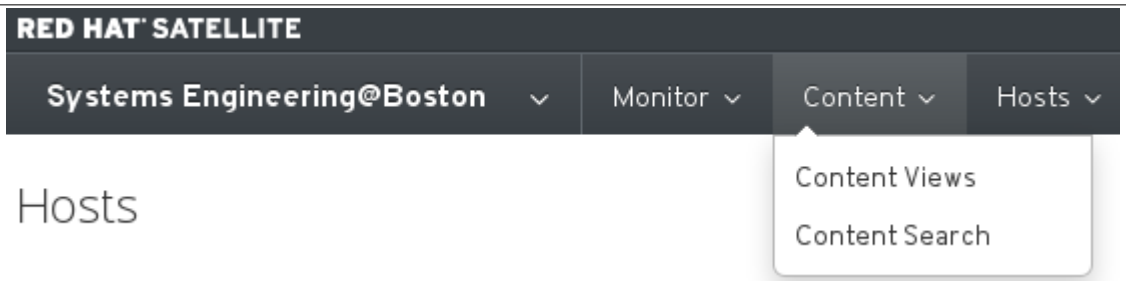


Figure 4.2.2.8: *test* User Permissions

4.3 Organization, Location and Lifecycle Environments

For the reference environment a single organization, location, and multiple lifecycle environments are created.

4.3.1 Organization and Location Creation

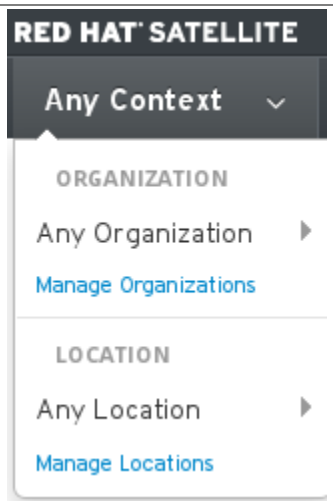


Figure 4.3.1.1: Create Organization

As the a*dmin* user, login to the Satellite 6 user interface and navigate to **Any Context** and select **Manage Organizations**.

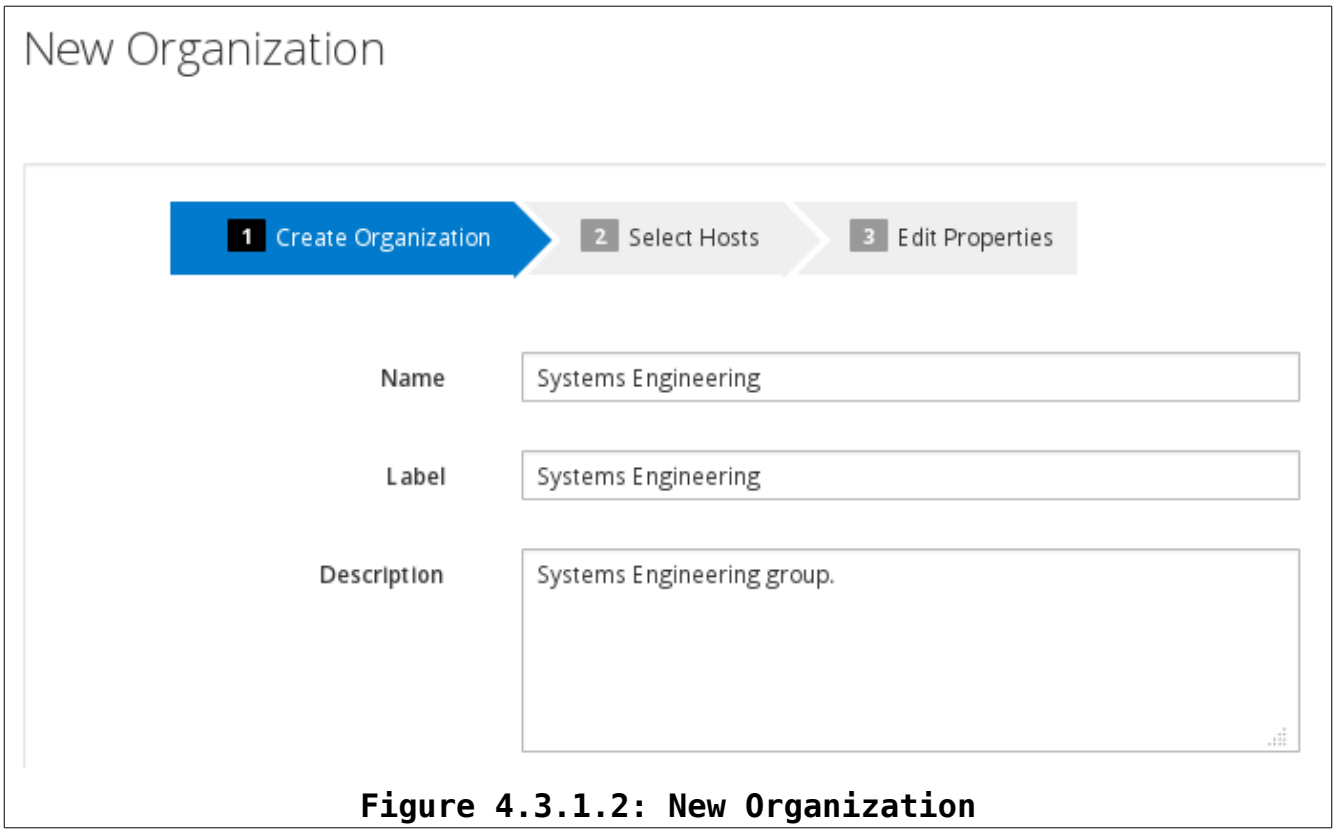


Figure 4.3.1.2: New Organization

Assign hosts as desired and/or click **Proceed to Edit**. Within **Edit Properties**, provide the desired inputs for each entry. For the reference environment the following entries are provided:

Setting	Value
Users	Clark Jones
Capsules	(integrated capsule; FQDN of Sat 6 server)
Subnets	(configured subnet for network)
Compute Resources	N/A
Media	RHEL 6 and 7 Server kickstarts
Templates	Kickstart default PXELinux, Kickstart default iPXE, OSE, PXELinux global default, RHEL OSP, RHEV Hosted, RHEVM Hosted, Satellite Finish Default, Satellite Kickstart Default, Satellite User Data Default, freeipa_register, idm_register, puppet.conf, subscription_manager_registration
Domains	(DNS domain for reference environment)
Realms	N/A
Puppet Environments	All minus *production*
Host Groups	Broker, Nodes, RHEL OSP 6, RHEV 3.5, RHEV 3.5 Manager
Locations	N/A
Parameters	N/A

Table 4.3.1-1: Organization Settings

Click **Submit** when finished.

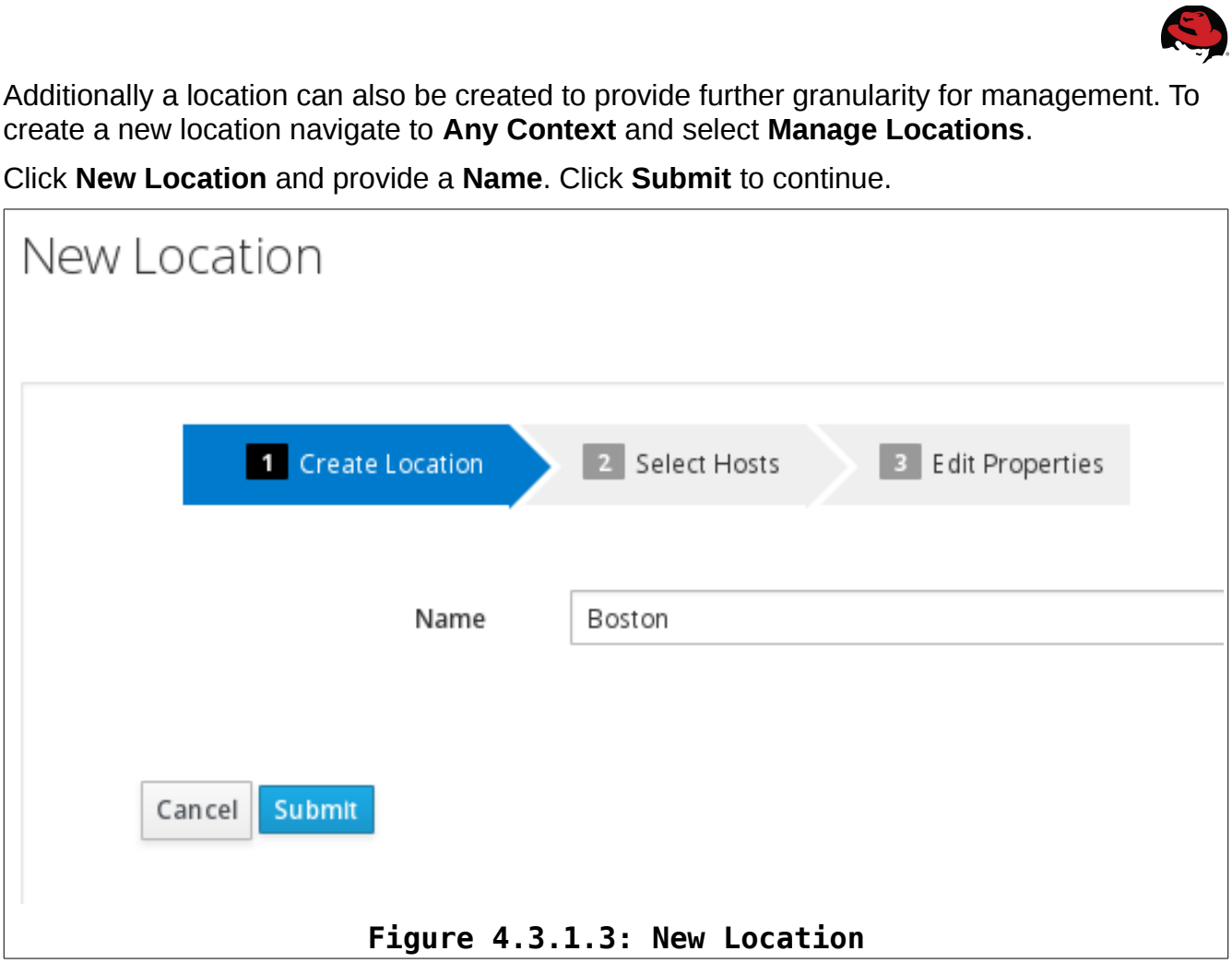


Figure 4.3.1.3: New Location

Additionally a location can also be created to provide further granularity for management. To create a new location navigate to **Any Context** and select **Manage Locations**.

Click **New Location** and provide a **Name**. Click **Submit** to continue.

Choose to assign hosts or or **Proceed to Edit**.

Within **Edit Properties**, provide the desired inputs for each entry. For the reference environment the following entries are provided:

Setting	Value
Users	N/A
Capsules	(integrated capsule; FQDN of Sat 6 server)
Subnets	(configured subnet for network)
Compute Resources	N/A
Media	RHEL 6 and 7 Server kickstarts
Templates	N/A
Domains	(DNS domain for reference environment)
Realms	N/A
Puppet Environments	All minus *production*
Host Groups	Broker, Nodes, RHEL OSP 6, RHEV 3.5, RHEV 3.5 Manager
Organizations	Systems Engineering
Parameters	N/A

Table 4.3.1-2: Location Settings

Click **Submit** when finished.

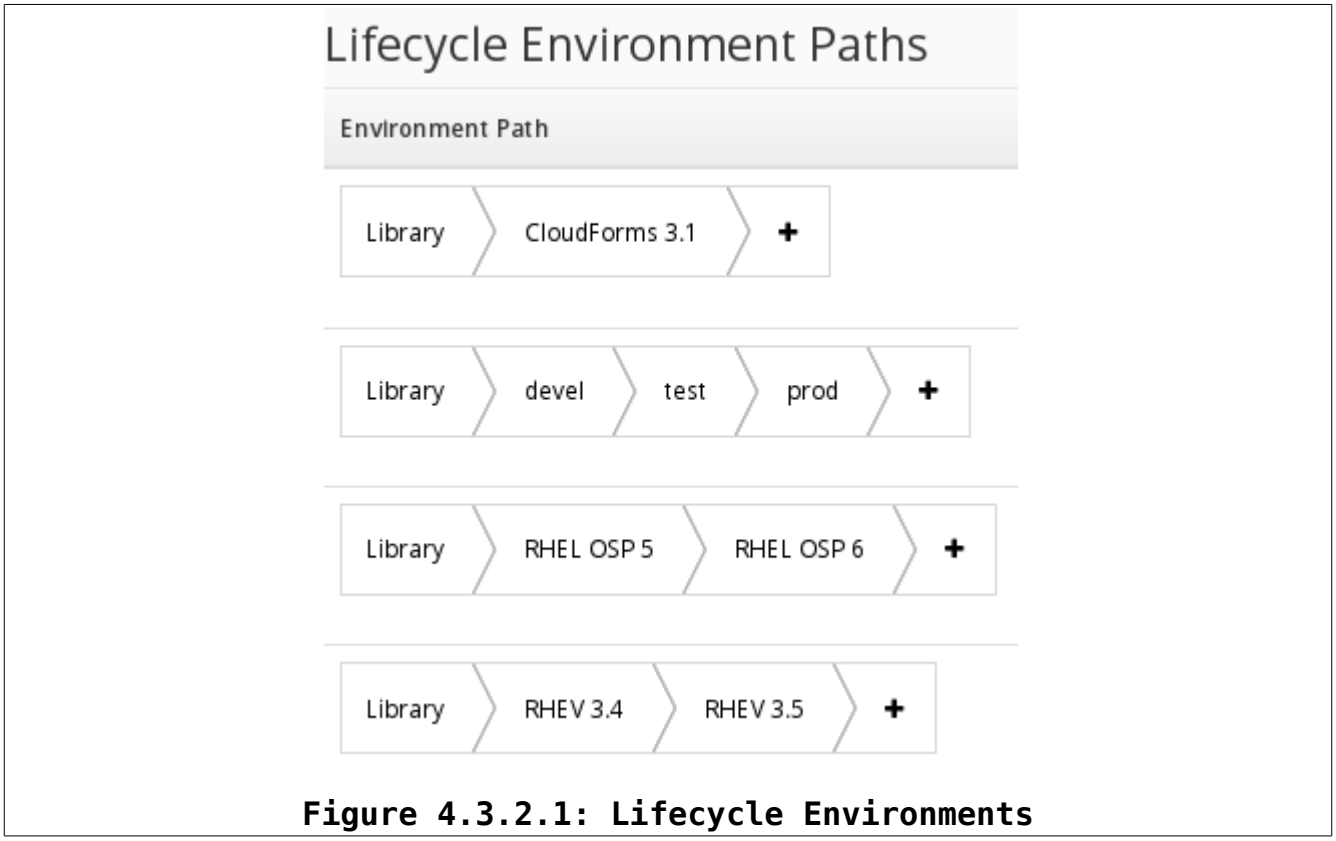


Figure 4.3.2.1: Lifecycle Environments

4.3.2 Lifecycle Environment Creation

Lifecycle Environments[3] are important to promote various content versions throughout the managed environment. For the reference environment the following lifecycle environment paths are created:

3 https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.0/html/User_Guide/sect-Lifecycle_Environments.html

To create a new Lifecycle Environment navigate to **Content** > **Lifecycle Environments.** Click on **New Environment Path**. Under **Create Environment**, provide a **Name**, **Label**, and **Description**.

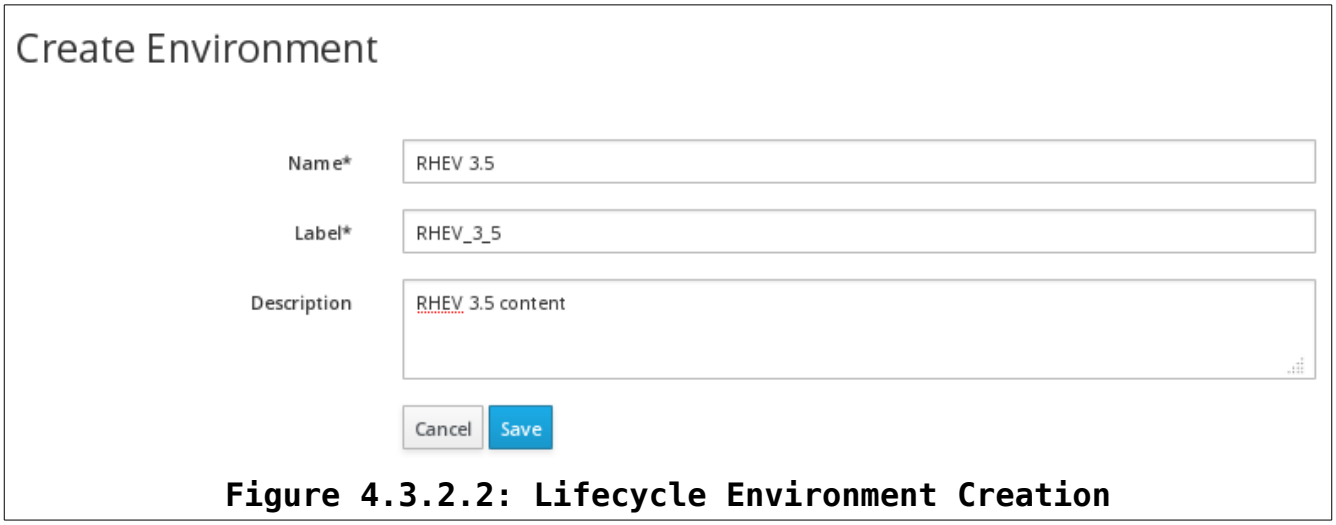


Figure 4.3.2.2: Lifecycle Environment Creation

Click **Save** to complete. Continue completing the necessary lifecycle environment paths.

4.4 Manifest Creation and Import

Create a manifest for the necessary Red Hat subscriptions. Log into the Red Hat Customer Portal at https://access.redhat.com. Navigate to **My Subscriptions > Subscription Management Applications > Satellite**.

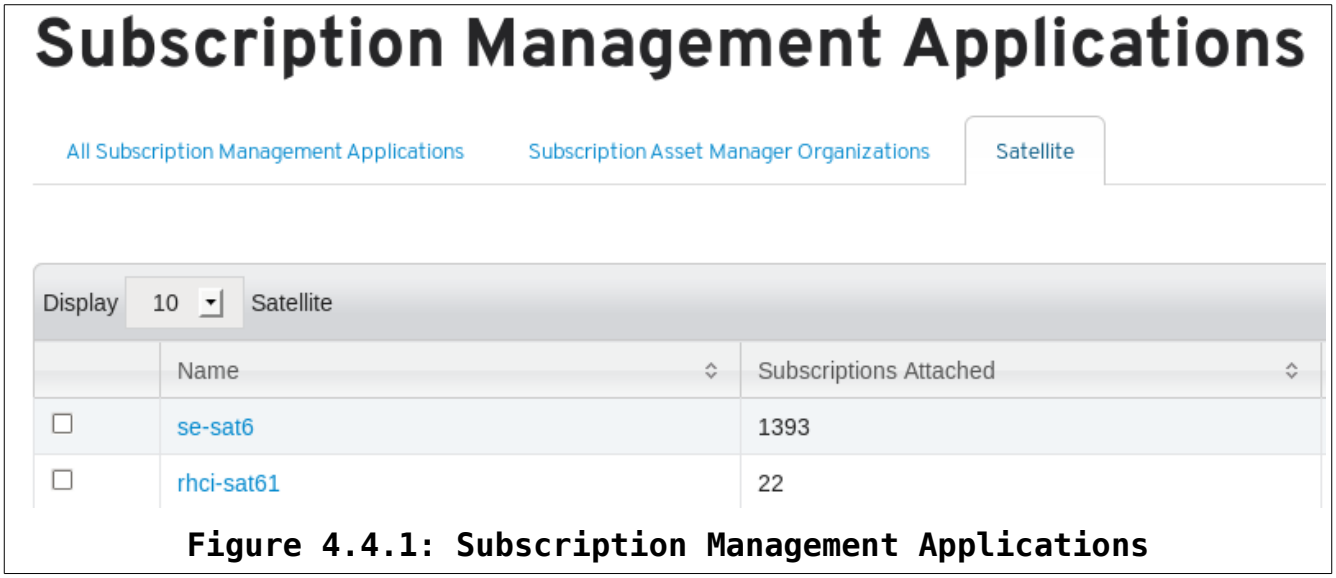


Figure 4.4.1: Subscription Management Applications

Click on **Register a Satellite**, provide a **Name** and select *Satellite 6.0* from the drop down list. Click **Register** to continue.

Figure 4.4.2: Register Satellite Server

On the right click **Attach a Subscription**. From the list select the appropriate subscriptions. For the reference environment the following subscriptions are chosen:

Subscription	Quantity
Red Hat Enterprise Linux Server, Standard (8 sockets) (Unlimited guests)	35
Red Hat Enterprise Linux Server, Standard (1 Virtual Machine up to 8 vCPUs)	20
Red Hat Cloud Infrastructure, Standard (8-sockets)	10
*OpenShift Employee Subscription	10

Table 4.4-1: Subscriptions

*Refer to *OpenShift Enterprise 2 Deployment Guide: Installing and Configuring OpenShift Enterprise*[4] for necessary subscriptions.

With the desired subscriptions attached, click **Download Manifest**. Save the *manifest.zip* to a desired location.

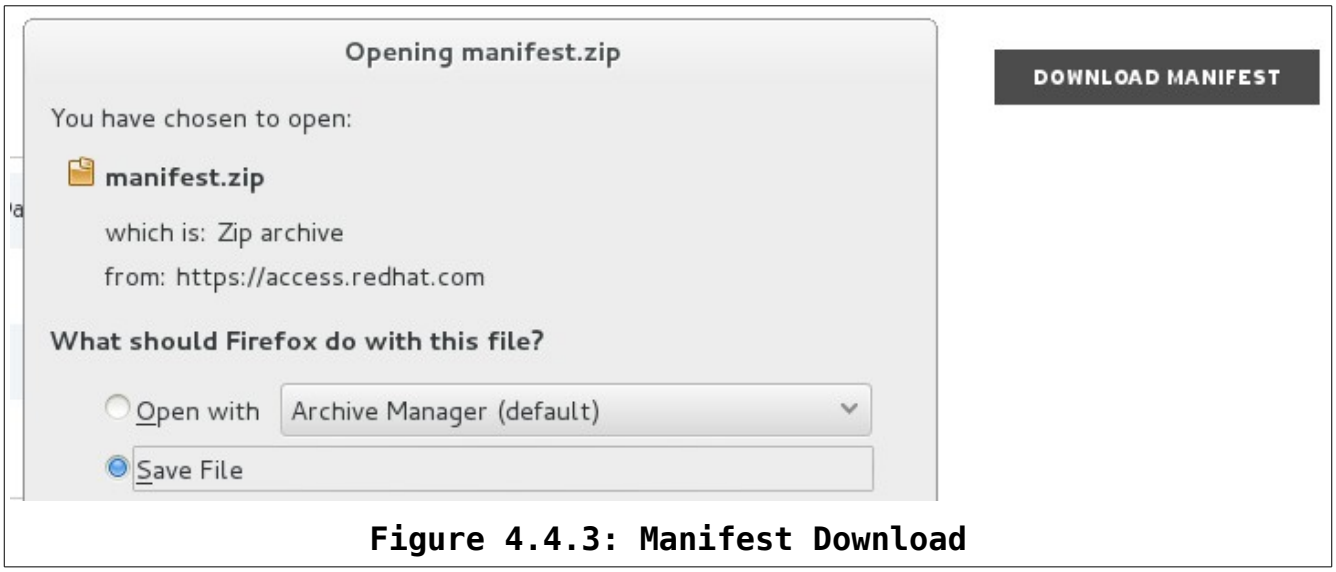


Figure 4.4.3: Manifest Download

4 https://access.redhat.com/documentation/en-US/OpenShift_Enterprise/2/html/Deployment_Guide/Red_Hat_Subscription_Requirements.html

Once the manifest is created with the proper subscriptions attached, the next step is importing into the Satellite server.

As the a*dmin* user, login to the Satellite 6 user interface and select the appropriate *Organization* and *Location.* Navigate to **Content** > **Red Hat Subscriptions**. Under **Upload New Manifest** browse to the location of the downloaded manifest. To complete click the **Upload** button.

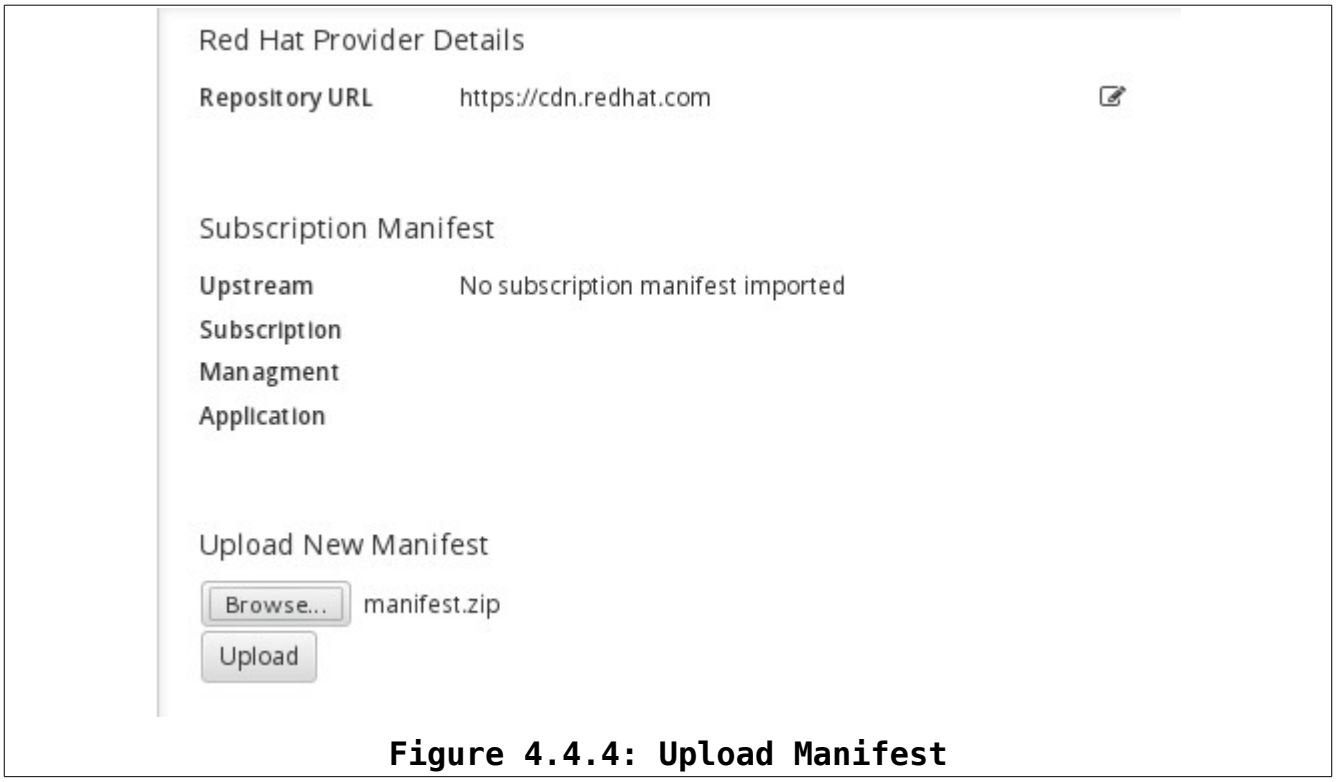


Figure 4.4.4: Upload Manifest

4.5 Repository, Product, and Content Management

The reference environment synchronizes a set of repositories and custom content based on OpenShift Origin puppet modules. The following Red Hat repositories are selected for synchronization.

Repository	Product Requirement
Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server	RHEV, OpenShift, CloudForms
Red Hat Enterprise Linux 6 Server - Supplementary RPMs x86_64 6Server	RHEV
Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.6	RHEV, OpenShift
Red Hat Enterprise Virtualization Manager 3.5 RPMs x86_64 6Server	RHEV

Repository	Product Requirement
Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 6Server	Satellite, RHEV, OpenShift
JBoss Enterprise Application Platform 6 RHEL 6 Server RPMs x86_64 6Server	RHEV, OpenShift
JBoss Enterprise Application Platform 6 RHEL 7 Server RPMs x86_64 7Server	RHEV
Red Hat Enterprise Virtualization Hypervisor RPMs x86_64 6Server	RHEV
Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 Infrastructure RPMs x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 Application Node RPMs x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 Client Tools RPMs x86_64 6Server	OpenShift
JBoss Enterprise Web Server 2 RHEL 6 Server RPMs x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 JBoss EAP add-on RPMs x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 JBoss A-MQ add-on RPMs x86_64 6Server	OpenShift
Red Hat OpenShift Enterprise 2.2 JBoss FUSE add-on RPMs x86_64 6Server	OpenShift
Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server	Satellite, RHEV, RHEL OSP
Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1	RHEV, RHEL OSP
Red Hat OpenStack 6.0 for RHEL 7 Platform Installer RPMs x86_64 7Server	RHEL OSP
Red Hat OpenStack 6.0 for RHEL 7 RPMs x86_64 7Server	RHEL OSP
Red Hat CloudForms Management Engine 5.3 RPMs x86_64	CloudForms
Red Hat Enterprise Linux 7 Server - RH Common RPMs x86_64 7Server	Satellite, RHEV, RHEL OSP
Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64 7Server	RHEL OSP

Repository	Product Requirement
Red Hat Enterprise Linux 7 Server - Optional RPMs x86_64 7Server	RHEV
Red Hat Enterprise Linux 7 Server - Supplementary RPMs x86_64 7Server	RHEV
Red Hat Enterprise Virtualization Agents for RHEL 6 Server RPMs x86_64 6Server	RHEV
Red Hat Enterprise Virtualization Hypervisor 7 RPMs x86_64 7Server	RHEV
Red Hat Enterprise Virtualization Management Agents for RHEL 7 RPMs x86_64 7Server	RHEV
Red Hat Enterprise Virtualization Management Agents RPMs x86_64 6Server	RHEV
Red Hat CloudForms Management Engine RPMs x86_64	CloudForms

Table 4.5-1: Red Hat Repositories

4.5.1 Red Hat Repository Selection and Synchronization

To select the necessary Red Hat repositories, navigate to **Content** > **Red Hat Repositories**. Under each tab, expand the necessary product and place a check next to the desired version.

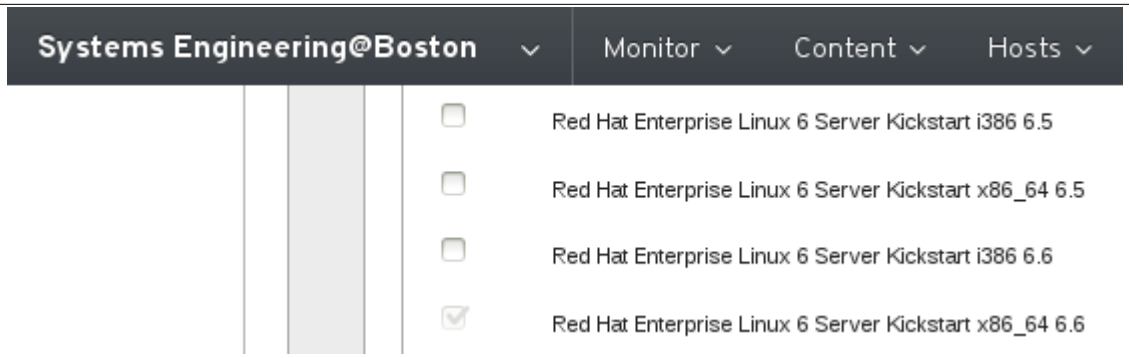


Figure 4.5.1.1: Red Hat Repository Selection

Note: *6Server* or *7Server* of a repository contains the latest version of that particular repo. This is desirable if a specific version of a product is not desired. When selecting a kickstart repository, choose a specific version of the operating system.

Once the needed Red Hat Repositories are selected, repository synchronization downloads the content to the Satellite server. Navigate to **Content** > **Sync Status**, click **Select All** and click the **Synchronize Now** button. A status bar appears indicating the progress of the synchronization. It may take several hours to complete the initial content sync.

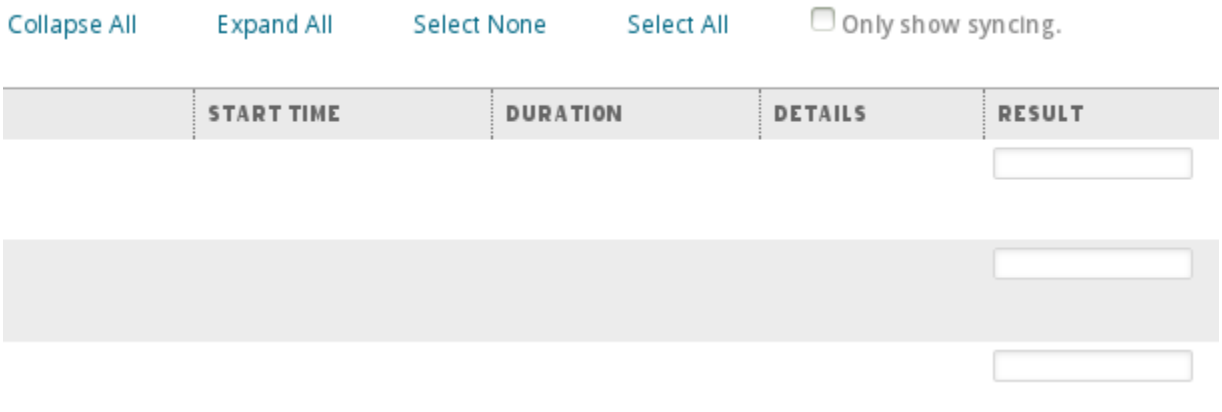


Figure 4.5.1.2: Red Hat Repository Sync

Additionally a sync plan can be created to run the repository sync on a predetermined schedule. To create a sync plan navigate to **Content** > **Sync Plans** and click **New Sync Plan**.

Provide a **Name**, **Interval**, **Start Date**, and **Start Time**.

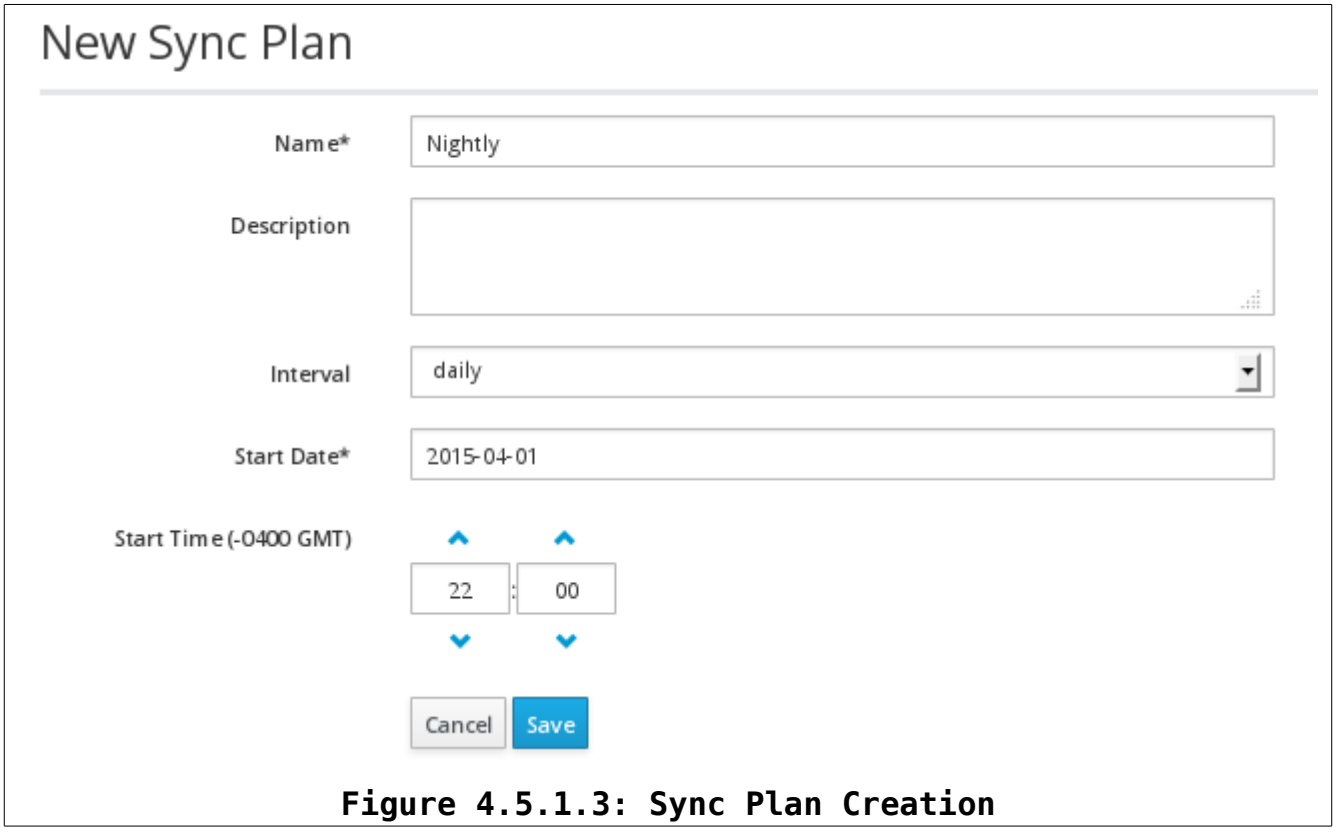


Figure 4.5.1.3: Sync Plan Creation

Click **Save** to complete.

Add products to the newly created sync plan by navigating to **Content** > **Products**. Place a check mark next to **Name** to select all and click the **Bulk Actions** button.

Under *Bulk Actions* select **Alter Sync Plans**. Select the sync plan listed and click the **Update Sync Plan** button.

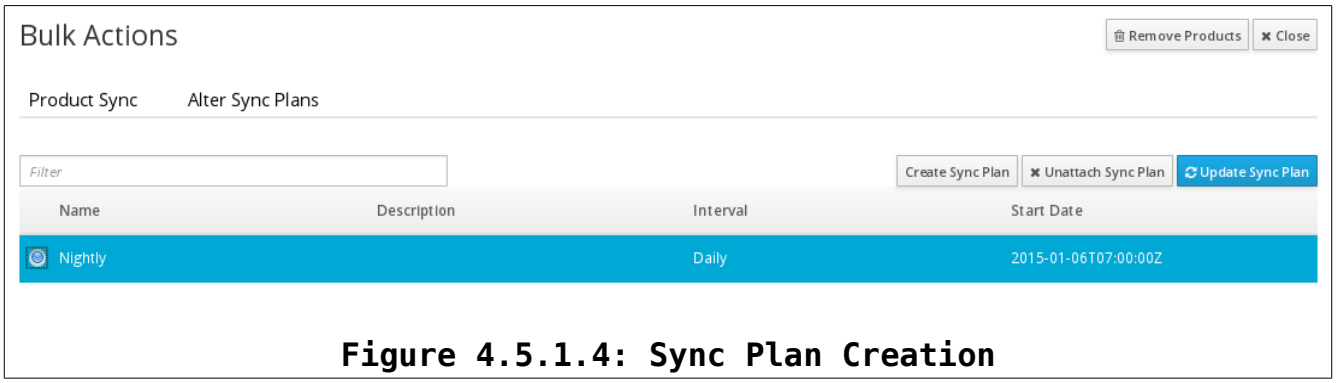


Figure 4.5.1.4: Sync Plan Creation

4.5.2 Custom Product Creation

For the reference environment a custom product is created for the OpenShift Origin puppet modules.

To create a custom product repository, navigate to **Content** > **Products** and click **New Products**. Provide a **Name** and **Label**. Optionally, a **Sync Plan** can be selected.

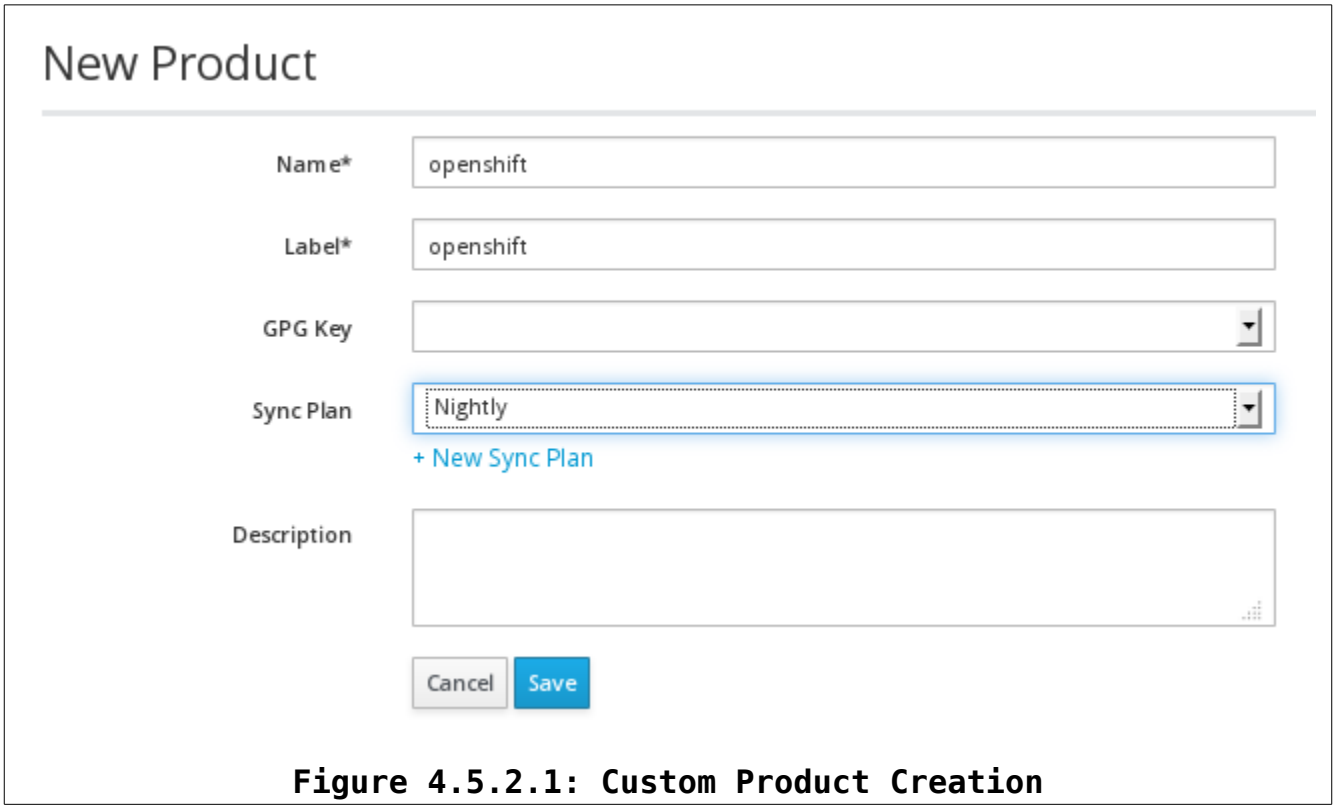


Figure 4.5.2.1: Custom Product Creation

Click **Save** to complete.

Next create a repository by selecting the newly created product and clicking the **Repository** tab. Click **Create Repository** and provide a **Name**, **Label**, and **Type**. Ensure a check mark is placed next to **Publish via HTTP**.

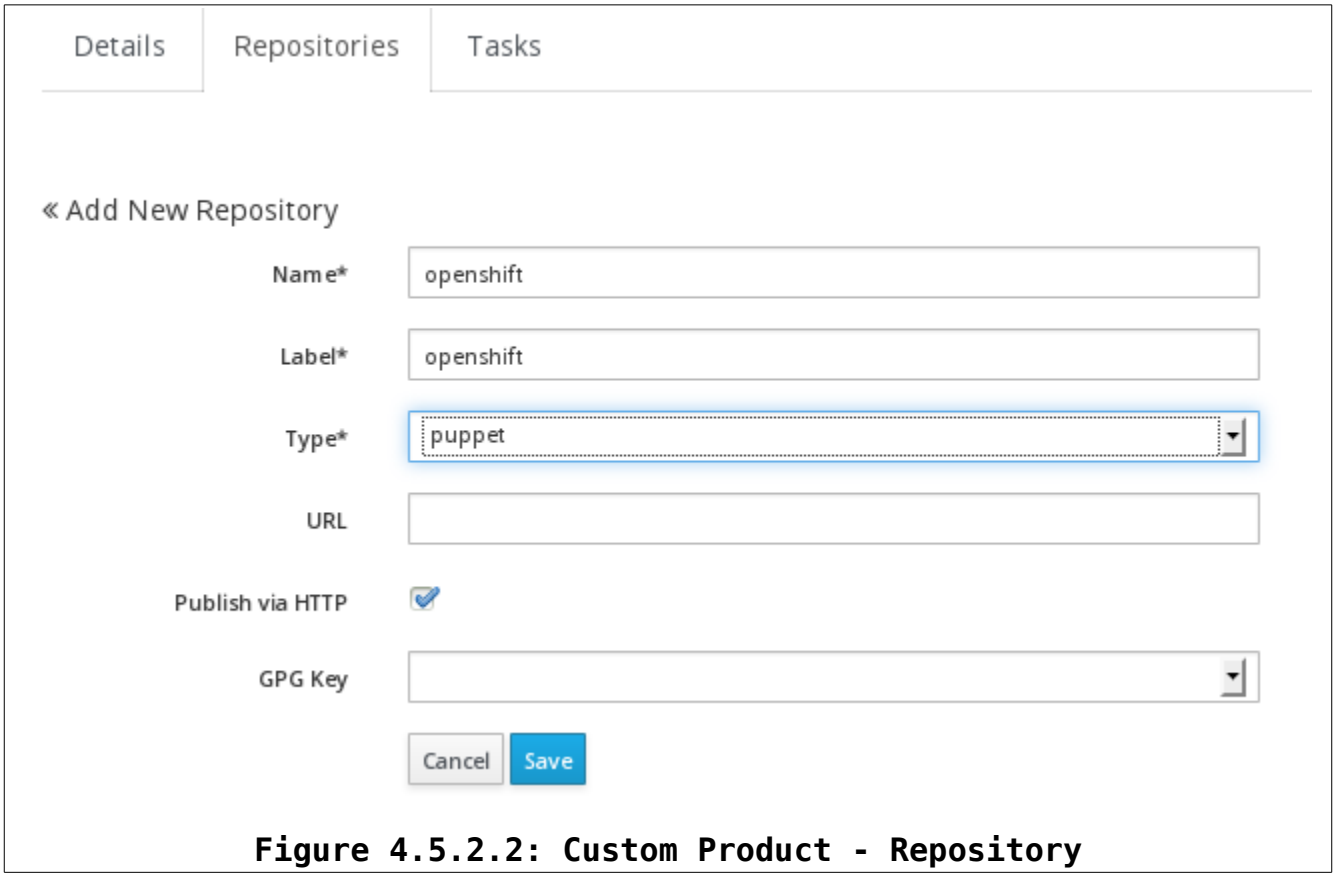


Figure 4.5.2.2: Custom Product - Repository

Click **Save** to continue.

Note: In a typical scenario a URL path is provided for the custom product and it's associated repo are synced to import content. Due to the following bugs:

https://bugzilla.redhat.com/show_bug.cgi?id=1146916

https://bugzilla.redhat.com/show_bug.cgi?id=1167788

...it is necessary to perform the following tasks:

1. Edit vi */etc/puppet/rack/config.ru* and add the following to the file:

```
#Ensure UTF-8 is our default (sadly Ruby 1.9 sets to US-ASCII)
Encoding.default_external = Encoding::UTF_8 if defined? Encoding
```

Upon completion restart the necessary services:

```
#katello-service restart
#systemctl restart puppet
```

2. Use the *big_hammer.sh* **Appendix E: Scripts and Configuration Files** script to download and import the OpenShift Origin puppet modules.

```
#./big_hammer.sh
```

The mentioned issues are to be resolved in a future Satellite 6 release.

Upon completing the download and import of the OpenShift Origin puppet modules, ten puppet modules are listed within the custom *openshift* product, repo, content.

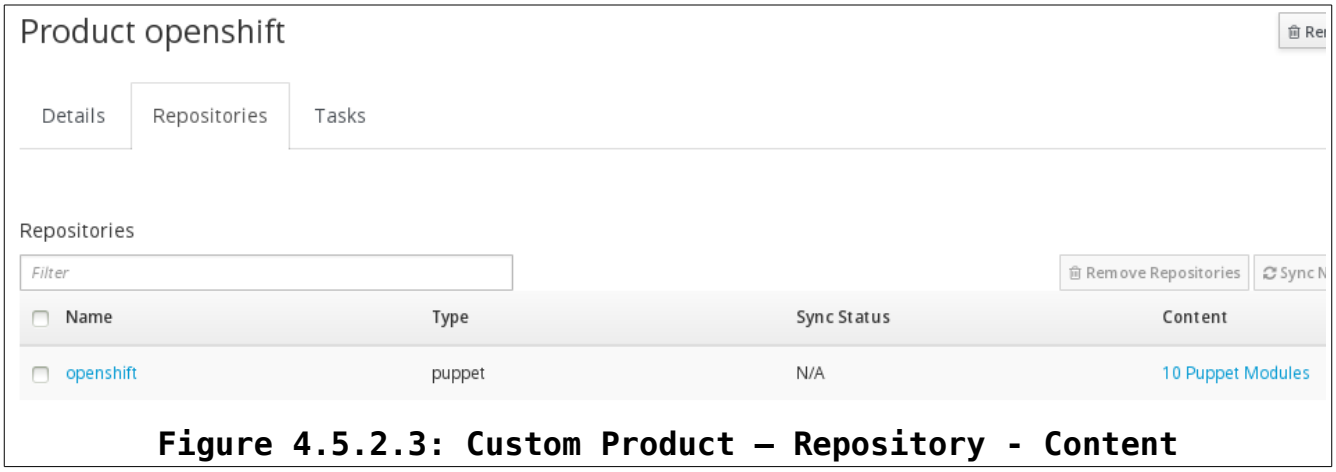


Figure 4.5.2.3: Custom Product – Repository - Content

4.5.3 Content View Creation and Promotion

With repositories and custom products synchronized and created, the next step is to create content views and promote to the desired lifecycle environments.

To create a content view, navigate to **Content** > **Content Views** and click **Create New View**. Provide a **Name**, **Label**, and click **Save** to complete.

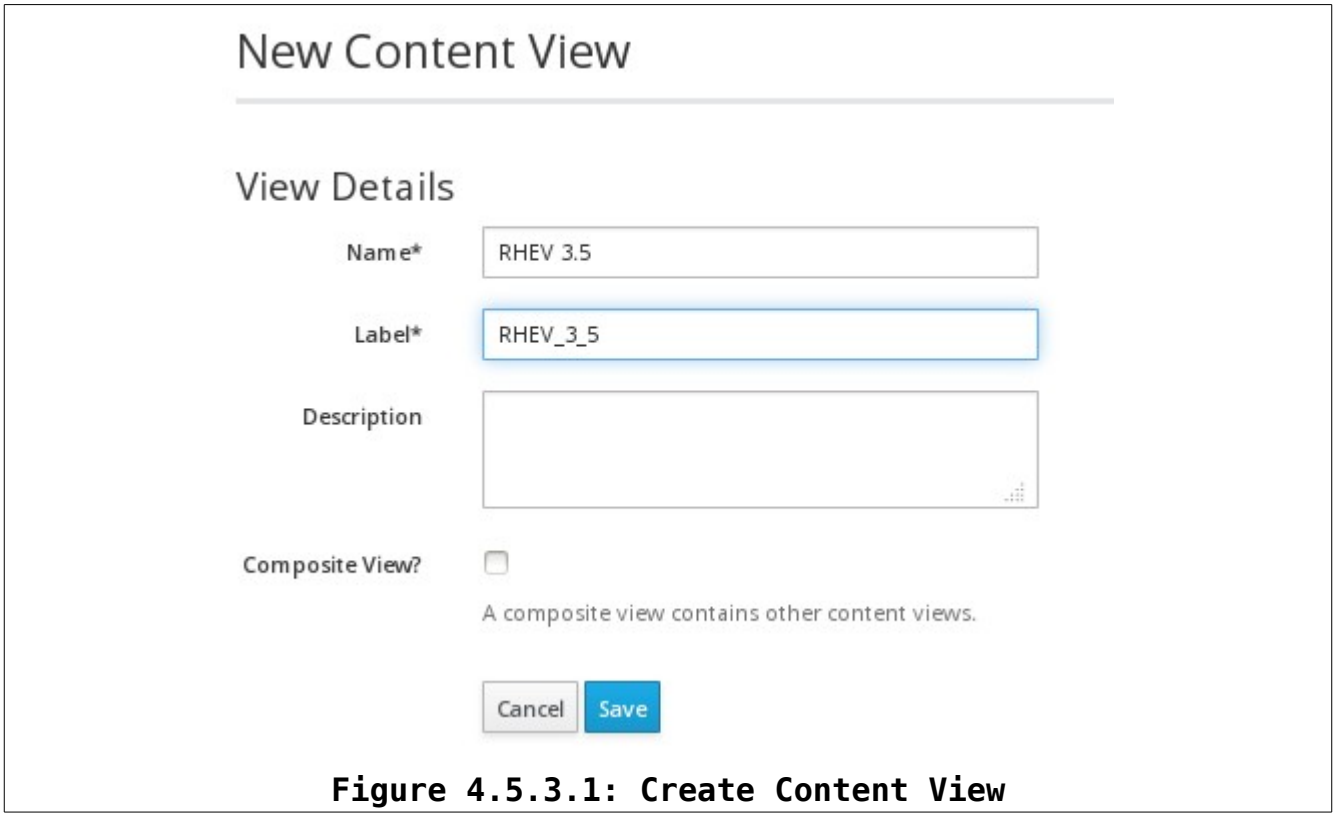


Figure 4.5.3.1: Create Content View

Select the newly created content view, click the **Content** tab, and choose **Add** to add the needed repositories. For the reference environment the following content views are created with associated repositories and puppet modules:

Content View	Repository
RHEV 3.5	JBoss Enterprise Application Platform 6 RHEL 6 Server RPMs x86_64 6Server
	JBoss Enterprise Application Platform 6 RHEL 7 Server RPMs x86_64 7Server
	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.6
	Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 6Server
	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
	Red Hat Enterprise Linux 6 Server - Supplementary RPMs x86_64 6Server
	Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1
	Red Hat Enterprise Linux 7 Server - Optional RPMs x86_64 7Server
	Red Hat Enterprise Linux 7 Server - RH Common RPMs x86_64 7Server
	Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server
	Red Hat Enterprise Linux 7 Server - Supplementary RPMs x86_64 7Server
	Red Hat Enterprise Virtualization Agents for RHEL 6 Server RPMs x86_64 6Server
	Red Hat Enterprise Virtualization Hypervisor 7 RPMs x86_64 7Server
	Red Hat Enterprise Virtualization Hypervisor RPMs x86_64 6Server
	Red Hat Enterprise Virtualization Management Agents for RHEL 7 RPMs x86_64 7Server
	Red Hat Enterprise Virtualization Management Agents RPMs x86_64 6Server
	Red Hat Enterprise Virtualization Manager 3.5 RPMs x86_64 6Server

Content View	Repository
RHEL OSP 6	Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1
	Red Hat Enterprise Linux 7 Server - RH Common RPMs x86_64 7Server
	Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server
	Red Hat OpenStack 6.0 for RHEL 7 Platform Installer RPMs x86_64 7Server
	Red Hat OpenStack 6.0 for RHEL 7 RPMs x86_64 7Server
	Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64 7Server
RHEL 6 w/OSE	JBoss Enterprise Application Platform 6 RHEL 6 Server RPMs x86_64 6Server
	JBoss Enterprise Web Server 2 RHEL 6 Server RPMs x86_64 6Server
	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.6
	Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 6Server
	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 Application Node RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 Client Tools RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 Infrastructure RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 JBoss A-MQ add-on RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 JBoss EAP add-on RPMs x86_64 6Server
	Red Hat OpenShift Enterprise 2.2 JBoss FUSE add-on RPMs x86_64 6Server
	Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server x86_64 6Server
CloudForms	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server
	Red Hat CloudForms Management Engine RPMs x86_64

Content View	Puppet Module
RHEL 6 w/OSE	haproxy
	ntp
	stdlib
	sysctl
	java_ks
	lokkit
	selinux_types
	concat
	keepalived
	openshift_origin

Table 4.5.3-1: Content Views – Repos and Puppet Modules

With the repositories and puppet modules chosen, publish the content by clicking **Publish New Version**. Enter a comment if desired and click **Save** to begin the content view publication.

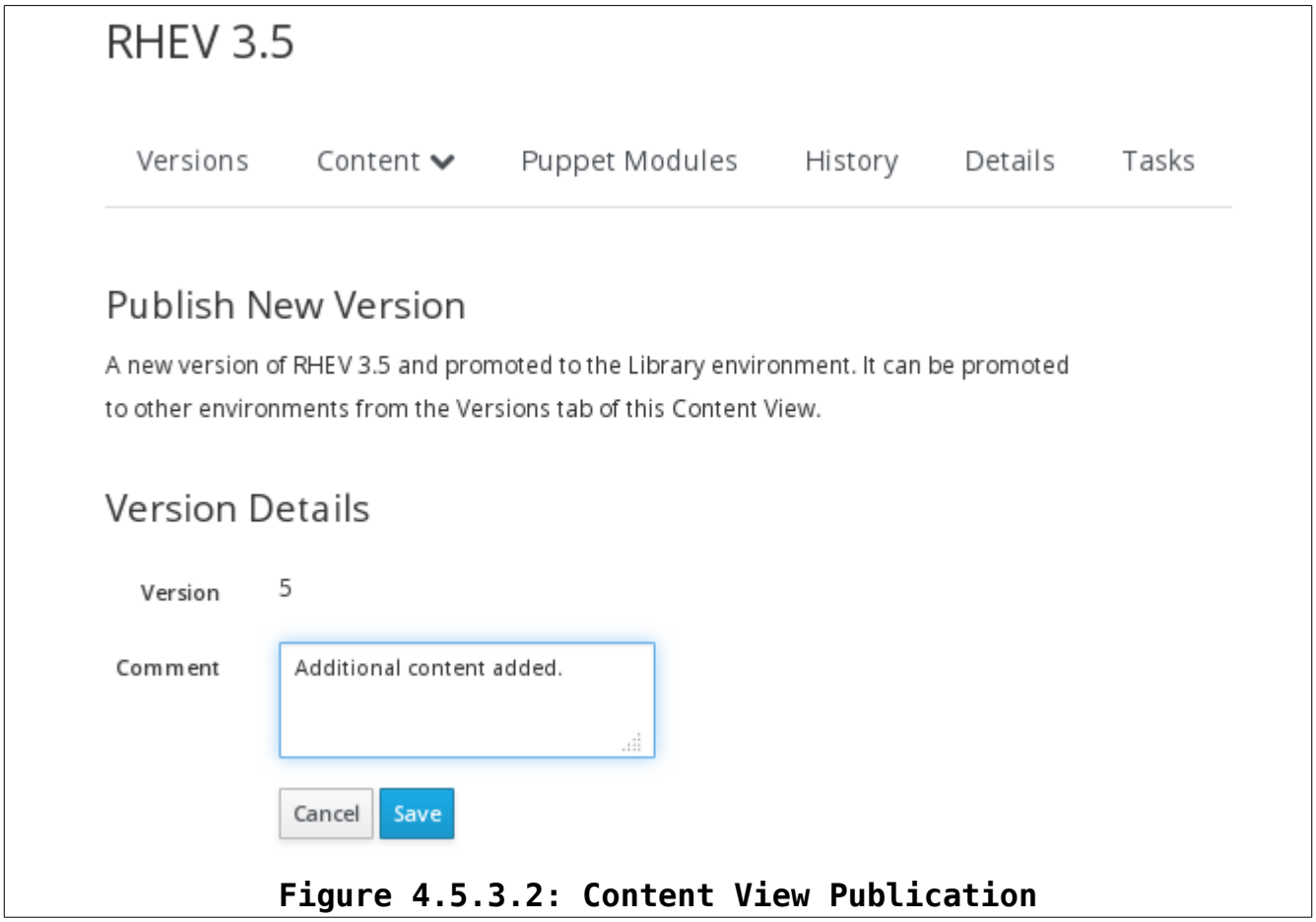


Figure 4.5.3.2: Content View Publication

Promote the newly published content to a desired lifecycle environment by clicking **Promote** for the newly published content.

Select the appropriate lifecycle environment and click **Promote Version**.

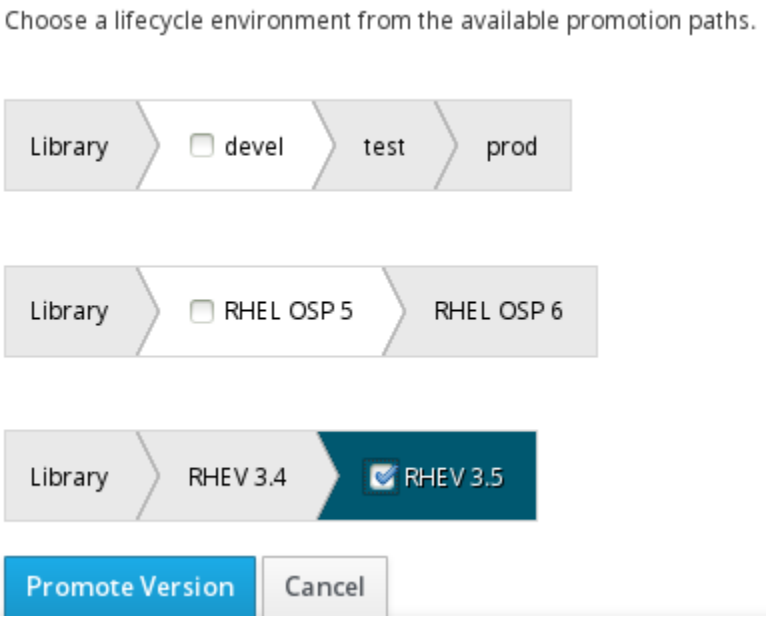


Figure 4.5.3.3: Content View Promotion

Perform the same steps for the each content view created.

4.6 Activation Keys

Activation keys allow for system registration to selected Red Hat repositories and products. For the reference environment, the following activation keys are created:

Activation key	Subscriptions	Environment	Content View
RHEL 6 w/OSE	OpenShift Employee Subscription	devel	RHEL 6 w/OSE
	Red Hat Enterprise Linux Server, Standard (1 Virtual Machine up to 8 vCPUs)		
	openshift		
RHEL OSP	Red Hat Cloud Infrastructure, Standard (8-sockets)	RHEL OSP 6	RHEL OSP 6
	Red Hat Enterprise Linux Server, Standard (8 sockets) (Unlimited guests)		

Activation key	Subscriptions	Environment	Content View
RHEV Hypervisor	Red Hat Cloud Infrastructure, Standard (8-sockets)	RHEV 3.5	RHEV 3.5
	Red Hat Enterprise Linux Server, Standard (8 sockets) (Unlimited guests)		
RHEV Manager	Red Hat Cloud Infrastructure, Standard (8-sockets)	RHEV 3.5	RHEV 3.5
	Red Hat Enterprise Linux Server, Standard (1 Virtual Machine up to 8 vCPUs)		
CloudForms	Red Hat Cloud Infrastructure, Standard (8-sockets)	CloudForms 3.1	CloudForms
	Red Hat Enterprise Linux Server, Standard (1 Virtual Machine up to 8 vCPUs)		

Table 4.6-1: Activation Keys – Subscriptions, Products, and Environments

To create an activation key navigate to **Content** > **Activation keys** and click **New Activation Key**. Provide a **Name**, determine **Content Host Limit**, **Description**, select the appropriate lifecycle **Environment**, and choose a **Content View** associated with the lifecycle **Environment**. Click **Save** to complete.

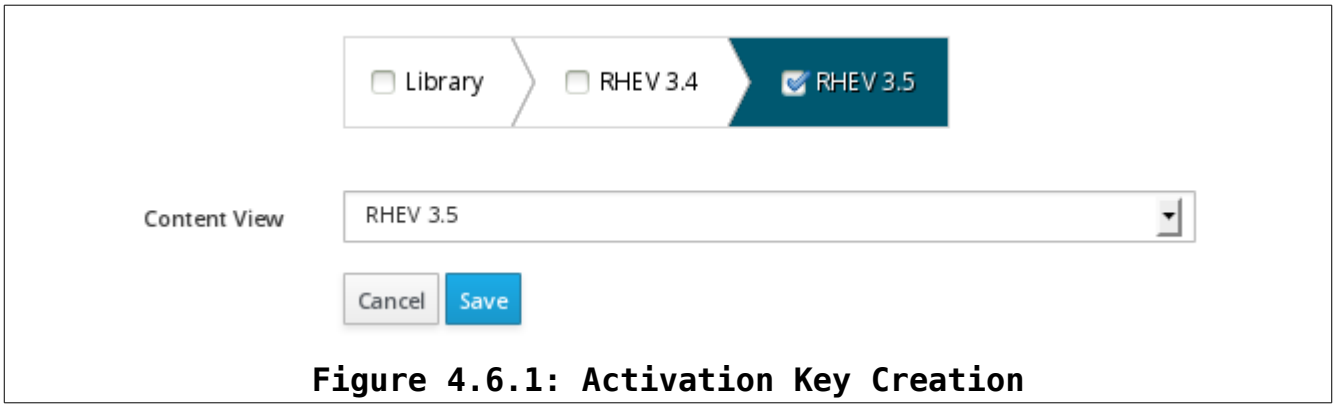


Figure 4.6.1: Activation Key Creation

Once a content view is created and subscriptions attached, the next step is to enable the **Product Content**. Within the activation key, select the **Product Content** tab and set **Enabled** option to *Yes* for the desired items. For the reference environment the following products are enabled for each activation key:

Activation key	Enabled Products
RHEL 6 w/OSE	Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server
	Red Hat OpenShift Enterprise 2.2 JBoss EAP add-on (RPMs)
	Red Hat OpenShift Enterprise 2.2 JBoss FUSE add-on (RPMs)
	Red Hat OpenShift Enterprise 2.2 Application Node (RPMs)
	Red Hat OpenShift Enterprise 2.2 Client Tools (RPMs)
	JBoss Enterprise Web Server 2 (RHEL 6 Server) (RPMs)
	JBoss Enterprise Application Platform 6 (RHEL 6 Server) (RPMs)
	Red Hat Enterprise Linux 6 Server (Kickstart)
	Red Hat Enterprise Linux 6 Server - RH Common (RPMs)
	Red Hat Enterprise Linux 6 Server (RPMs)
	Red Hat OpenShift Enterprise 2.2 Infrastructure (RPMs)
	openshift
	Red Hat OpenShift Enterprise 2.2 JBoss A-MQ add-on (RPMs)
RHEL OSP	Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server
	Red Hat Enterprise Linux 7 Server (Kickstart)
	Red Hat Enterprise Linux 7 Server - RH Common (RPMs)
	Red Hat Enterprise Linux 7 Server (RPMs)
	Red Hat OpenStack 6.0 for RHEL 7 (RPMs)
	Red Hat OpenStack 6.0 for RHEL 7 Platform Installer (RPMs)
RHEV Hypervisor	Red Hat Enterprise Virtualization Manager 3.5 (RPMs)
	Red Hat Enterprise Virtualization Hypervisor 7 (RPMs)
	Red Hat Enterprise Virtualization Management Agents for RHEL 7 (RPMs)
	Red Hat Enterprise Linux 7 Server - Optional (RPMs)
	Red Hat Enterprise Linux 7 Server (Kickstart)
	Red Hat Enterprise Linux 7 Server - RH Common (RPMs)
	Red Hat Enterprise Linux 7 Server (RPMs)

Activation key	Enabled Products
RHEV Hypervisor	Red Hat Enterprise Linux 7 Server - Supplementary (RPMs)
RHEV Manager	Red Hat Enterprise Virtualization Hypervisor (RPMs)
	Red Hat Enterprise Virtualization Manager 3.5 (RPMs)
	Red Hat Enterprise Virtualization Management Agents (RPMs)
	JBoss Enterprise Application Platform 6 (RHEL 6 Server) (RPMs)
	Red Hat Enterprise Virtualization Agents for RHEL 6 Server (RPMs)
	Red Hat Enterprise Linux 6 Server (Kickstart)
	Red Hat Enterprise Linux 6 Server - RH Common (RPMs)
	Red Hat Enterprise Linux 6 Server (RPMs)
	Red Hat Enterprise Linux 6 Server - Supplementary (RPMs)
CloudForms	Red Hat CloudForms Management Engine 5.3 (RPMs)
	Red Hat Enterprise Linux 6 Server (RPMs)

Table 4.6-2: Activation Keys – Enabled Products

4.7 Host Groups

Host groups provide a template-like function when deploying hosts with Satellite 6. This reduces the amount of time to deploy hosts that require similar settings. For the reference environment the following host groups are created:

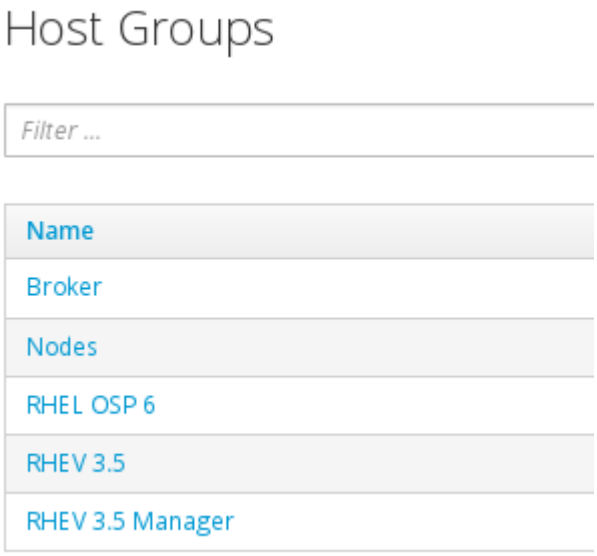


Figure 4.7.1: Host Groups

To create a new host group, navigate to **Configure** > **Host groups** and click on **New Host Group**. For the reference environment the following settings for each host group are configured:

Host Group	Tab	Option	Value
Broker	Host Group	Name	Broker
		Lifecycle Environment	devel
		Puppet Environment	RHEL_6_w_OSE
		Content Source	rhci-sat6.refarch.bos.redhat.com
		Puppet CA	rhci-sat6.refarch.bos.redhat.com
		Puppet Master	rhci-sat6.refarch.bos.redhat.com
	Puppet Classes	Included Classes	openshift_origin
	Network	Domain	refarch.bos.redhat.com
		Subnet	syseng(10.19.10.0/23)
	Operating System	Architecture	x86_64
		Operating system	RHEL Server 6.6
		Media	RHSE/Library/Red_Hat_6_Server _Kickstart_x86_64_6_6
		Partition table	Kickstart default
		Root password	<REDACTED>
	Parameters	All	Defaults
	Locations	Selected items	Boston
	Organizations	Selected items	Systems Engineering
	Activation Keys	Activation keys	RHEL 6 w/OSE
Nodes	Host Group	Name	Nodes
		Lifecycle Environment	devel
		Puppet Environment	RHEL_6_w_OSE
		Content Source	rhci-sat6.refarch.bos.redhat.com
		Puppet CA	rhci-sat6.refarch.bos.redhat.com
		Puppet Master	rhci-sat6.refarch.bos.redhat.com
	Puppet Classes	Included Classes	openshift_origin
	Network	Domain	refarch.bos.redhat.com
		Subnet	syseng(10.19.10.0/23)

Host Group	Tab	Option	Value
Nodes	Operating System	Architecture	x86_64
		Operating system	RHEL Server 6.6
		Media	RHSE/Library/Red_Hat_6_Server _Kickstart_x86_64_6_6
		Partition table	Kickstart default
		Root password	<REDACTED>
	Parameters	All	Defaults
	Locations	Selected items	Boston
	Organizations	Selected items	Systems Engineering
	Activation Keys	Activation keys	RHEL 6 w/OSE
RHEL OSP 6	Host Group	Name	RHEL OSP 6
		Lifecycle Environment	RHEL_OSE_6
		Puppet Environment	RHEL_OSE_6
		Content Source	rhci-sat6.refarch.bos.redhat.com
		Puppet CA	rhci-sat6.refarch.bos.redhat.com
		Puppet Master	rhci-sat6.refarch.bos.redhat.com
	Puppet Classes	N/A	N/A
	Network	Domain	refarch.bos.redhat.com
		Subnet	syseng(10.19.10.0/23)
	Operating System	Architecture	x86_64
		Operating system	RedHat 7.1
		Media	RHSE/Library/Red_Hat_7_Server _Kickstart_x86_64_7_1
		Partition table	Kickstart default
		Root password	<REDACTED>
	Parameters	N/A	N/A
	Locations	Selected items	Boston
	Organizations	Selected items	Systems Engineering
	Activation Keys	Activation keys	RHEL OSP
RHEV 3.5	Host Group	Name	RHEV 3.5
		Lifecycle Environment	RHEV_3_5
		Puppet Environment	RHEV_3_5

Host Group	Tab	Option	Value
RHEV 3.5	Host Group	Content Source	RHEV_3_5
		Puppet CA	rhci-sat6.refarch.bos.redhat.com
		Puppet Master	rhci-sat6.refarch.bos.redhat.com
	Puppet Classes	Included Classes	rhci-sat6.refarch.bos.redhat.com
	Network	Domain	refarch.bos.redhat.com
		Subnet	syseng(10.19.10.0/23)
	Operating System	Architecture	x86_64
		Operating system	RedHat 7.1
		Media	RHSE/Library/Red_Hat_7_Server _Kickstart_x86_64_7_1
		Partition table	Kickstart default
		Root password	<REDACTED>
	Parameters	N/A	N/A
	Locations	Selected items	Boston
	Organizations	Selected items	Systems Engineering
	Activation Keys	Activation keys	RHEV Hypervisor
RHEV 3.5 Manager	Host Group	Name	RHEV 3.5 Manager
		Lifecycle Environment	RHEV_3_5
		Puppet Environment	RHEV_3_5
		Content Source	RHEV_3_5
		Puppet CA	rhci-sat6.refarch.bos.redhat.com
		Puppet Master	rhci-sat6.refarch.bos.redhat.com
	Puppet Classes	Included Classes	rhci-sat6.refarch.bos.redhat.com
	Network	Domain	refarch.bos.redhat.com
		Subnet	syseng(10.19.10.0/23)
	Operating System	Architecture	x86_64
		Operating system	RHEL Server 6.6
		Media	RHSE/Library/Red_Hat_6_Server _Kickstart_x86_64_6_6
		Partition table	Kickstart default
		Root password	<REDACTED>

Host Group	Tab	Option	Value
RHEV 3.5 Manager	Parameters	N/A	N/A
	Locations	Selected items	Boston
	Organizations	Selected items	Systems Engineering
	Activation Keys	Activation keys	RHEV Manager

Table 4.7-1: Host Group Settings

Click **Submit** to complete each host group creation.

4.8 Provisioning Templates

Provisioning templates provide a way to customize host deployment with Satellite 6. This includes kickstarts and snippets. For the reference environment the following custom provisioning templates are used for the defined host groups:

Host Group	Provisioning Template
RHEV 3.5 Manager	RHEVM Hosted
RHEV 3.5	RHEV Hosted
RHEL OSP 6	RHEL OSP
Broker	OSE
Nodes	OSE

Table 4.8-1: Provisioning Templates

Specifics regarding the contents of each custom provisioning template are found in **Appendix F: kickstarts and snippets**.

To create a custom provisioning template navigate to **Hosts** > **Provisioning templates**. Either click **New Template** or **Clone** an existing template. For the reference environment the *Satellite Kickstart Default* template is cloned and a custom provisioning template is modified.

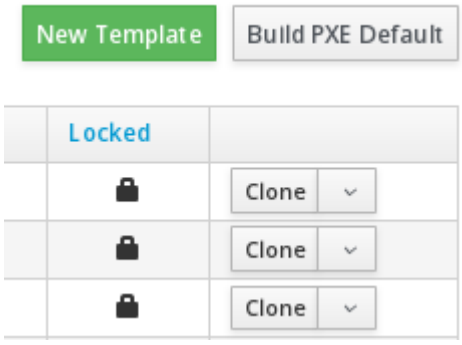


Figure 4.8.1: Provisioning Templates

Each custom provisioning template is associated to **Applicable Operating Systems**, **Host Group**, and **Environment**.

To create the associations, within the custom provisioning template, select the **Association** tab. For **Applicable Operating Systems** select the desired operating system to move under **Selected items**. Click **Add combination** to assign the needed **Host Group** and **Environment**. For the reference environment the following associations are made for each custom provisioning template:

Provisioning Template	Associations	Setting
RHEVM Hosted	Applicable Operating Systems	RHEL Server 6.6
	Host Group	RHEV 3.5 Manager
	Environment	KT_RHSE_RHEV_3_5_RHEV_3_5_9
RHEV Hosted	Applicable Operating Systems	RedHat 7.1
	Host Group	RHEV 3.5
	Environment	KT_RHSE_RHEV_3_5_RHEV_3_5_9
RHEL OSP	Applicable Operating Systems	RedHat 7.1
	Host Group	RHEL OSP 6
	Environment	N/A
OSE	Applicable Operating Systems	RHEL Server 6.6
	Host Group	Broker
	Environment	KT_RHSE_devel_RHEL_6_w_OSE_4
	Host Group	Nodes
	Environment	KT_RHSE_devel_RHEL_6_w_OSE_4

Table 4.8-2: Provisioning Template Associations

Click **Submit** to complete for each custom provision template.

Note: Type is set to *provision* for all custom provisioning templates.

5 Deploying Red Hat Enterprise Virtualization 3.5

For the reference environment a Red Hat Enterprise Virtualization self-hosted engine[5] configuration is deployed from Red Hat Satellite 6. Additionally, only the needed steps to deploy from Satellite 6 are outlined and references provided to existing product documentation for the remaining steps.

5.1 NFS Configuration

To support the self-hosted engine install, a dedicated NFS server is configured hosting an NFS share. This share provides storage space for the RHEV Manager virtual machine.

Specific NFS configuration options and iptables settings for *cf-vms* are located in **Appendix D: iptables and firewalld** and **Appendix E: Scripts and Configuration Files**.

5.2 Deploy RHEV Self-Hosted Engine

The self-hosted engine is deployed onto a bare metal machine. This is a requirement as this machine also serves as a hypervisor in the deployed RHEV environment.

As the *admin* user, select the previously created organization and location, and navigate to **Hosts** > **New host**. Provide a **Name**, select an **Organization**, select a **Location**, **Host Group**, and selection an option for **Deploy on.**

For the reference environment the following values are used:

Option	Value
Name	rhci-rhev
Organization	Systems Engineering
Location	Boston
Host Group	RHEV 3.5
Deploy on	Bare Metal

Table 5.2-1: Self-Hosted Engine Host Deployment Options

Upon selecting the **Host Group**, the previously defined host group values populate the various options.

5 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/html/Installation_Guide/chap-The_Self-Hosted_Engine.html

Click the **Network** tab and provide the **MAC address** for the host machine under **Primary Interface**. This is the network interface used to PXE boot and deploy the operating system on the system.

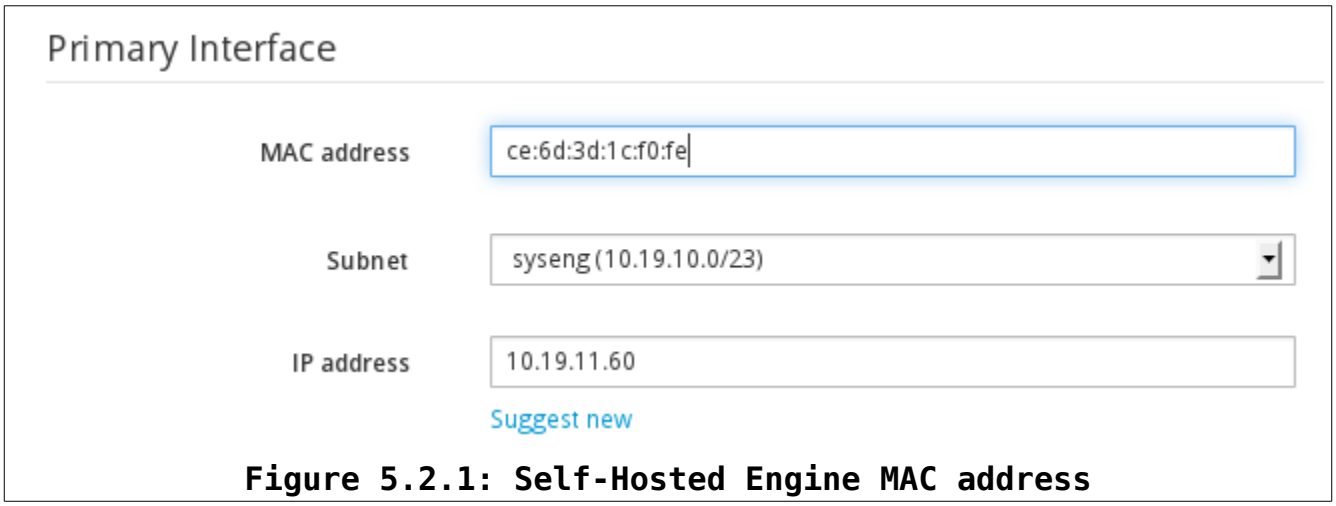


Figure 5.2.1: Self-Hosted Engine MAC address

On the **Operating System** tab verify the information listed and click **Resolve** to populate the **Provisioning templates**.

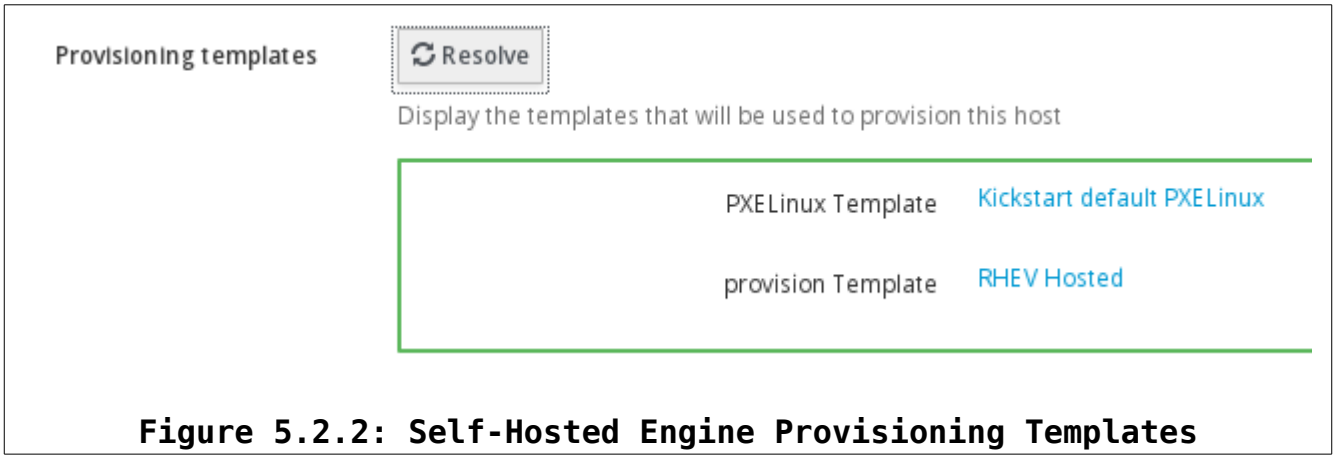


Figure 5.2.2: Self-Hosted Engine Provisioning Templates

With all remaining options verified, click **Submit** to begin the host deployment. Power on the bare metal machine to initiate the PXE boot process. Upon successful install, the host appears under **Hosts** > **All hosts** with a green status.

Figure 5.2.3: Self-Hosted Engine Deployment Status

Note: If issues arise during the deployment of a host, review the *install.post.log* file located under */root* on the host machine.

Login to the console of the deployed self-hosted engine host and perform the self-hosted engine install referring to the *Red Hat Enterprise Virtualization 3.5 Installation Guide*[6].

> **Note:** Configure the primary Ethernet interface of the self-hosted engine host with a static IP address prior to running `hosted-engine --deploy`.

The following specific values are provided during the self-hosted engine install:

Option	Value
Please specify the storage you would like to use	nfs3
Please specify the full shared storage connection path to use	*<path to nfs server/share>*
Please indicate a nic to set rhevm bridge on	eth1
Please specify the device to boot the VM from	pxe
You may specify a MAC address for the VM or accept a randomly generated default	Random (this MAC address is needed when deploying the RHEV VM from Satellite; please document)

Table 5.2-2: Self-Hosted Engine Install Options

> **Note:** Before confirming the installation settings at the *Configuration Preview* screen to continue the self-hosted engine install, use the assigned MAC address to create a new host within Satellite 6.

As the *admin* user, select the previously created organization and location, and navigate to **Hosts** > **New host**. Provide a **Name**, select an **Organization**, select a **Location**, **Host Group**, and selection an option for **Deploy on.**

For the reference environment the following values are used:

Option	Value
Name	rhci-rhevm
Organization	Systems Engineering
Location	Boston
Host Group	RHEV 3.5
Deploy on	Bare Metal

Table 5.2-3: RHEV Manager Host Deployment Options

Upon selecting the **Host Group**, the previously defined host group values populate the various options.

6 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/html/Installation_Guide/Installing_the_Self-Hosted_Engine.html

Click the **Network** tab and provide the **MAC address** for the host machine under **Primary Interface**. This is the previously captured MAC address which is used to PXE boot and deploy the operating system on the system.

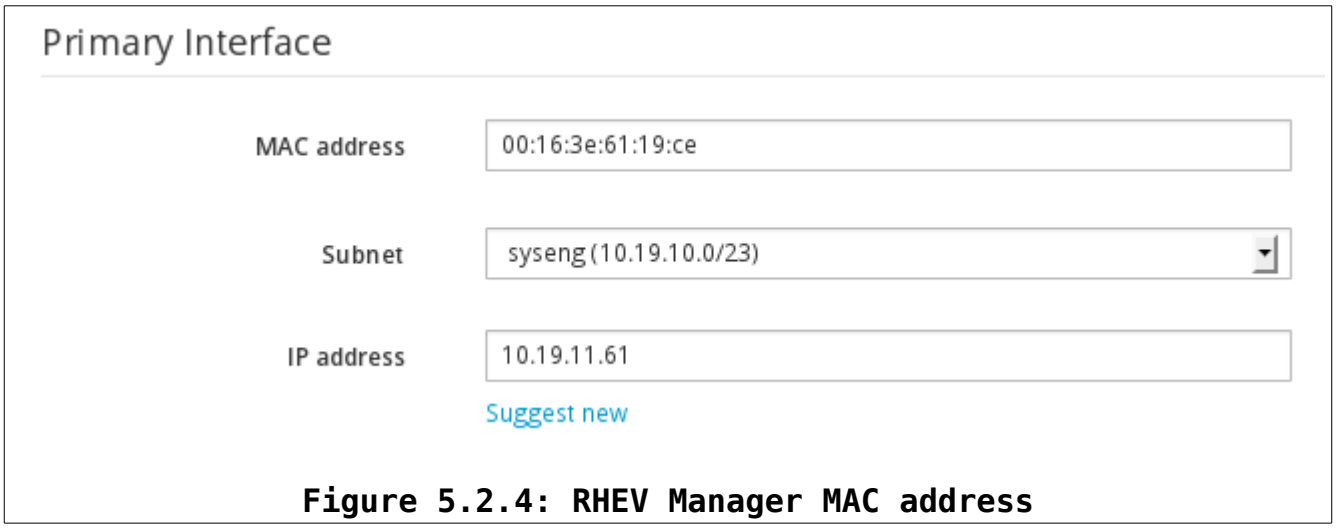


Figure 5.2.4: RHEV Manager MAC address

On the **Operating System** tab verify the information listed and click **Resolve** to populate the **Provisioning templates**.

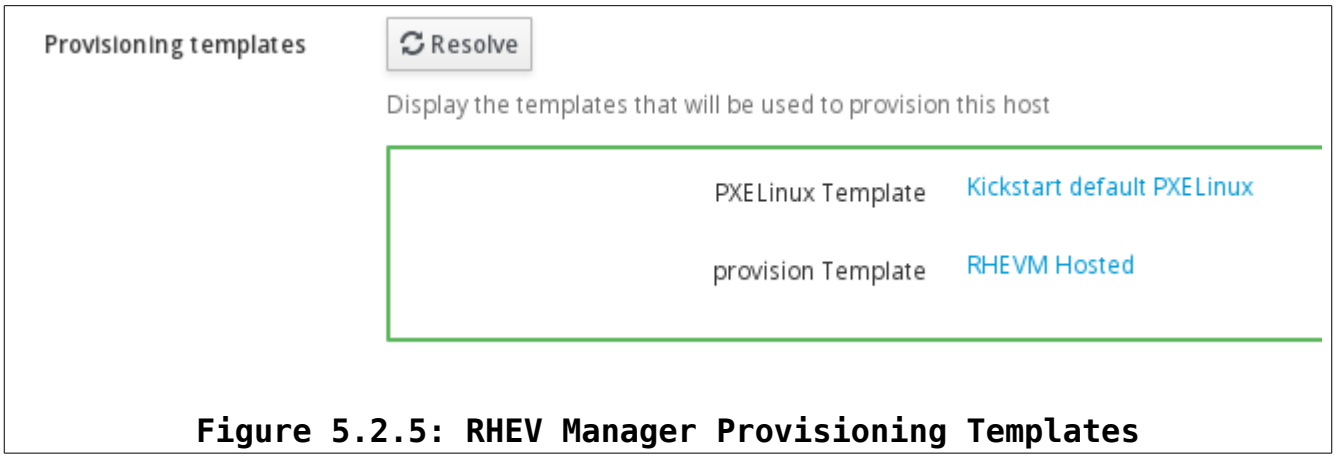


Figure 5.2.5: RHEV Manager Provisioning Templates

With all remaining options verified, click **Submit** to begin the RHEV Manager host deployment.

Continue the self-hosted engine install by confirming the installation settings at the *Configuration Preview* screen.

Complete the self-hosted engine install following the steps detailed in the *Red Hat Enterprise Virtualization 3.5 Installation Guide*[6]. Upon successful install, the host appears under **Hosts** > **All hosts** with a green status.

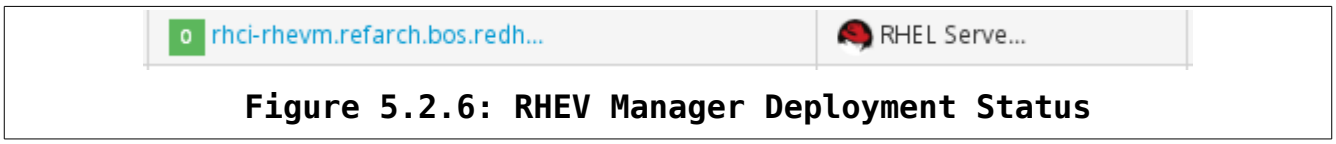


Figure 5.2.6: RHEV Manager Deployment Status

Upon successful self-hosted engine deployment, complete the configuration of the RHEV environment by adding the RHEV Hypervisor (self-hosted physical machine) and configuring a Data storage domain[7].

5.3 IdM Integration (optional)

Additionally RHEV supports integration with directory service providers. For the reference environment Red Hat Identity Management is used. Steps for configuring IdM as a directory service provider for RHEV are found in the *Red Hat Enterprise Virtualization 3.5 Administration Guide*[8].

For the reference environment IdM user *devel* (Clark Jones) is added as a RHEV user and assigned the *DataCenterAdmin* role.

User	Authorization provider	Namespace	Role
admin (admin@internal)	internal	*	SuperUser
admin (admin@internal)	internal	*	PowerUserRole
Clark Jones (devel@profile1-authz)	profile1-authz	dc=refarch,dc=bos,dc=redhat,...	DataCenterAdmin

Figure 5.3.1: IdM RHEV User

Reference configuration files are located in **Appendix E Scripts and Configuration Files**.

> **Note:** An IdM user may be required to change their password. To do so the user can login to the IdM user portal to reset.

5.4 Satellite 6 Compute Resource

With RHEV deployed and configured, add it as a compute resource to Satellite. As the **admin** user select the appropriate **Organization** and **Location**, navigate to **Infrastructure** > **Compute** resources, and click **New Compute Resource**.

7 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/html/Installation_Guide/part-Basic_Setup.html

8 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/html/Administration_Guide/sect-Directory_Users.html#Configuring_a_Generic_LDAP_Provider

Provide a **Name**, select a **Provider**, **URL** to the provider, a **Username**, a **Password**, and click **Load Datacenters**. For the reference environment the following values are used:

Option	Value
Name	RHEV3.5
Provider	RHEV
URL	https://rhci-rhevm.refarch.bos.redhat.com/api
Username	admin@internal
Password	[REDACTED]

Table 5.4-1: Compute Resource Values

Under the **Locations** and **Organizations** tab and choose the desired location and organization.

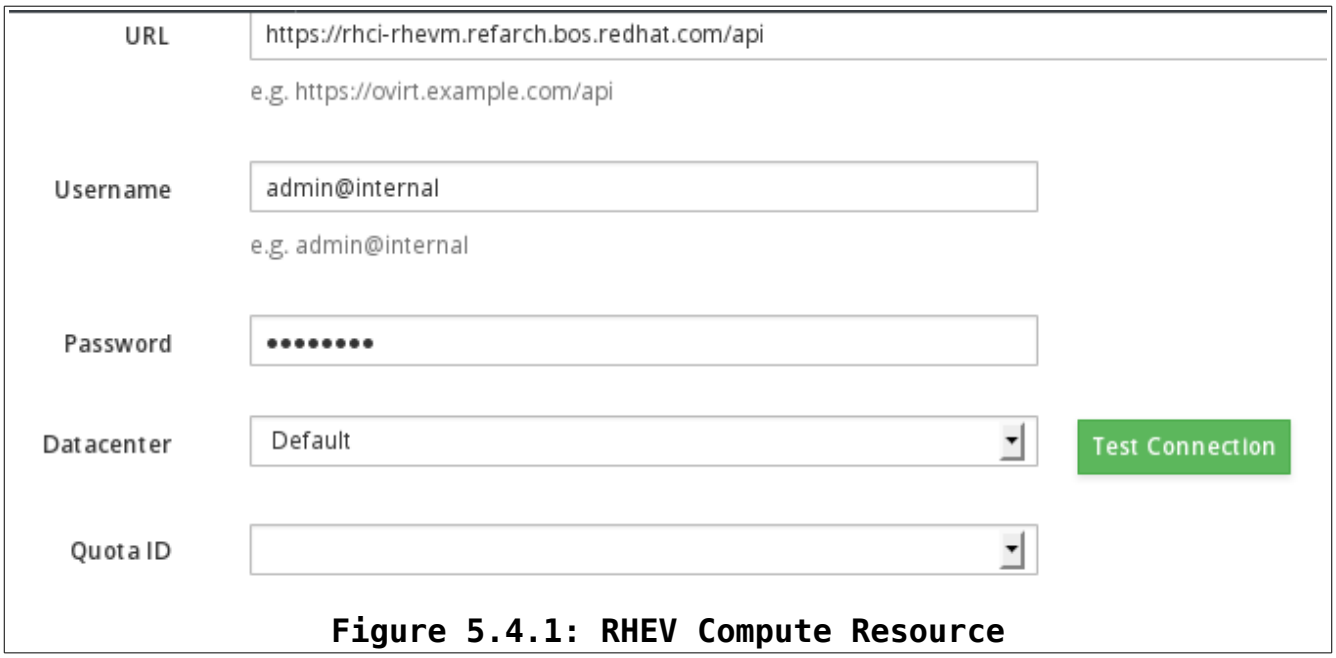


Figure 5.4.1: RHEV Compute Resource

Click **Submit** to complete.

6 Deploying Red Hat Enterprise Linux OpenStack Platform 6

For the reference environment, the Red Hat Enterprise Linux OpenStack Platform Installer[9] is deployed from Red Hat Satellite 6. Additionally, only the needed steps to deploy from Satellite 6 are outlined and references provided to existing product documentation and reference architectures for the remaining steps.

6.1 Deploy RHEL OSP Installer

The RHEL OSP Installer is deployed onto a bare metal machine and becomes a deployment host for the RHEL OSP environment. Due to this fact, configuration management (`puppet` and `katello-agent`) of this host from Satellite 6 is not permitted as it conflicts with the necessary configuration management required for the RHEL OSP Installer that is used to manage the deployed RHEL OSP hosts.

> **Note:** As part of the provisioning template customization, puppet and *katello-agent* installation and configuration are disabled. Refer to **Appendix F kickstarts and snippets** for additional details.

As the *admin* user, select the previously created organization and location, and navigate to **Hosts** > **New host**. Provide a **Name**, select an **Organization**, select a **Location**, select a **Host Group**, and select a option for **Deploy on.**

For the reference environment the following values are used:

Option	Value
Name	rhci-rhelosp-inst
Organization	Systems Engineering
Location	Boston
Host Group	RHEL OSP 6
Deploy on	Bare Metal

Table 6.1-1: RHEL OSP Installer Host Deployment Options

Upon selecting the **Host Group**, the previously defined host group values populate the various options.

9 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/index.html

Click the **Network** tab and provide the **MAC address** for the host machine under **Primary Interface**. This is the network interface used to PXE boot and deploy the operating system on the system.

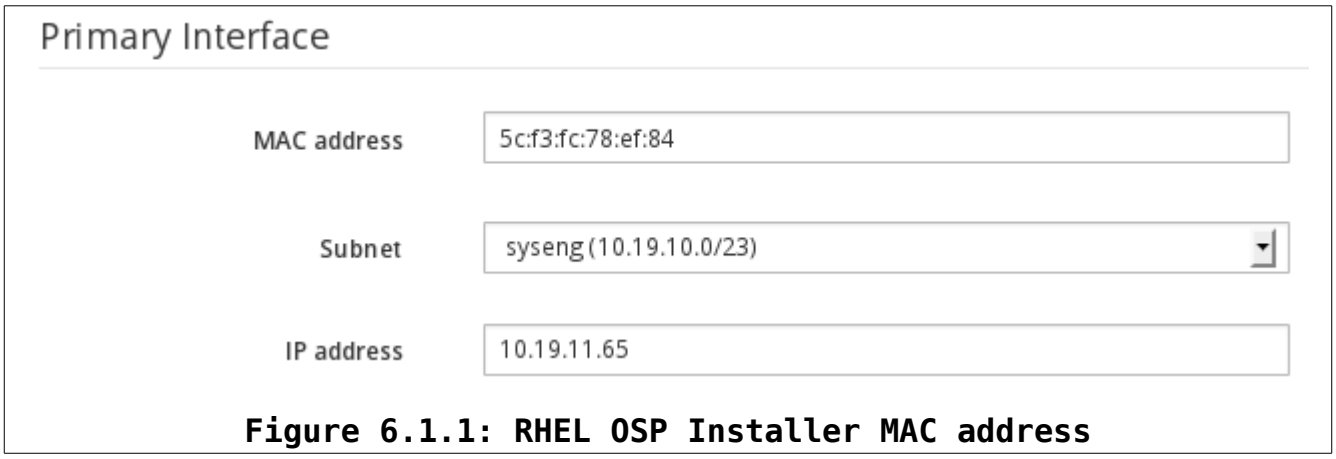


Figure 6.1.1: RHEL OSP Installer MAC address

On the **Operating System** tab verify the information listed and click **Resolve** to populate the **Provisioning templates**.

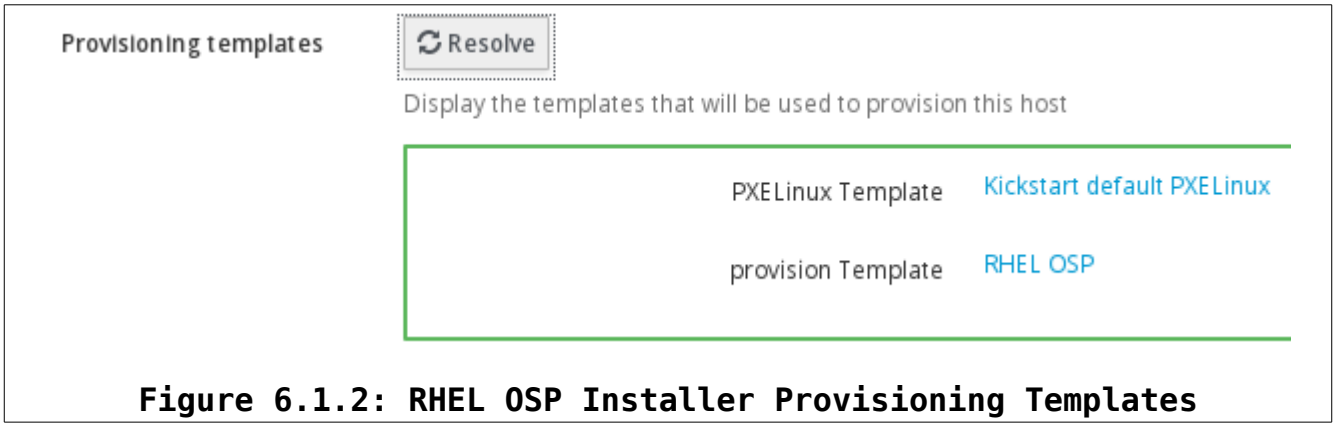


Figure 6.1.2: RHEL OSP Installer Provisioning Templates

With all remaining options verified, click **Submit** to begin the host deployment. Power on the bare metal machine to initiate the PXE boot process. Upon successful install, the host appears under **Hosts** > **All hosts**.

Figure 6.1.3: RHEL OSP Installer Deployment Status

Note: The host appears with a status of *No reports*. This is due to the `katello-agent` and `puppet` config management being disabled as mentioned previously in this section.

If issues arise during the deployment of a host, review the *install.post.log* file located under */root* on the host machine.

6.2 Pre-Deployment Preparation and `rhel-osp-installer` Execution

After deploying the RHEL OSP Installer host, several actions are required to prepare the host before running the *rhel-osp-installer* setup.

To prepare the RHEL OSP Installer host perform the following:

1. Set a static IP configuration for eno1
2. Configure eno2 for a static IP used for the provisioning network
3. Configure the RHEL OSP Installer host to perform as a gateway[10]
4. Configure iptables to open necessary ports
5. Modify */etc/resolv.conf* for necessary DNS servers

Refer to **Appendix D iptables and firewalld** and **Appendix E Scripts and Configuration Files** for settings used for the reference environment.

With the RHEL OSP Installer host pre-configured, execute the `rhel-ops-installer` command to complete the installation of the host prior to deploying the RHEL OSP environment. Refer to *Deploying OpenStack: Enterprise Environments (Red Hat Enterprise Linux OpenStack Platform)*[11] for the installation steps.

For the reference environment a basic deployment[12] is selected and the following settings are used during the `rhel-osp-installer` setup:

Network Configuration

```
Network interface: 'eno2'
```

10 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/Configuring_a_Gateway.html
11 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/index.html
12 https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/chap-Deployment_Scenario_1_Basic_Environment.html

```
IP address: '192.168.0.1'
Network mask: '255.255.255.0'
Network address: '192.168.0.0'
Host Gateway: '10.19.11.254'
DHCP range start: '192.168.0.2'
DHCP range end: '192.168.0.254'
DHCP Gateway: '192.168.0.1'
DNS forwarder: '10.19.11.248'
Domain: 'refarch.bos.redhat.com'
NTP sync host: '10.5.26.10'
Timezone: 'America/New_York'
```

Client Authentication Configuration

Set *root* password**.**

Installation Medium Configuration

```
Enter RHEL repo path:
1. Set RHEL repo path (http or https URL):
http://10.19.11.51/pulp/repos/RHSE/Library/content/dist/rhel/server/7/7.1/x8
6_64/kickstart/
```

Note: The URL path points to the Satellite 6 server in the reference environment.

Subscription Management Configuration

```
Enter your subscription manager credentials:
1. Subscription manager username: [REDACTED]
2. Subscription manager password: [REDACTED]
3. Comma separated repositories: rhel-7-server-openstack-6.0-rpms
rhel-7-server-openstack-6.0-installer-rpms rhel-7-server-rh-common-rpms
rhel-ha-for-rhel-7-server-rpms
4. Subscription manager pool (recommended): 8a85f9814bfa3a50014c234c0eca3812
5. Subscription manager proxy hostname:
6. Subscription manager proxy port:
7. Subscription manager proxy username:
8. Subscription manager proxy password:
9. Proceed with configuration
10. Skip this step (provisioning won't subscribe your machines)
```

Note: At the time of this publication the RHEL OSP Installer subscription manager options does not support activation keys which is required for Satellite 6. The following feature request has been filed: https://bugzilla.redhat.com/show_bug.cgi?id=1194839

Upon completion of the *rhel-osp-installer* execution, a URL and credentials are provided to access the RHEL OSP Installer user interface. It is recommended to change the admin password upon initial login.

```
Success!
  * Foreman is running at /
https://www.rhci-rhelosp-inst.refarch.bos.redhat.com
      Initial credentials are admin / Gzw2AiipJH72TwNW
```

```
  * Foreman Proxy is running at /
https://www.rhci-rhelosp-inst.refarch.bos.redhat.com:8443
  * Puppetmaster is running at port 8140
  The full log is at /var/log/rhel-osp-installer/rhel-osp-installer.log
```

Note: For high availability and more complex deployments using the RHEL OSP Installer, refer to the following reference architecture: *Deploying Highly Available Red Hat Enterprise Linux OpenStack Platform 6 with Ceph Storage - Using Red Hat Enterprise Linux OpenStack Platform Installer* [13]

6.3 Deploy RHEL OSP

Once the RHEL OSP Installer node setup is complete, login to the RHEL OSP Installer UI as admin and create a new subnet for external traffic.

Navigate to **Infrastructure** > **Subnets** and click **New Subnet**. For the reference environment the following values are used:

Option	Value
Name	External
Network address	10.19.10.0
Network mask	255.255.254.0
Gateway address	10.19.11.254
Primary DNS server	10.19.11.51 (address of the Satellite 6 server)
Secondary DNS server	10.19.11.248
IPAM	None
VLAN ID	<none>
Boot mode	DHCP

Table 6.3-1: RHEL OSP Installer External Subnet

Click the **Domains** tab and place a check mark next to the domain.

Figure 6.3.1: RHEL OSP Installer External Subnet Domains

Click **Submit** to complete.

Note: Verify the settings for the *default* subnet. Ensure the **Start of IP range** is set to a number greater than the RHEL OSP nodes deployed or plan to be deployed to avoid potential conflicts.

13 http://www.redhat.com/en/resources/deploy-rhel-openstack-platform-6-with-ceph-storage

Create a new RHEL OSP deployment by navigating to **OpenStack Installer** > **New deployment**. For the reference environment the following values are provided:

Tab	Option	Value
Deployment Settings	Name	rhci-rhelosp-inst
	Description	First RHEL OSP deployment
	Networking	Nova Network
	Message Provider	Qpid
	Platform	Red Hat Enterprise Linux OpenStack Platform 6 on RHEL 7
	Service Password	[REDACTED]
	Custom Repos	None
Network Configuration	External	Move to external subnet
	Tenant	Move to default subnet
Services Overview	N/A	N/A
Services Configuration (Nova)	Tenant Network Type	Flat with DHCP
	Floating IP range for external network	10.19.10.0/23
	Fixed IP range for tenant networks	192.168.0.0/24
	Tenant network device MTU	None
Services Configuration (Glance)	Choose Driver Backend	Local File
Services Configuration (Cinder)	Choose Driver Backend	LVM

Table 6.3-2: RHEL OSP Installer External Subnet

Click **Submit** to complete.

On the **Overview** tab under **Deployment Roles**, click the **+** next to **Controller**. Place a check mark next to the desired host and click **Assign Hosts**.

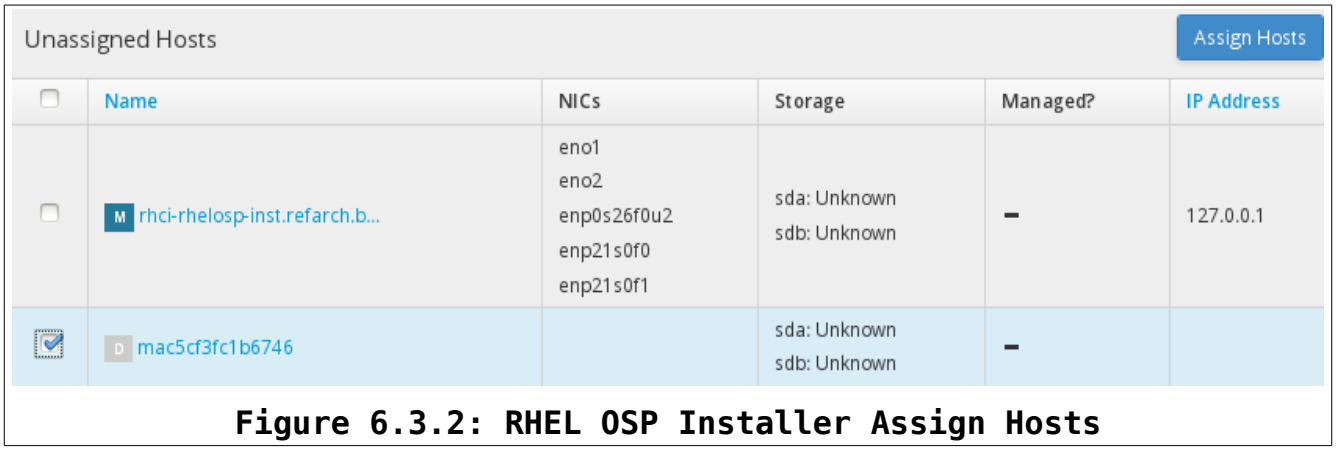


Figure 6.3.2: RHEL OSP Installer Assign Hosts

Perform the same for **Compute (Nova)**.

Click the **Hosts** tab and under **Assigned**, place a check mark next to one of the hosts and click **Configure Networks**. Move the *External* network to *eno1*. Click **Done** to complete. Perform the same for each remaining host.

Upon completion, click the **Deploy** button to begin deployment for the RHEL OSP environment.

A confirmation screen appears. Place a check mark next to **The networks have been configured for these hosts** and click **Deploy**.

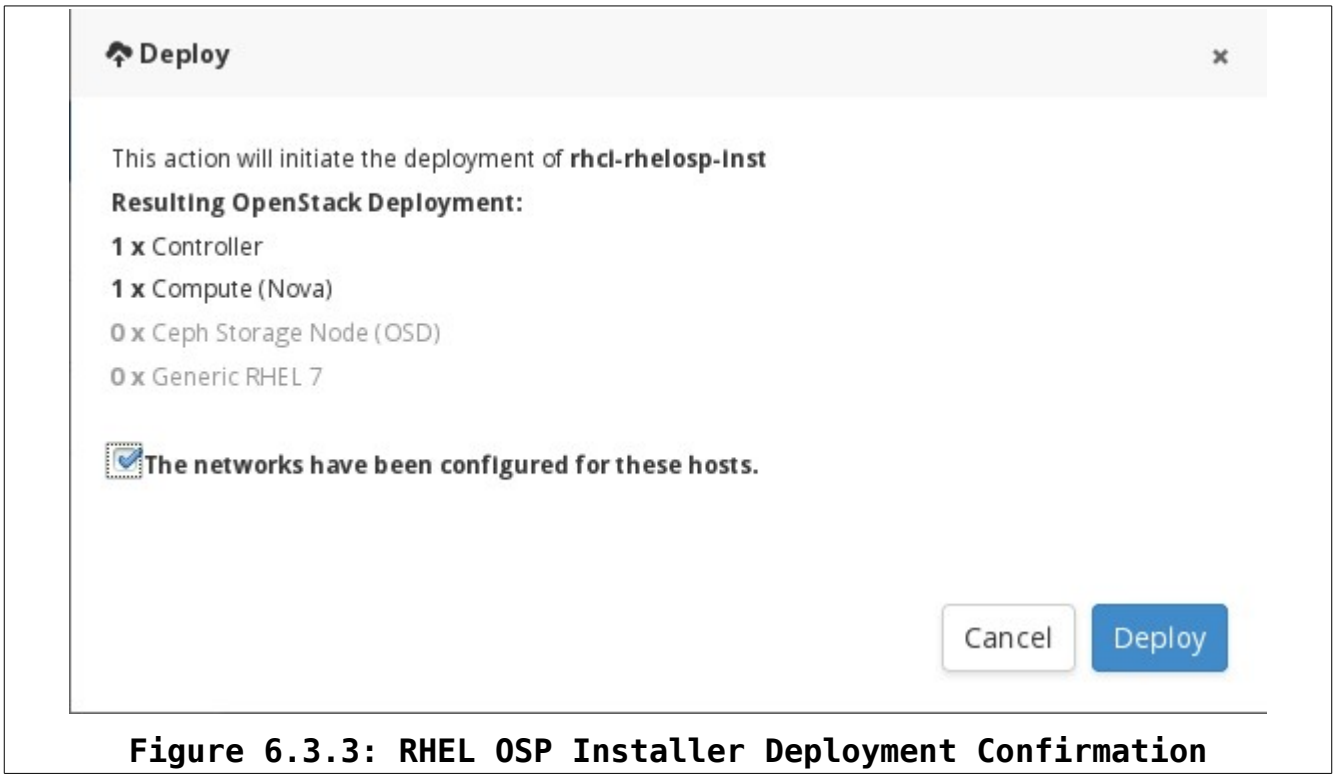


Figure 6.3.3: RHEL OSP Installer Deployment Confirmation

The RHEL OSP deployment begins and deployment progress is displayed.

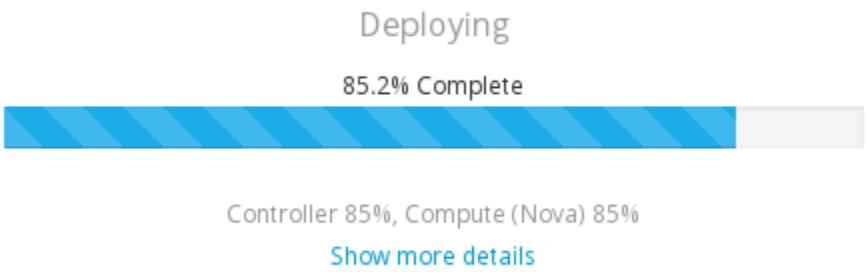


Figure 6.3.4: RHEL OSP Installer Deployment Progress

Note: Each RHEL OSP host must be PXE booted from the Foreman discovery image to begin the RHEL operating system install. For the reference environment, host deployment takes roughly two hours for completion.

Upon the RHEL OSP deployment completion, a success status is displayed along with the URL and credentials to the RHEL OSP dashboard. Additionally, Access all details button displays all deployed services and configured IP address.

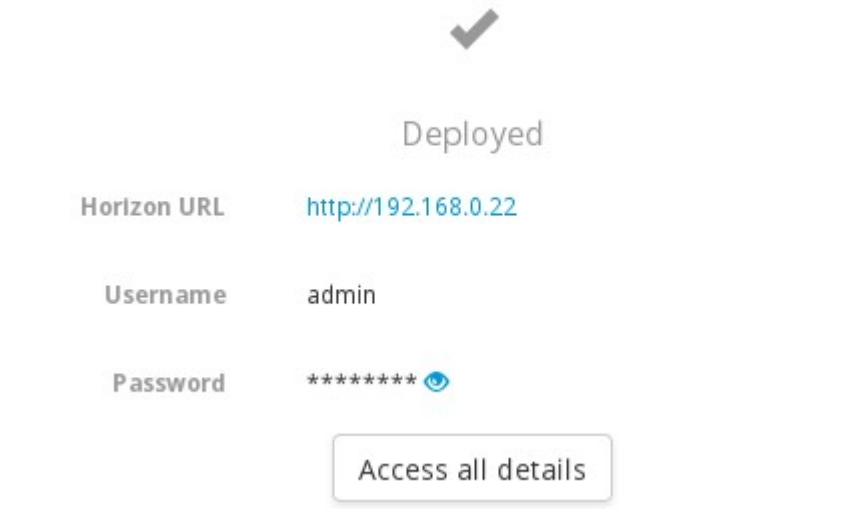


Figure 6.3.5: RHEL OSP Installer Deployment Completion

7 Deploying OpenShift Enterprise 2.2

With the underlying Red Hat Could Infrastructure deployed, the next step is deploying a cloud based Platform-as-a Service, such as Red Hat OpenShift Enterprise, to include development applications. For the reference environment OpenShift Enterprise is deployed onto the self-hosted RHEV compute resource.

7.1 Pre-deployment Preparation

Chapter 4 Red Hat Satellite 6 Configuration discusses preparing the Satellite environment to support the deployment of OpenShift Enterprise. In addition to configuring Satellite, the imported *openshift_origin* puppet module requires customization of the *Smart Class Parameters*[14].

To modify the openshift origin smart class parameter settings, navigate to **Configure > Puppet classes**, select the *openshift_origin* puppet class, click the **Smart Class Parameter tab**, and modify each parameter with the desired values.

The following parameters are modified for the reference environment:

Parameter	Override	Parameter type	Default Value
bind key	<checked>	string	4q9vJH2UPhtwHfQJzl8zu8XXtCYzExBMevU5SLG7VuGsLRi4hk8AuPVozb2fx3BJ4y1ejFZvpPJk1rtaOGEmqg==
broker hostname	<checked>	string	osebroker.refarch.bos.redhat.com
broker ip addr	<checked>	string	10.19.11.70
conf nameserver allow recursion	<checked>	boolean	true
Conf namerserver upstream dns	<checked>	array	["10.19.11.51"]
datastore_hostname	<checked>	string	osebroker.refarch.bos.redhat.com
dns infrastructure key	<checked>	string	4q9vJH2UPhtwHfQJzl8zu8XXtCYzExBMevU5SLG7VuGsLRi4hk8AuPVozb2fx3BJ4y1ejFZvpPJk1rtaOGEmqg==
dns infrastructure names	<checked>	array	[{"hostname":"osebroker.refarch.bos.redhat.com","ipaddr":"10.19.11.70"},{"hostname":"rhci-sat6.refarch.bos.redhat.com","ipaddr":"10.19.11.51"}]
dns infrastructure zone	<checked>	string	refarch.bos.redhat.com

14 http://www.theforeman.org/manuals/1.8/index.html#4.2.5ParameterizedClasses

Parameter	Override	Parameter type	Default Value
install cartridges	<checked>	array	["cron","diy","haproxy","mongodb","nodejs","perl","php","postgresql","python","ruby","jenkins","jenkins-client","mysql"]
install cartridges optional deps	<checked>	array	["php"]
install method	<checked>	string	none
msgserver hostname	<checked>	string	osebroker.refarch.bos.redhat.com
nameserver hostname	<checked>	string	osebroker.refarch.bos.redhat.com
nameserver ip addr	<checked>	string	10.19.11.70
node hostname	<checked>	string	<%= @host.name %>
ose version	<checked>	string	2.2
register host with nameserver	<checked>	boolean	true
*roles	<checked>	array	["broker","node","msgserver","datastore","nameserver"]

Table 7.1-1: OpenShift Origin Puppet Module Smart Class Parameters

*Additional settings for the *role* parameter:

Section	Option	Validator type
Optional Input Validator	Required <checked>	None
Override Value For Specific Hosts	Match	hostgroup=Broker
	Value	["broker","msgserver","datastore","nameserver"]
	Match	hostgroup=Nodes
	Value	["node"]

Table 7.1-2: OpenShift Origin roles Parameter Settings

Click **Submit** to complete.

7.2 Deployment

The first step to deploy the OpenShift Enterprise environment is deployment of the Broker host. To deploy onto the self-hosted RHEV environment, navigate to **Hosts** > **New host**.

For the reference environment the following values are used:

Option	Value
Name	osebroker
Organization	Systems Engineering
Location	Boston
Host Group	Broker
Deploy on	RHEV3.5 (RHEV)

Table 7.2-1: OpenShift Broker Host Deployment Options

On the **Network** tab, assign the configured IP address used for the *openshift_origin*, *broker ip addr* parameter.

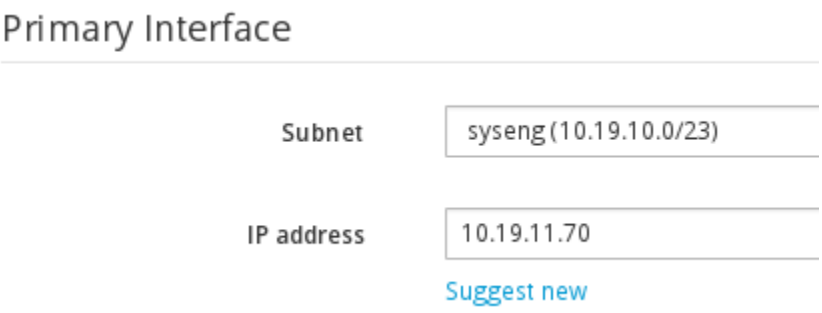


Figure 7.2.1: OpenShift Enterprise Broker – IP Address

Select the **Virtual Machine tab** and provide the desired values. For the reference environment the following values are used:

Option		Value
Cluster		Default
Template		Select template
Cores		1
Memory		4 GB
Network Interfaces	Name	nic1
	Network	rhevm
Volumes	Size (GB)	20
	Storage domain	Data

Option		Value
Volumes	Bootable	<checked>
Start	Power ON this machine	<checked>

Table 7.2-2: OpenShift Broker Virtual Machine Deployment Options

Click **Submit** to initiate the Broker host deployment.

Complete the OpenShift Enterprise deployment by deploying multiple OpenShift nodes by navigating to **Hosts** > **New host**.

For the reference environment the following values are used:

Option	Value
Name	osenode{1,2,3}
Organization	Systems Engineering
Location	Boston
Host Group	Nodes
Deploy on	RHEV3.5 (RHEV)

Table 7.2-3: OpenShift Nodes Host Deployment Options

On the **Network** tab, assign an IP address for each node. For the reference environment the following IP addresses are used for each node deployed:

Node	IP Address
osenode1	10.19.11.71
osenode2	10.19.11.72
osenode3	10.19.11.73

Table 7.2-4: OpenShift Nodes Host IP Address Assignment

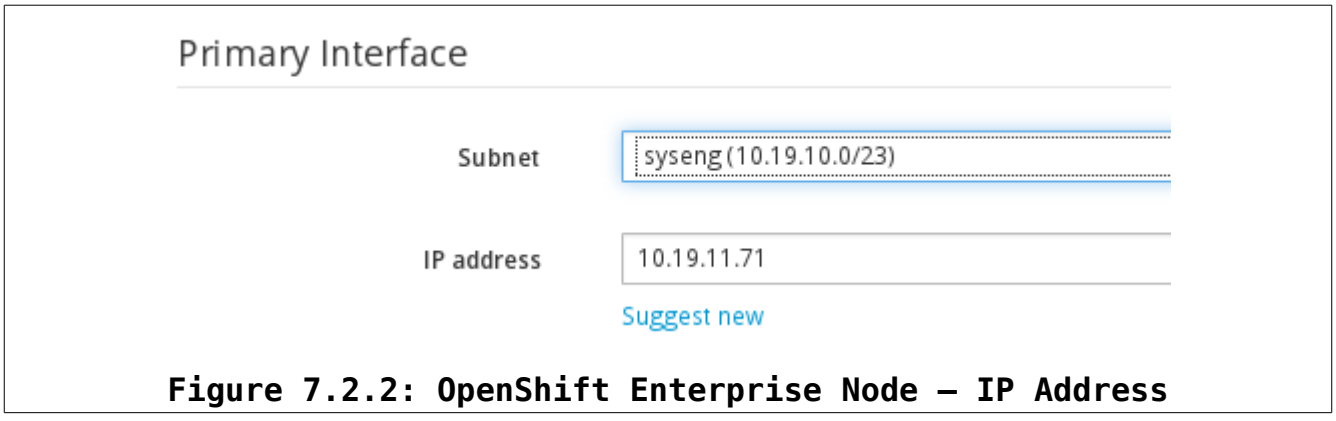


Figure 7.2.2: OpenShift Enterprise Node – IP Address

Select the **Virtual Machine tab** and provide the desired values. For the reference environment the following values are used:

Option		Value
Cluster		Default
Template		Select template
Cores		1
Memory		4 GB
Network Interfaces	Name	nic1
	Network	rhevm
Volumes	Size (GB)	20
	Storage domain	Data
	Bootable	<checked>
Start	Power ON this machine	<checked>

Table 7.2-5: OpenShift Node Virtual Machine Deployment Options

Click **Submit** to initiate the Node host deployment.

Navigate to Hosts > All hosts to review the deployment status.

Figure 7.2.3: OpenShift Enterprise Host Status

Note: "**A**" indicates the node(s) are actively running puppet configuration management. I.E. *openshift_origin* parameter customization for example.

7.3 Post Deployment Testing

Prior to deploying applications, cartridges, gears, etc. within OpenShift Enterprise, test the configuration by executing `oo-diagnostics` on the broker. Resolve any errors prior to proceeding.

Validate the OpenShift Enterprise deployment by performing the following actions:

1. On the broker create a district.

```
# oo-admin-ctl-district -c create -n php -p small
Successfully created district: 5536f1ab1bca37223b000001

{"_id"=>"5536f1ab1bca37223b000001",
```

```
 "uuid"=>"5536f1ab1bca37223b000001",
 "available_uids"=>"<6000 uids hidden>",
 "name"=>"php",
 "platform"=>"linux",
 "gear_size"=>"small",
 "available_capacity"=>6000,
 "max_uid"=>6999,
 "max_capacity"=>6000,
 "active_servers_size"=>0,
 "updated_at"=>2015-04-22 00:56:11 UTC,
 "created_at"=>2015-04-22 00:56:11 UTC}
```

2. Add a node to the district.

```
# oo-admin-ctl-district -c add-node -n jenkins -i /
osenode1.refarch.bos.redhat.com
Success for node 'osenode1.refarch.bos.redhat.com'!


{"_id"=>"5536f1ab1bca37223b000001",
 "active_servers_size"=>1,
 "available_capacity"=>6000,
 "available_uids"=>"<6000 uids hidden>",
 "created_at"=>2015-04-22 00:56:11 UTC,
 "gear_size"=>"small",
 "max_capacity"=>6000,
 "max_uid"=>6999,
 "name"=>"php",
 "platform"=>"linux",
 "servers"=>
  [{"_id"=>"5536f1d21bca37c02d000001",
    "active"=>true,
    "name"=>"osenode1.refarch.bos.redhat.com",
    "unresponsive"=>false}],
 "updated_at"=>2015-04-22 00:56:11 UTC,
 "uuid"=>"5536f1ab1bca37223b000001"}
```

Note: To add more nodes execute the same command for each node hostname.

3. Import cartridges from the node and view from the broker

```
# oo-admin-ctl-cartridge -c import-node --activate
Importing cartridges from node 'osenode2.refarch.bos.redhat.com'.
No change

# oo-admin-ctl-cartridge -c list
* cron-1.4         plugin    Cron 1.4           2015/04/22 00:51:28 UTC
* jenkins-client-1 plugin    Jenkins Client     2015/04/22 00:51:28 UTC
* mongodb-2.4      service   MongoDB 2.4        2015/04/22 00:51:28 UTC
* mysql-5.1        service   MySQL 5.1          2015/04/22 00:51:28 UTC
* mysql-5.5        service   MySQL 5.5          2015/04/22 00:51:28 UTC
* postgresql-8.4   service   PostgreSQL 8.4     2015/04/22 00:51:28 UTC
* postgresql-9.2   service   PostgreSQL 9.2     2015/04/22 00:51:28 UTC
* diy-0.1          web       Do-It-Yourself 0.1 2015/04/22 00:51:28 UTC
* jenkins-1        web       Jenkins Server     2015/04/22 00:51:28 UTC
* nodejs-0.10      web       Node.js 0.10       2015/04/22 00:51:28 UTC
```

```
* perl-5.10         web       Perl 5.10          2015/04/22 00:51:28 UTC
* php-5.3           web       PHP 5.3            2015/04/22 00:51:28 UTC
* php-5.4           web       PHP 5.4            2015/04/22 00:51:28 UTC
* python-2.6        web       Python 2.6         2015/04/22 00:51:28 UTC
* python-2.7        web       Python 2.7         2015/04/22 00:51:28 UTC
* python-3.3        web       Python 3.3         2015/04/22 00:51:28 UTC
* ruby-1.8          web       Ruby 1.8           2015/04/22 00:51:28 UTC
* ruby-1.9          web       Ruby 1.9           2015/04/22 00:51:28 UTC
* ruby-2.0          web       Ruby 2.0           2015/04/22 00:51:28 UTC
* haproxy-1.4       web_proxy Web Load Balancer  2015/04/22 00:51:28 UTC
```

4. Access the web console to complete application deployment

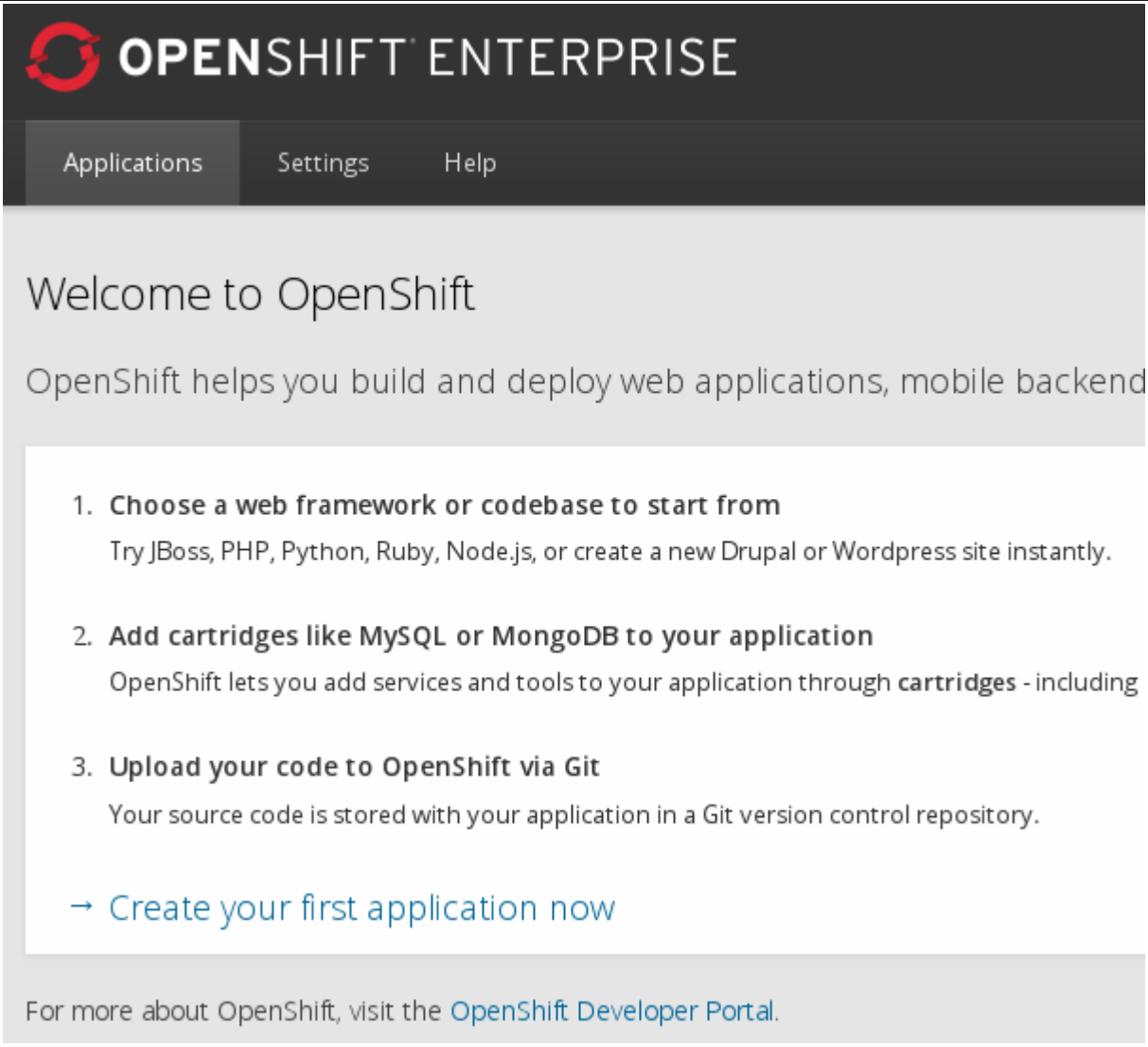


Figure 7.3.1: OpenShift Enterprise Web Console

Note: To access the OpenShift Enterprise web console, enter the hostname/IP address of the broker into a web browser and enter the default credentials: *demo/changeme*

5. Click **Create your first application now**. Choose the type of application. For the reference environment the *php* application is selected with the following values:

Option	Value
Public URL	http://php-newapp.example.com
Scaling	Scale with web traffic
Will you be changing the code of this application?	Not now, continue

Table 7.3-1: OpenShift Application Deployment

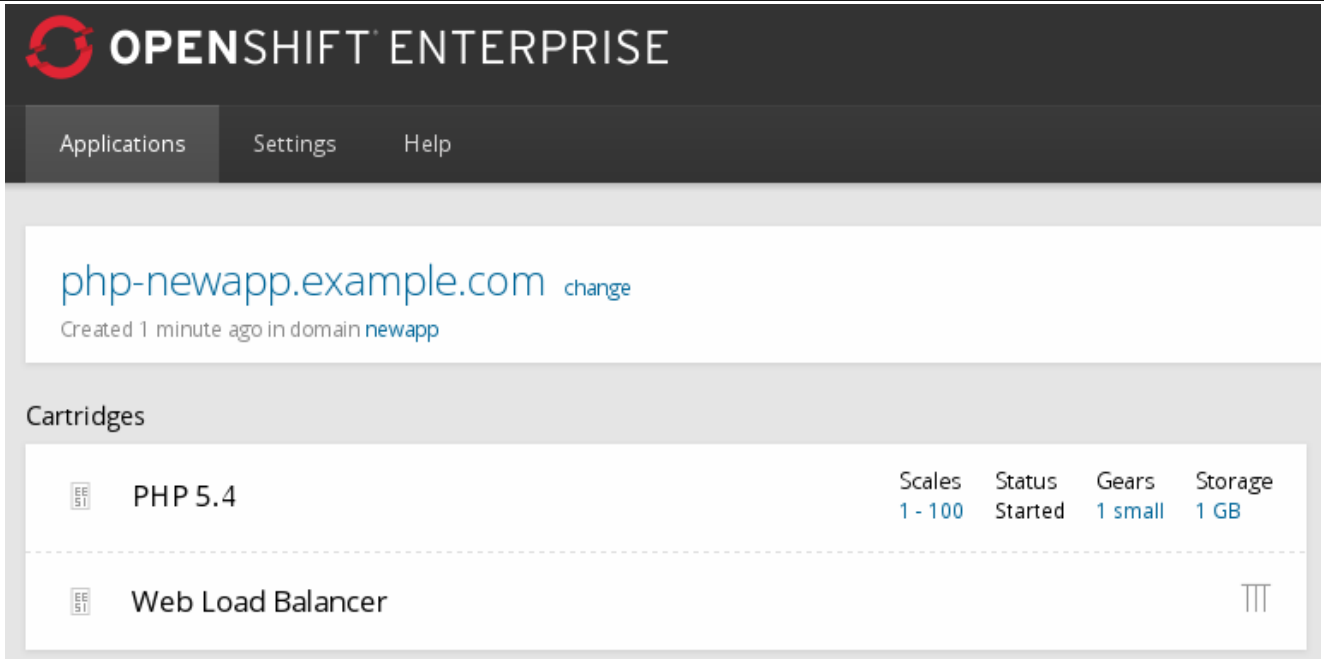


Figure 7.3.2: OpenShift *php* Application

Access the *php* application by navigating to the application web console.

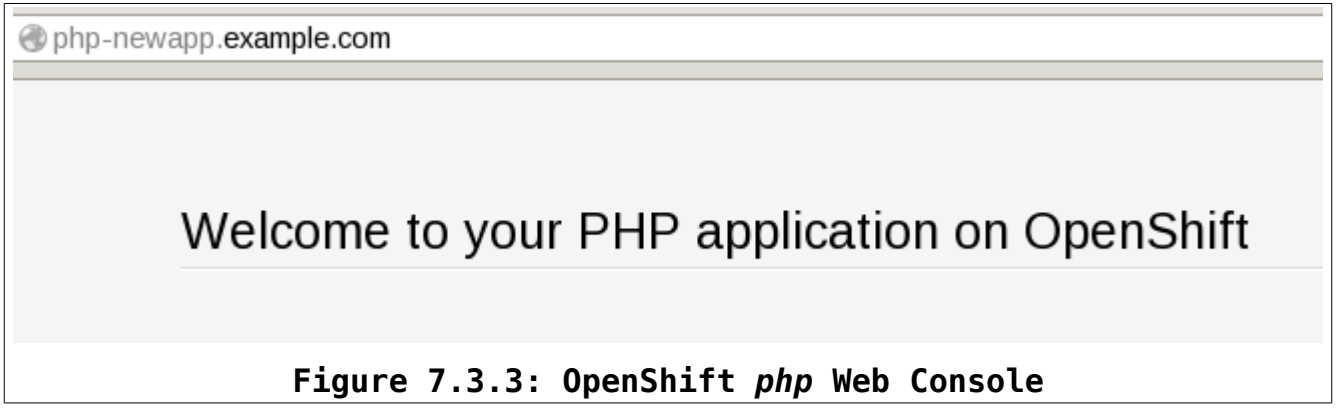


Figure 7.3.3: OpenShift *php* Web Console

7.4 Scaling an Application

In addition to deploying OpenShift Enterprise applications, the deployed applications can be scaled up by performing the following on the broker host:

```
# rhc app scale-up php -k
Login to osebroker.refarch.bos.redhat.com: demo
Password: ********

RESULT:
php scaled up
```

To verify scaling, perform the following:

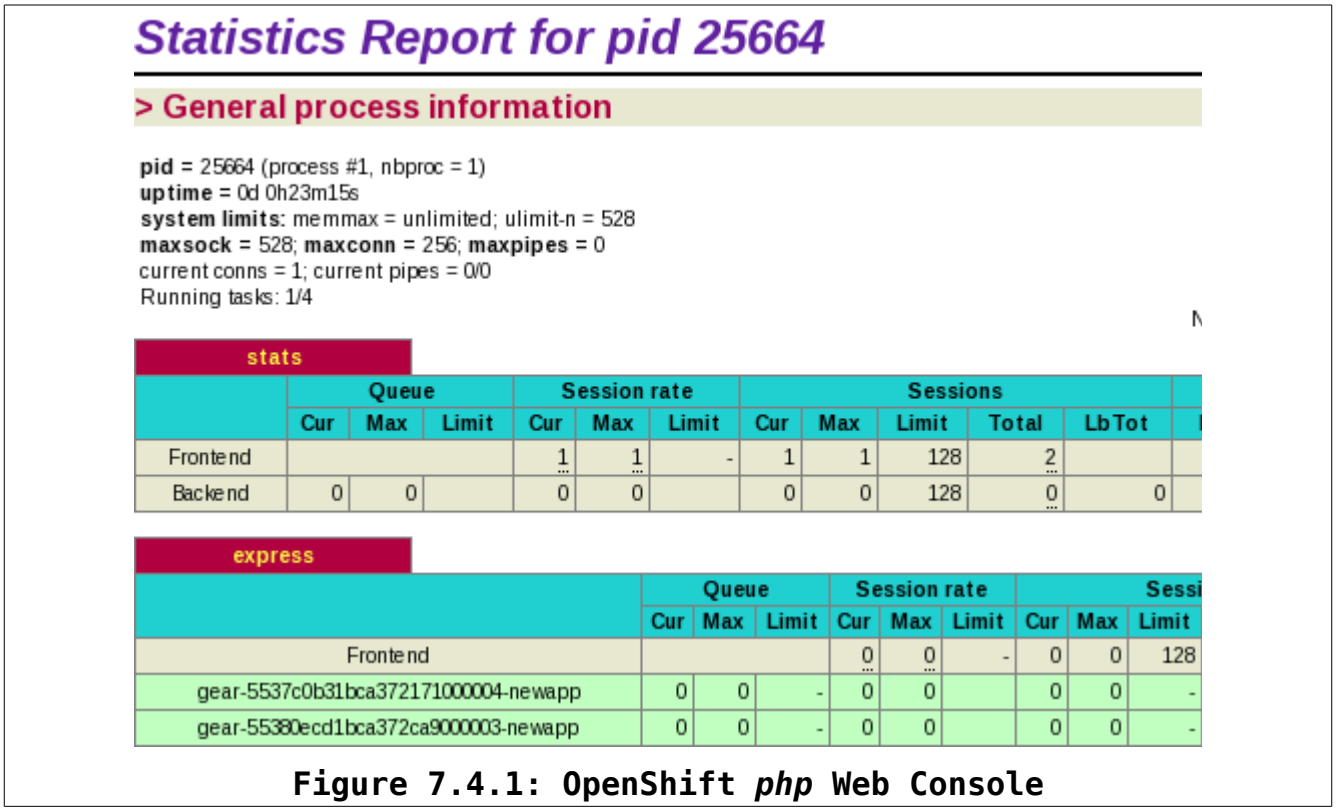
Before:

```
# rhc app show php --gears -v -k
Login to osebroker.refarch.bos.redhat.com: demo
Password: ********
ID                       State   Cartridges          Size  SSH URL
------------------------ ------- ------------------- -----
-------------------------------------------------------------------
5537bd511bca37af59000001 started php-5.4 haproxy-1.4 small \
5537bd511bca37af59000001@php-newapp.example.com
5537bf111bca37680b000013 started php-5.4             small \
5537bf111bca37680b000013@5537bf111bca37680b000013-newapp.example.com
```

After:

```
rhc app show php --gears -v -k
Login to osebroker.refarch.bos.redhat.com: demo
Password: ********
ID                       State   Cartridges          Size  SSH URL
------------------------ ------- ------------------- -----
-------------------------------------------------------------------
5537bd511bca37af59000001 started php-5.4 haproxy-1.4 small \
5537bd511bca37af59000001@php-newapp.example.com
5537c0b31bca372171000004 started php-5.4             small \
5537c0b31bca372171000004@5537c0b31bca372171000004-newapp.example.com
55380ecd1bca372ca9000003 started php-5.4             small \
55380ecd1bca372ca9000003@55380ecd1bca372ca9000003-newapp.example.com
```

For the *php* application, the following URL can be accessed to view status:
http://php-newapp.example.com/haproxy-status/



Statistics Report for pid 25664

> General process information

pid = 25664 (process #1, nbproc = 1)
uptime = 0d 0h23m15s
system limits: memmax = unlimited; ulimit-n = 528
maxsock = 528; **maxconn** = 256; **maxpipes** = 0
current conns = 1; current pipes = 0/0
Running tasks: 1/4

stats

	Queue			Session rate			Sessions					
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	
Frontend				1	1	-	1	1	128	2		
Backend	0	0		0	0		0	0	128	0	0	

express

	Queue			Session rate			Sessi		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit
Frontend				0	0	-	0	0	128
gear-5537c0b31bca372171000004-newapp	0	0	-	0	0		0	0	-
gear-55380ecd1bca372ca9000003-newapp	0	0	-	0	0		0	0	-

Figure 7.4.1: OpenShift *php* Web Console

7.5 Moving Gears Between Nodes

OpenShift applications[15] consist of cartridges and gears. Gears are the underlying component for which cartridges are composed of depending on the service for the application.

For scheduled maintenance it may be necessary to move gears between the OpenShift nodes. To move gears between nodes, perform the following:

1. Setup shared SSH keys between the broker and nodes:

Broker

```
# ssh-keygen -t rsa -b 2048 -f ~/.ssh/rsync_id_rsa
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/rsync_id_rsa.
Your public key has been saved in /root/.ssh/rsync_id_rsa.pub.
The key fingerprint is:
7d:12:50:8b:05:09:f5:ad:19:3c:15:59:fd:38:cc:da
root@osebroker.refarch.bos.redhat.com
The key's randomart image is:
+--[ RSA 2048]----+
|      .o++o o+.. |
```

15 https://www.openshift.com/products/architecture

```
|        .* +.   .|
|        . B .o ..|
|         . *  = .|
|        S = .o . |
|           o. E  |
|                 |
|                 |
|                 |
+-----------------+
```

2. Copy the public and private keys to */etc/openshift*

Broker

```
# cp /root/.ssh/rsync_id_rsa* /etc/openshift/
```

3. `scp` the shared ssh keys to each node:

Broker

```
# scp .ssh/rsync_id_rsa* \
root@osenode1.refarch.bos.redhat.com:/etc/openshift/
The authenticity of host 'osenode1.refarch.bos.redhat.com (10.19.11.71)'
can't be established.
RSA key fingerprint is 01:77:38:51:4d:74:e3:5a:78:ac:89:8f:6c:f8:6b:f6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'osenode1.refarch.bos.redhat.com,10.19.11.71'
(RSA) to the list of known hosts.
root@osenode1.refarch.bos.redhat.com's password:
rsync_id_rsa
100% 1675     1.6KB/s   00:00
rsync_id_rsa.pub
100%  419     0.4KB/s   00:00
```

4. On each node, follow the steps detailed in the *OpenShift Enterprise 2 Deployment Guide: Installing and Configuring OpenShift Enterprise*[16]*.*

5. Capture the UUID for the gears running under the php app:

Broker

```
# rhc app show php --gears -v -k
Login to osebroker.refarch.bos.redhat.com: demo
Password: ********
ID                       State   Cartridges          Size  SSH URL
------------------------ ------- ------------------ -----
-------------------------------------------------------------------
5537bd511bca37af59000001 started php-5.4 haproxy-1.4 small \
5537bd511bca37af59000001@php-newapp.example.com
5537c0b31bca372171000004 started php-5.4             small \
5537c0b31bca372171000004@5537c0b31bca372171000004-newapp.example.com
```

16 https://access.redhat.com/documentation/en-US/OpenShift_Enterprise/2/html/Deployment_Guide/Configuring_SSH_Keys_on_the_Node_Host.html

6. Move a gear from *osenode2* to *osenode1*:

Broker

```
# oo-admin-move --gear_uuid 5537bd511bca37af59000001 -i
osenode1.refarch.bos.redhat.com
URL: http://php-newapp.example.com
Login: demo
App UUID: 5537bd511bca37af59000001
Gear UUID: 5537bd511bca37af59000001
DEBUG: Source district uuid: 553721251bca3710ab000001
DEBUG: Destination district uuid: 553721251bca3710ab000001
DEBUG: Getting existing app 'php' status before moving
DEBUG: Gear component 'php-5.4' was running
DEBUG: Stopping existing app cartridge 'php-5.4' before moving
DEBUG: Stopping existing app cartridge 'haproxy-1.4' before moving
DEBUG: Force stopping existing app before moving
DEBUG: Gear platform is 'linux'
DEBUG: Creating new account for gear '5537bd511bca37af59000001' on
osenode1.refarch.bos.redhat.com
DEBUG: Moving content for app 'php', gear '5537bd511bca37af59000001' to
osenode1.refarch.bos.redhat.com
Identity added: /etc/openshift/rsync_id_rsa (/etc/openshift/rsync_id_rsa)
Agent pid 26935
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 26935 killed;
DEBUG: Moving system components for app 'php', gear
'5537bd511bca37af59000001' to osenode1.refarch.bos.redhat.com
Identity added: /etc/openshift/rsync_id_rsa (/etc/openshift/rsync_id_rsa)
Agent pid 26967
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 26967 killed;
DEBUG: Starting cartridge 'haproxy-1.4' in 'php' after move on
osenode1.refarch.bos.redhat.com
DEBUG: Starting cartridge 'php-5.4' in 'php' after move on
osenode1.refarch.bos.redhat.com
DEBUG: Fixing DNS and mongo for gear '5537bd511bca37af59000001' after move
DEBUG: Changing server identity of '5537bd511bca37af59000001' from
'osenode2.refarch.bos.redhat.com' to 'osenode1.refarch.bos.redhat.com'
DEBUG: Deconfiguring old app 'php' on osenode2.refarch.bos.redhat.com after
move
Successfully moved gear with uuid '5537bd511bca37af59000001' of app 'php'
from 'osenode2.refarch.bos.redhat.com' to 'osenode1.refarch.bos.redhat.com'
```

Additional information regarding deploying OpenShift Enterprise via puppet can be found in *OpenShift Enterprise 2 Puppet Deployment Guide: Installing and Configuring OpenShift Enterprise Using Puppet*[17].

17 https://access.redhat.com/documentation/en-US/OpenShift_Enterprise/2/html/Puppet_Deployment_Guide/index.html

8 Deploying Red Hat CloudForms 3.1

Deploying CloudForms is the final step in completing the Red Hat Cloud Infrastructure deployment and provides cloud management platform capabilities[18].

> **Note:** At the time of this writing, Satellite 6 does not support image based deployment. This feature i expected to be available in a future release of RHCI.

8.1 CloudForms Deployment

To deploy CloudForms, download the necessary image from the Red Hat Customer Portal. Navigate to **Downloads** and under **Cloud Products**, select **Red Hat CloudForms**. For the reference environment the *CloudForms OpenStack Virtual Appliance* is used.

Required steps and configuration to support running CloudForms within RHEL OSP are found in *Red Hat Cloud Infrastructure 5: Quick Start for Red Hat Enterprise Linux OpenStack Platform and Red Hat CloudForms with Smart Management*[19].

For the reference environment the following settings and configuration are used.

Option	Value
RHEL OSP Project	rhci
RHEL OSP User	rhci
Volume used for CFME database	20 GB

Table 8.1-1: RHEL OSP CloudForms Settings

Additionally the *default* security group is used with the following rules:

Direction	Protocol	Port
Ingress	TCP	22
Ingress	TCP	25
Ingress	UDP	53
Ingress	TCP	80
Ingress	TCP	389
Ingress	TCP	443
Ingress	TCP	636
Ingress	TCP	5000

18 http://www.redhat.com/en/technologies/cloud-computing/cloudforms

19 https://access.redhat.com/documentation/en-US/Red_Hat_Cloud_Infrastructure/5/html/Quick_Start_for_Red_Hat_Enterprise_Linux_OpenStack_Platform_and_Red_Hat_CloudForms_with_Smart_Management/index.html

Direction	Protocol	Port
Ingress	TCP	5432
Ingress	TCP	5672
Ingress	TCP	8773
Ingress	TCP	8774
Ingress	TCP	8777
Ingress	TCP	9292
Ingress	TCP	9696
Ingress	ICMP	N/A

Table 8.1-2: CloudForms RHEL OSP Security Group Rules

8.2 CloudForms Discovery

Discover the deployed self-hosted RHEV environment by logging into the CloudForms user interface with the *admin* account and navigate to **Infrastructure** > **Providers**. Click **Configuration** > **Add a New Infrastructure Provider** and enter the details for the RHEV provider. For the reference environment the following values are provided:

Section	Option	Value
Basic Information	Name	RHEV 3.5
	Type	Red Hat Enterprise Virtualization Manager
	Hostname	rhci-rhevm.refarch.bos.redhat.com
	IP Address	10.19.11.61
	API Port	443
	Zone	default
Credentials	User ID	admin@internal
	Password	[REDACTED]
	Verify Password	[REDACTED]

Table 8.2-1: RHEV Infrastructure Provider

After the RHEV infrastructure provider is added, place a check mark next to it, click on **Configuration**, and select **Refresh Relationships and Power States**. This immediately performs a discovery of the RHEV environment.

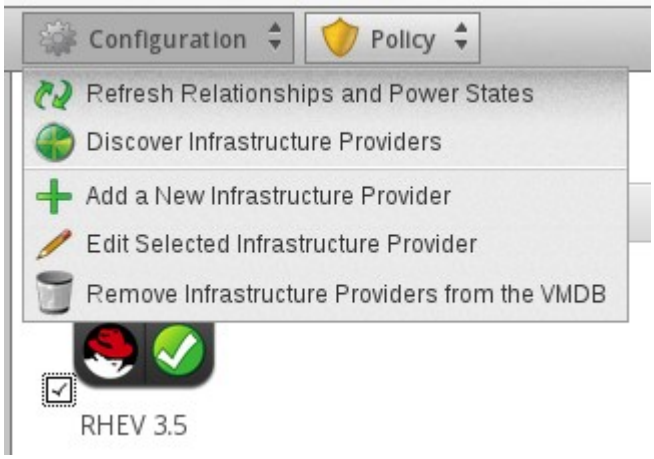


Figure 8.2.1: CloudForms RHEV Provider Relationship Refresh

Navigate to **Infrastructure** > **Virtual Machines** to begin managing machines in the self-deployed RHEV environment.

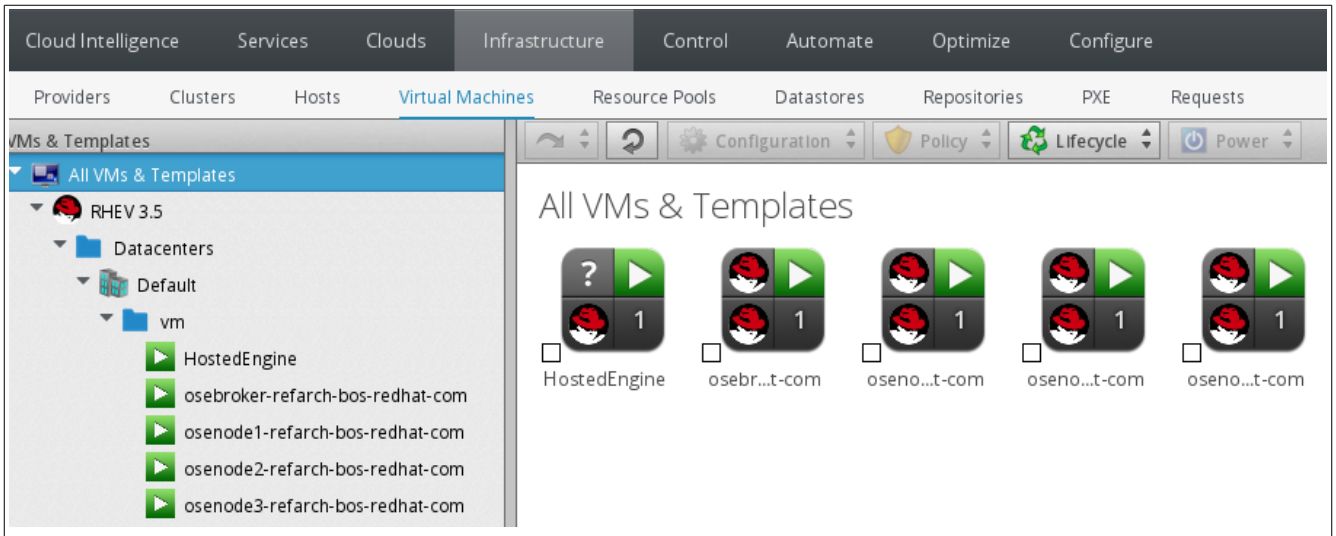


Figure 8.2.2: CloudForms RHEV Provider Virtual Machines

8.3 Satellite Registration

CloudForms 3.1 supports registering with on-premise Satellite 6 through the use of `subscription-manager`.

> **Note:** At the time of this writing registering the CloudForms appliance through the CloudForms portal is not supported. However this planned for a future release.

To register the CloudForms appliance with the on-premise Satellite 6 server perform the following actions by accessing the appliance via `ssh`:

1. SSH to the CloudForms appliance.

```
# ssh root@192.168.0.5
root@192.168.0.5's password:
[root@cloudforms ~]#
```

2. Use subscription-manager to register the CloudForms appliance to the on-premise Satellite 6 server.

```
# subscription-manager register --org RHSE --activationkey CloudForms
The system has been registered with ID: b68eb2cf-ad42-4faa-b1ec-b902edbf27be
Installed Product Current Status:
Product Name: Red Hat Software Collections (for RHEL Server)
Status:       Subscribed

Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed

Product Name: Red Hat CloudForms
Status:       Subscribed
```

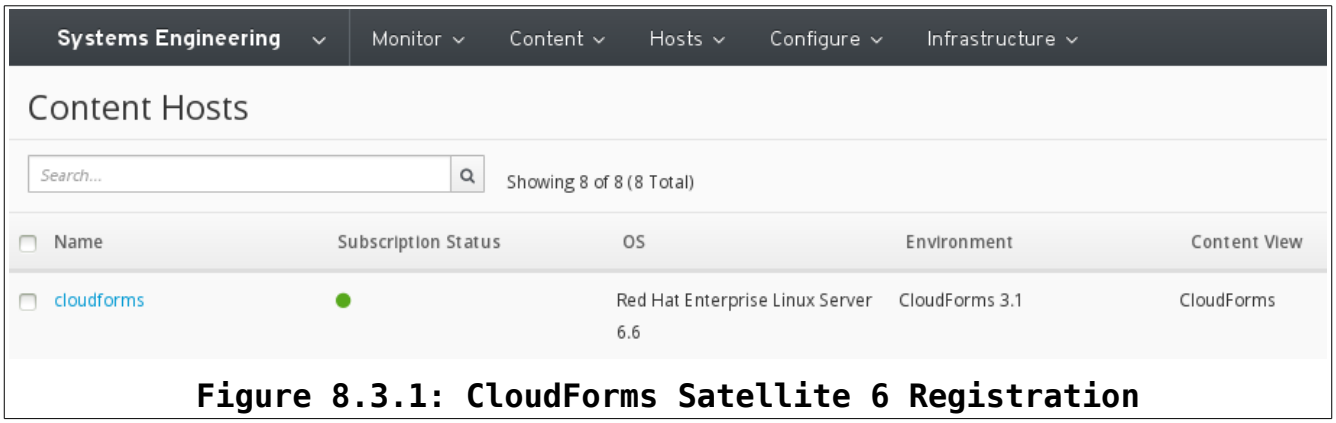


Figure 8.3.1: CloudForms Satellite 6 Registration

8.4 IdM Integration (optional)

Optionally CloudForms can be configured to support LDAP authentication for users and groups. For the reference environment LDAP authentication is provided Red Hat Identity Management.

To configure CloudForms for LDAP authentication, navigate to **Configure** > **Configuration** > A**uthentication**. Change **Mode** to *LDAP*. Fill in the appropriate details for the LDAP environment. For the reference environment the following values are used:

Section	Option	Value
LDAP Settings	LDAP Host Names	10.19.11.22
	LDAP Port	389
	User Type	Distinguished Name (CN=User)
	User Suffix: CN=<user>,	cn=users,cn=accounts,dc=refarch,dc=bos, dc=redhat,dc=com
Role Settings	Get User Groups from LDAP	<checked>
	Get Roles from Home Forest	<checked>
	Follow Referrals	<unchecked>
	Base DN	cn=users,cn=accounts,dc=refarch,dc=bos, dc=redhat,dc=com
	Bind DN	uid=admin,cn=users,cn=accounts,dc=refarch,dc=bos,dc=redhat,dc=com
	Bind Password	[REDACTED]

Table 8.4-1: CloudForms LDAP Settings

To add LDAP users and groups for authentication and role based access control (RBAC) refer to *Red Hat CloudForms 3.1: Management Engine 5.3 Settings and Operations Guide*[20].

20 https://access.redhat.com/documentation/en-US/Red_Hat_CloudForms/3.1/html/Management_Engine_5.3_Settings_and_Operations_Guide/sect-Access_Control.html#LDAP_Groups

For the reference environment, the IdM *development* group is added with the *devel* user (Clark Jones) assigned the *EVMRole – Administrator* with access only to the RHEV 3.5 infrastructure provider.

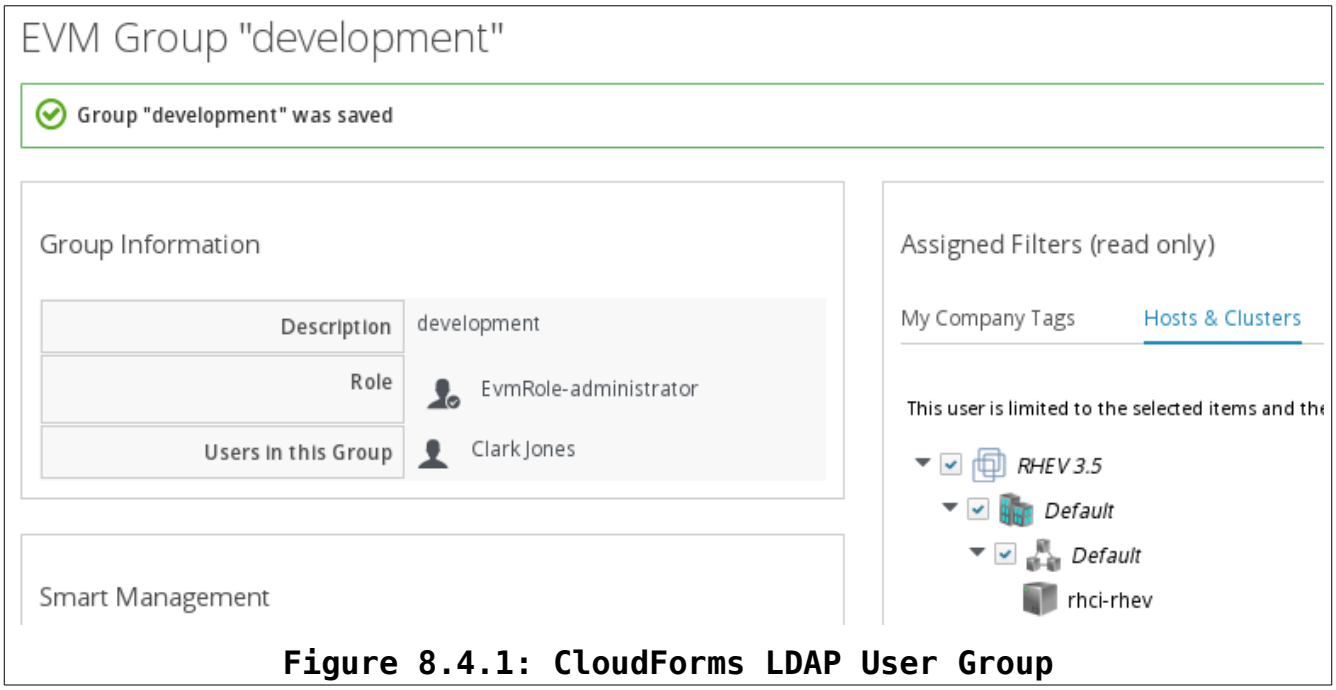


Figure 8.4.1: CloudForms LDAP User Group

9 Conclusion

Whether deploying highly customized virtual machines with complex lifecycle management or enabling agile development environments, Red Hat Cloud Infrastructure has it covered providing end-to-end, Infrastructure-as-a-Service (IaaS) on premise cloud capabilities.

The goal of this reference architecture is to demonstrate the the configuration of Red Hat Satellite 6 to successfully deploy and manage Red Hat Enterprise Virtualization 3.5, Red Hat Enterprise Linux OpenStack Platform 6, Red Hat OpenShift Entperprise 2.2, and Red Hat CloudForms 3.1 in a private cloud setting. The following use cases are successfully demonstrated:

- Configuration of Satellite 6 to support deploying RHCI components
- Deploying a self-hosted RHEV environment from Satellite 6
- Deploying a RHEL OSP environment from Satellite 6
- Deploying OpenShift Enterprise using OpenShift Origin puppet modules onto RHEV
- Deploying CloudForms and demonstrating discovery and management of RHCI on-premise cloud

In addition to, optional integration with Red Hat Identity Management for Red Hat Cloud Infrastructure components is demonstrated where applicable.

Appendix A: Revision History

Revision 1.0	Thursday, April 30, 2015	Brett Thurber
Initial Release		
Revision 1.1	Monday, September 21, 2015	Brett Thurber
Section 7 updates		

Appendix B: Contributors

Contributor	Title	Contribution
Scott Dodson	Sr. Software Engineer	Content, Review
Balaji Jayavelu	Principal Software Engineer	Content, Review

Appendix C: Satellite 6 Install Settings

```
#katello-installer --foreman-admin-username admin –foreman-admin-password \
[REDACTED] --capsule-dhcp true --capsule-dhcp-gateway 10.19.11.254 \
--capsule-dhcp-nameservers 10.19.11.51 --capsule-dhcp-range "10.19.11.150 \
10.19.11.200" --capsule-dns true --capsule-dns-forwarders 10.19.143.248 \
--capsule-dns-reverse 11.19.10.in-addr.arpa –capsule-dns-zone \
refarch.bos.redhat.com --capsule-tftp true --capsule-tftp-servername \
$(hostname) --capsule-puppet true --capsule-puppetca true \
--capsule-dhcp-interface eth0 --capsule-dns-interface eth0
```

Appendix D: iptables and firewalld

cf-vms

```
# Completed on Thu May 26 05:59:18 2011
# Generated by iptables-save v1.4.7 on Thu May 26 05:59:18 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [43:20252]
:CF - [0:0]
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -j CF
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 111 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 111 -j ACCEPT
```

```
-A INPUT -p udp -m state --state NEW -m udp --dport 662 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 662 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 875 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 875 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 892 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 892 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 2049 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 32769 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 32803 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.122.0/24 -o virbr0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.122.0/24 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
-A CF -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A CF -p udp -m state --state NEW -m udp --dport 2049 -j ACCEPT
COMMIT
# Completed on Thu May 26 05:59:18 2011
```

rhci-rhelosp-inst

```
# Generated by iptables-save v1.4.21 on Sun Apr 19 03:54:47 2015
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [1:142]
:POSTROUTING ACCEPT [1:142]
-A POSTROUTING -s 192.168.0.0/24 -j MASQUERADE
COMMIT
# Completed on Sun Apr 19 03:54:47 2015
# Generated by iptables-save v1.4.21 on Sun Apr 19 03:54:47 2015
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m multiport --ports 22 -m comment --comment "22 accept -
ssh" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 443 -m comment --comment "443 accept -
apache" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 53 -m comment --comment "53 accept -
dns tcp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 53 -m comment --comment "53 accept -
dns udp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 67 -m comment --comment "67 accept -
dhcp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 68 -m comment --comment "68 accept -
bootp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 69 -m comment --comment "69 accept -
tftp" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 80 -m comment --comment "80 accept -
```

```
apache" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 8140 -m comment --comment "8140 accept
- puppetmaster" -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 111 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 111 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 32803 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 32769 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2020 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 2020 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 662 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 662 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 892 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 892 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 875 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 875 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 69 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 8140 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8140 -j ACCEPT
-A FORWARD -o eno+ -j ACCEPT
-A FORWARD -s 192.168.0.0/24 -j ACCEPT
-A FORWARD -d 192.168.0.0/24 -j ACCEPT
-A FORWARD -i eno+ -j ACCEPT
COMMIT
# Completed on Sun Apr 19 03:54:47 2015
```

rhci-sat6

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers
on networks to not harm your computer. Only selected incoming connections
are accepted.</description>
  <service name="dhcpv6-client"/>
  <service name="ssh"/>
  <port protocol="tcp" port="443"/>
  <port protocol="tcp" port="80"/>
  <port protocol="tcp" port="8140"/>
  <port protocol="tcp" port="9090"/>
  <port protocol="tcp" port="8080"/>
  <port protocol="udp" port="69"/>
  <port protocol="udp" port="636"/>
  <port protocol="tcp" port="69"/>
  <port protocol="tcp" port="53"/>
```

```
  <port protocol="udp" port="389"/>
  <port protocol="tcp" port="389"/>
  <port protocol="udp" port="53"/>
  <port protocol="tcp" port="5671"/>
  <port protocol="tcp" port="636"/>
</zone>
```

rhci-rhev

```
# oVirt default firewall configuration. Automatically generated by vdsm
bootstrap script.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
# vdsm
-A INPUT -p tcp --dport 54321 -j ACCEPT
# SSH
-A INPUT -p tcp --dport 22 -j ACCEPT
# snmp
-A INPUT -p udp --dport 161 -j ACCEPT


# libvirt tls
-A INPUT -p tcp --dport 16514 -j ACCEPT

# guest consoles
-A INPUT -p tcp -m multiport --dports 5900:6923 -j ACCEPT

# migration
-A INPUT -p tcp -m multiport --dports 49152:49216 -j ACCEPT


# Reject any other input traffic
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev ! --physdev-is-bridged -j REJECT --reject-with
icmp-host-prohibited
COMMIT
```

rhci-rhevm

```
# Generated by ovirt-engine installer
#filtering rules
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 7410 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 6100 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

#drop all rule
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

rhci-idm

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers
on networks to not harm your computer. Only selected incoming connections
are accepted.</description>
  <service name="dhcpv6-client"/>
  <service name="ssh"/>
  <port protocol="tcp" port="443"/>
  <port protocol="tcp" port="80"/>
  <port protocol="tcp" port="464"/>
  <port protocol="tcp" port="138"/>
  <port protocol="udp" port="88"/>
  <port protocol="udp" port="464"/>
  <port protocol="tcp" port="445"/>
  <port protocol="tcp" port="88"/>
  <port protocol="tcp" port="7389"/>
  <port protocol="udp" port="139"/>
  <port protocol="udp" port="123"/>
  <port protocol="tcp" port="139"/>
  <port protocol="tcp" port="389"/>
  <port protocol="tcp" port="22"/>
  <port protocol="tcp" port="53"/>
  <port protocol="udp" port="389"/>
  <port protocol="udp" port="138"/>
  <port protocol="udp" port="445"/>
  <port protocol="udp" port="53"/>
  <port protocol="tcp" port="636"/>
</zone>
```

mac001a6476000b – RHEL OSP Controller

```
# Generated by iptables-save v1.4.21 on Mon Apr 20 23:26:42 2015
*filter
:INPUT ACCEPT [1:52]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [3:206]
:nova-api-FORWARD - [0:0]
:nova-api-INPUT - [0:0]
:nova-api-OUTPUT - [0:0]
:nova-api-local - [0:0]
:nova-filter-top - [0:0]
```

```
-A INPUT -j nova-api-INPUT
-A INPUT -p tcp -m multiport --dports 15672,5672 -m comment --comment "001
amqp incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 80,443 -m comment --comment "001
apache incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 8776 -m comment --comment "001 cinder
incoming" -j ACCEPT
-A INPUT -p udp -m multiport --dports 5404,5405 -m comment --comment "001
corosync mcast" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 3306,9200,4567,4568,4444 -m comment
--comment "001 galera incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 9191,9292 -m comment --comment "001
glance incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 8004 -m comment --comment "001 heat
incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 5000,35357 -m comment --comment "001
keystone incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 81 -m comment --comment "001 load
balancer incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 27017 -m comment --comment "001 nosql
incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 8774,8775,6080,6081 -m comment
--comment "001 nova incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 2224 -m comment --comment "001 pcsd"
-j ACCEPT
-A INPUT -p tcp -m multiport --dports 3260 -m comment --comment "010 iscsi
incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 11211 -m comment --comment "010
memcached incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 873 -m comment --comment "010 rsync
incoming" -j ACCEPT
-A FORWARD -j nova-filter-top
-A FORWARD -j nova-api-FORWARD
-A OUTPUT -j nova-filter-top
-A OUTPUT -j nova-api-OUTPUT
-A nova-api-INPUT -d 192.168.0.2/32 -p tcp -m tcp --dport 8775 -j ACCEPT
-A nova-filter-top -j nova-api-local
COMMIT
# Completed on Mon Apr 20 23:26:42 2015
# Generated by iptables-save v1.4.21 on Mon Apr 20 23:26:42 2015
*nat
:PREROUTING ACCEPT [241:16844]
:INPUT ACCEPT [137:9916]
:OUTPUT ACCEPT [31832:1915535]
:POSTROUTING ACCEPT [31832:1915535]
:nova-api-OUTPUT - [0:0]
:nova-api-POSTROUTING - [0:0]
:nova-api-PREROUTING - [0:0]
:nova-api-float-snat - [0:0]
:nova-api-snat - [0:0]
:nova-postrouting-bottom - [0:0]
-A PREROUTING -j nova-api-PREROUTING
-A OUTPUT -j nova-api-OUTPUT
-A POSTROUTING -j nova-api-POSTROUTING
-A POSTROUTING -j nova-postrouting-bottom
```

```
-A nova-api-snat -j nova-api-float-snat
-A nova-postrouting-bottom -j nova-api-snat
COMMIT
# Completed on Mon Apr 20 23:26:42 2015
# Generated by iptables-save v1.4.21 on Mon Apr 20 23:26:42 2015
*mangle
:PREROUTING ACCEPT [649919:202124561]
:INPUT ACCEPT [649803:202116817]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [650820:205097465]
:POSTROUTING ACCEPT [650820:205097465]
:nova-api-POSTROUTING - [0:0]
-A POSTROUTING -j nova-api-POSTROUTING
COMMIT
# Completed on Mon Apr 20 23:26:42 2015
```

mac5cf3fc1b6746 – RHEL OSP Compute

```
# Generated by iptables-save v1.4.21 on Sun Apr 19 01:28:06 2015
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m multiport --dports 5900:5999 -m comment --comment "001
nova compute incoming" -j ACCEPT
COMMIT
# Completed on Sun Apr 19 01:28:06 2015
```

osebroker

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 27017 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 61613 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

osenodes{1,2,3}

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
```

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8000 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8443 -j ACCEPT
:rhc-app-comm - [0:0]
-A INPUT -m tcp -p tcp --dport 35531:65535 -m state --state NEW -j ACCEPT
-A INPUT -j rhc-app-comm
-A OUTPUT -j rhc-app-comm
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Appendix E: Scripts and Configuration Files

big_hammer.sh

```
#!/bin/bash

#This is the "big hammer" approach to installing the Openshift Origin puppet
module for Red Hat Satellite 6.
#Courtesy of Scott Dodson; sdodson@redhat.com

puppet module install openshift-openshift_origin
cd /etc/puppet/modules
for i in `ls`; do cd /etc/puppet/modules/$i && puppet module build; done
find /etc/puppet/modules/ -wholename '*pkg/*.tar.gz' -exec hammer -v
repository upload-content --product=openshift --name=openshift --path={}
--organization="Systems Engineering" \;
```

cf-vms – /etc/sysconfig/nfs

```
# Define which protocol versions mountd
# will advertise. The values are "no" or "yes"
# with yes being the default
#MOUNTD_NFS_V2="no"
#MOUNTD_NFS_V3="no"
#
#
# Path to remote quota server. See rquotad(8)
#RQUOTAD="/usr/sbin/rpc.rquotad"
# Port rquotad should listen on.
#RQUOTAD_PORT=875
# Optional options passed to rquotad
#RPCRQUOTADOPTS=""
```

```
#
#
# Optional arguments passed to in-kernel lockd
#LOCKDARG=
# TCP port rpc.lockd should listen on.
LOCKD_TCPPORT=32803
# UDP port rpc.lockd should listen on.
LOCKD_UDPPORT=32769
#
#
# Optional arguments passed to rpc.nfsd. See rpc.nfsd(8)
# Turn off v2 and v3 protocol support
#RPCNFSDARGS="-N 2 -N 3"
# Turn off v4 protocol support
#RPCNFSDARGS="-N 4"
# Number of nfs server processes to be started.
# The default is 8.
#RPCNFSDCOUNT=8
# Stop the nfsd module from being pre-loaded
#NFSD_MODULE="noload"
# Set V4 grace period in seconds
#NFSD_V4_GRACE=90
#
#
#
# Optional arguments passed to rpc.mountd. See rpc.mountd(8)
#RPCMOUNTDOPTS=""
# Port rpc.mountd should listen on.
MOUNTD_PORT=892
#
#
# Optional arguments passed to rpc.statd. See rpc.statd(8)
#STATDARG=""
# Port rpc.statd should listen on.
STATD_PORT=662
# Outgoing port statd should used. The default is port
# is random
#STATD_OUTGOING_PORT=2020
# Specify callout program
#STATD_HA_CALLOUT="/usr/local/bin/foo"
#
#
# Optional arguments passed to rpc.idmapd. See rpc.idmapd(8)
#RPCIDMAPDARGS=""
#
# Set to turn on Secure NFS mounts.
#SECURE_NFS="yes"
# Optional arguments passed to rpc.gssd. See rpc.gssd(8)
#RPCGSSDARGS=""
# Optional arguments passed to rpc.svcgssd. See rpc.svcgssd(8)
#RPCSVCGSSDARGS=""
#
# To enable RDMA support on the server by setting this to
# the port the server should listen on
#RDMA_PORT=20049
```

cf-vms – /etc/exports

```
/nfsshare  10.19.11.*(rw,sync,no_root_squash,fsid=0)
```

rhci-rhelosp-inst - /etc/sysconfig/network-scripts/ifcfg-eno1

```
NAME="eno1"
DEVICE="eno1"
ONBOOT=yes
NETBOOT=yes
PEERDNS=no
UUID="a84075d1-d1b1-4d4d-b08a-88b7ecb8c160"
BOOTPROTO=static
IPADDR=10.19.11.65
NETMASK=255.255.254.0
GATEWAY=10.19.11.254
TYPE=Ethernet
```

rhci-rhelosp-inst - /etc/sysconfig/network-scripts/ifcfg-eno2

```
DEVICE=eno2
BOOTPROTO=none
HWADDR=5c:f3:fc:78:ef:86
ONBOOT=yes
HOTPLUG=yes
TYPE=Ethernet
IPADDR=192.168.0.1
NETMASK=255.255.255.0
PEERDNS=yes
DNS1=192.168.0.1
DNS2=10.19.11.248
NM_CONTROLLED=no
```

rhci-rhelosp-inst - /usr/lib/sysctl.d/00-system.conf

```
# Kernel sysctl configuration file
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Disable netfilter on bridges.
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 4294967295

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 268435456

#Gateway
net.ipv4.ip_forward = 1
```

rhci-rhelosp-inst - /etc/resolv.conf

```
# Generated by NetworkManager
search refarch.bos.redhat.com


# No nameservers found; try putting DNS servers into your
# ifcfg files in /etc/sysconfig/network-scripts like so:
#
# DNS1=xxx.xxx.xxx.xxx
# DNS2=xxx.xxx.xxx.xxx
# DOMAIN=lab.foo.com bar.foo.com
nameserver 192.168.0.1
nameserver 10.19.11.248
```

rhci-rhevm – profile1.properties

```
#
# Select one
#
#include = <openldap.properties>
#include = <389ds.properties>
#include = <rhds.properties>
include = <ipa.properties>
#include = <iplanet.properties>
#include = <rfc2307.properties>
#include = <rfc2307-openldap.properties>

#
# Server
#
vars.server = rhci-idm.refarch.bos.redhat.com

#
# Search user and its password.
#
vars.user =
uid=admin,cn=users,cn=accounts,dc=refarch,dc=bos,dc=redhat,dc=com
vars.password = [REDACTED]

pool.default.serverset.single.server = ${global:vars.server}
pool.default.auth.simple.bindDN = ${global:vars.user}
pool.default.auth.simple.password = ${global:vars.password}

# Create keystore, import certificate chain and uncomment
# if using ssl/tls.
#pool.default.ssl.startTLS = true
#pool.default.ssl.truststore.file = ${local:_basedir}/$
{global:vars.server}.jks
#pool.default.ssl.truststore.password = changeit
```

Appendix F: kickstarts and snippets

RHEVM Hosted

```
<%#
kind: provision
name: Satellite Kickstart Default
oses:
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
%>
<%
  rhel_compatible = @host.operatingsystem.family == 'Redhat' &&
@host.operatingsystem.name != 'Fedora'
  os_major = @host.operatingsystem.major.to_i
  # safemode renderer does not support unary negation
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = pm_set || @host.params['force-puppet']
  section_end = (rhel_compatible && os_major <= 5) ? '' : '%end'
%>
install
<%= @mediapath %>
lang en_US.UTF-8
selinux --enforcing
keyboard us
skipx
network --bootproto <%= @static ? "static --ip=#{@host.ip}
--netmask=#{@host.subnet.mask} --gateway=#{@host.subnet.gateway}
--nameserver=#{[@host.subnet.dns_primary,@host.subnet.dns_secondary].reject{
|n| n.blank?}.join(',')}" : 'dhcp' %> --hostname <%= @host %>
rootpw --iscrypted <%= root_pass %>
firewall --<%= os_major >= 6 ? 'service=' : '' %>ssh
authconfig --useshadow --passalgo=sha256 --kickstart
timezone --utc <%= @host.params['time-zone'] || 'UTC' %>

bootloader --location=mbr --append="nofb quiet splash=quiet" <%= grub_pass
%>
<% if os_major == 5 -%>
key --skip
<% end -%>


<% if @dynamic -%>
%include /tmp/diskpart.cfg
<% else -%>
<%= @host.diskLayout %>
<% end -%>

text
reboot

%packages --ignoremissing
yum
```

```
dhclient
ntp
wget
@Core
<%= section_end -%>

<% if @dynamic -%>
%pre
<%= @host.diskLayout %>
<%= section_end -%>
<% end -%>

%post --nochroot
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
cp -va /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
/usr/bin/chvt 1
) 2>&1 | tee /mnt/sysimage/root/install.postnochroot.log
<%= section_end -%>

%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '10.16.255.2' %>
/usr/sbin/hwclock --systohc
chkconfig ntpd on

<%= snippet "subscription_manager_registration" %>

<% if @host.respond_to?(:realm) && @host.respond_to?(:otp) && @host.realm &&
@host.otp && @host.realm.realm_type == "Red Hat Identity Management" -%>
<%= snippet "idm_register" %>
<% end -%>

# update all the base packages from the updates repository
yum -t -y -e 0 update
yum install -y rhevm

<% if puppet_enabled %>
# and add the puppet package
yum -t -y -e 0 install puppet

echo "Configuring puppet"
cat > /etc/puppet/puppet.conf << EOF
<%= snippet 'puppet.conf' %>
EOF

# Setup puppet to run on system reboot
```

```
/sbin/chkconfig --level 345 puppet on

/usr/bin/puppet agent --config /etc/puppet/puppet.conf -o --tags no_such_tag
<%= @host.puppetmaster.blank? ? '' : "--server #{@host.puppetmaster}" %>
--no-daemonize
<% end -%>

sync

<% if @provisioning_type == nil || @provisioning_type == 'host' -%>
# Inform the build system that we are done.
echo "Informing Satellite that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
<% end -%>
) 2>&1 | tee /root/install.post.log
exit 0

<%= section_end -%>
```

RHEV Hosted

```
<%#
kind: provision
name: Satellite Kickstart Default
oses:
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
%>
<%
  rhel_compatible = @host.operatingsystem.family == 'Redhat' &&
@host.operatingsystem.name != 'Fedora'
  os_major = @host.operatingsystem.major.to_i
  # safemode renderer does not support unary negation
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = pm_set || @host.params['force-puppet']
  section_end = (rhel_compatible && os_major <= 5) ? '' : '%end'
%>
install
<%= @mediapath %>
lang en_US.UTF-8
selinux --enforcing
keyboard us
skipx
network --bootproto <%= @static ? "static --ip=#{@host.ip}
--netmask=#{@host.subnet.mask} --gateway=#{@host.subnet.gateway}
--nameserver=#{[@host.subnet.dns_primary,@host.subnet.dns_secondary].reject{
|n| n.blank?}.join(',')}" : 'dhcp' %> --hostname <%= @host %>
rootpw --iscrypted <%= root_pass %>
firewall --<%= os_major >= 6 ? 'service=' : '' %>ssh
authconfig --useshadow --passalgo=sha256 --kickstart
timezone --utc <%= @host.params['time-zone'] || 'UTC' %>

bootloader --location=mbr --append="nofb quiet splash=quiet" <%= grub_pass
%>
```

```
<% if os_major == 5 -%>
key --skip
<% end -%>


<% if @dynamic -%>
%include /tmp/diskpart.cfg
<% else -%>
<%= @host.diskLayout %>
<% end -%>

text
reboot

%packages --ignoremissing
yum
dhclient
ntp
wget
@Core
<%= section_end -%>

<% if @dynamic -%>
%pre
<%= @host.diskLayout %>
<%= section_end -%>
<% end -%>

%post --nochroot
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
cp -va /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
/usr/bin/chvt 1
) 2>&1 | tee /mnt/sysimage/root/install.postnochroot.log
<%= section_end -%>

%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '10.16.255.2' %>
/usr/sbin/hwclock --systohc
systemctl enable ntpd

<%= snippet "subscription_manager_registration" %>

<% if @host.respond_to?(:realm) && @host.respond_to?(:otp) && @host.realm &&
@host.otp && @host.realm.realm_type == "Red Hat Identity Management" -%>
<%= snippet "idm_register" %>
```

```
<% end -%>

# update all the base packages from the updates repository
yum -t -y -e 0 update
yum group install -y 'Server with GUI'
yum install -y ovirt-hosted-engine-setup

<% if puppet_enabled %>
# and add the puppet package
yum -t -y -e 0 install puppet

echo "Configuring puppet"
cat > /etc/puppet/puppet.conf << EOF
<%= snippet 'puppet.conf' %>
EOF

# Setup puppet to run on system reboot
/sbin/chkconfig --level 345 puppet on

/usr/bin/puppet agent --config /etc/puppet/puppet.conf -o --tags no_such_tag
<%= @host.puppetmaster.blank? ? '' : "--server #{@host.puppetmaster}" %>
--no-daemonize
<% end -%>

sync

<% if @provisioning_type == nil || @provisioning_type == 'host' -%>
# Inform the build system that we are done.
echo "Informing Satellite that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
<% end -%>
) 2>&1 | tee /root/install.post.log
exit 0

<%= section_end -%>
```

RHEL OSP

```
<%#
kind: provision
name: RHEL OSP
oses:
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
%>
<%
  rhel_compatible = @host.operatingsystem.family == 'Redhat' &&
@host.operatingsystem.name != 'Fedora'
  os_major = @host.operatingsystem.major.to_i
  # safemode renderer does not support unary negation
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = pm_set || @host.params['force-puppet']
  section_end = (rhel_compatible && os_major <= 5) ? '' : '%end'
%>
```

```
install
<%= @mediapath %>
lang en_US.UTF-8
selinux --enforcing
keyboard us
skipx
network --bootproto <%= @static ? "static --ip=#{@host.ip}
--netmask=#{@host.subnet.mask} --gateway=#{@host.subnet.gateway}
--nameserver=#{[@host.subnet.dns_primary,@host.subnet.dns_secondary].reject{
|n| n.blank?}.join(',')}" : 'dhcp' %> --hostname <%= @host %>
rootpw --iscrypted <%= root_pass %>
firewall --<%= os_major >= 6 ? 'service=' : '' %>ssh
authconfig --useshadow --passalgo=sha256 --kickstart
timezone --utc <%= @host.params['time-zone'] || 'UTC' %>

bootloader --location=mbr --append="nofb quiet splash=quiet" <%= grub_pass
%>
<% if os_major == 5 -%>
key --skip
<% end -%>


<% if @dynamic -%>
%include /tmp/diskpart.cfg
<% else -%>
<%= @host.diskLayout %>
<% end -%>

text
reboot

%packages --ignoremissing
yum
dhclient
ntp
wget
@Core
<%= section_end -%>

<% if @dynamic -%>
%pre
<%= @host.diskLayout %>
<%= section_end -%>
<% end -%>

%post --nochroot
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
cp -va /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
/usr/bin/chvt 1
) 2>&1 | tee /mnt/sysimage/root/install.postnochroot.log
<%= section_end -%>
```

```
%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '10.16.255.2' %>
/usr/sbin/hwclock --systohc

<%= snippet "rhel_osp_sub_man" %>

<% if @host.respond_to?(:realm) && @host.respond_to?(:otp) && @host.realm &&
@host.otp && @host.realm.realm_type == "Red Hat Identity Management" -%>
<%= snippet "idm_register" %>
<% end -%>

# update all the base packages from the updates repository
yum -t -y -e 0 update

#<% if puppet_enabled %>
# and add the puppet package
#yum -t -y -e 0 install puppet

#echo "Configuring puppet"
#cat > /etc/puppet/puppet.conf << EOF
#<%= snippet 'puppet.conf' %>
#EOF

# Setup puppet to run on system reboot
#/sbin/chkconfig --level 345 puppet on

#/usr/bin/puppet agent --config /etc/puppet/puppet.conf -o --tags
no_such_tag <%= @host.puppetmaster.blank? ? '' : "--server
#{@host.puppetmaster}" %> --no-daemonize
#<% end -%>

sync

<% if @provisioning_type == nil || @provisioning_type == 'host' -%>
# Inform the build system that we are done.
echo "Informing Satellite that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
<% end -%>
) 2>&1 | tee /root/install.post.log
exit 0

<%= section_end -%>
```

OSE

```
<%#
kind: provision
name: Satellite Kickstart Default
```

```
oses:
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
%>
<%
  rhel_compatible = @host.operatingsystem.family == 'Redhat' &&
@host.operatingsystem.name != 'Fedora'
  os_major = @host.operatingsystem.major.to_i
  # safemode renderer does not support unary negation
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = pm_set || @host.params['force-puppet']
  section_end = (rhel_compatible && os_major <= 5) ? '' : '%end'
%>
install
<%= @mediapath %>
lang en_US.UTF-8
selinux --enforcing
keyboard us
skipx
network --bootproto <%= @static ? "static --ip=#{@host.ip}
--netmask=#{@host.subnet.mask} --gateway=#{@host.subnet.gateway}
--nameserver=#{[@host.subnet.dns_primary,@host.subnet.dns_secondary].reject{
|n| n.blank?}.join(',')}" : 'dhcp' %> --hostname <%= @host %>
rootpw --iscrypted <%= root_pass %>
firewall --<%= os_major >= 6 ? 'service=' : '' %>ssh
authconfig --useshadow --passalgo=sha256 --kickstart
timezone --utc <%= @host.params['time-zone'] || 'UTC' %>

bootloader --location=mbr --append="nofb quiet splash=quiet" <%= grub_pass
%>
<% if os_major == 5 -%>
key --skip
<% end -%>


<% if @dynamic -%>
%include /tmp/diskpart.cfg
<% else -%>
<%= @host.diskLayout %>
<% end -%>

text
reboot

%packages --ignoremissing
yum
dhclient
ntp
wget
@Core
<%= section_end -%>

<% if @dynamic -%>
%pre
```

```
<%= @host.diskLayout %>
<%= section_end -%>
<% end -%>

%post --nochroot
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
cp -va /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
/usr/bin/chvt 1
) 2>&1 | tee /mnt/sysimage/root/install.postnochroot.log
<%= section_end -%>

%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '10.16.255.2' %>
/usr/sbin/hwclock --systohc
chkconfig ntpd on

<%= snippet "subscription_manager_registration" %>

<% if @host.respond_to?(:realm) && @host.respond_to?(:otp) && @host.realm &&
@host.otp && @host.realm.realm_type == "Red Hat Identity Management" -%>
<%= snippet "idm_register" %>
<% end -%>

# update all the base packages from the updates repository
yum -t -y -e 0 update

<% if puppet_enabled %>
# and add the puppet package
yum -t -y -e 0 install puppet

echo "Configuring puppet"
cat > /etc/puppet/puppet.conf << EOF
<%= snippet 'puppet.conf' %>
EOF

# Setup puppet to run on system reboot
/sbin/chkconfig --level 345 puppet on

/usr/bin/puppet agent --config /etc/puppet/puppet.conf -o --tags no_such_tag
<%= @host.puppetmaster.blank? ? '' : "--server #{@host.puppetmaster}" %>
--no-daemonize
<% end -%>

sync
```

```
<% if @provisioning_type == nil || @provisioning_type == 'host' -%>
# Inform the build system that we are done.
echo "Informing Satellite that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
<% end -%>
) 2>&1 | tee /root/install.post.log
exit 0

<%= section_end -%>
```

rhel_osp_sub_man (snippet)

```
<% if @host.params['kt_org'] && @host.params['kt_activation_keys'] %>
# add subscription manager
yum -t -y -e 0 install subscription-manager
rpm -ivh <%= subscription_manager_configuration_url %>

echo "Registering the System"
subscription-manager register --org="<%= @host.params['kt_org']%>" --name="<
%= @host.name %>" --activationkey="<%= @host.params['kt_activation_keys']
%>"

<% if @host.content_source %>
  subscription-manager config --rhsm.baseurl=https://<%=
@host.content_source.hostname %>/pulp/repos
<% end %>

<% if @host.operatingsystem.name == "RedHat" %>
  # add the rhel rpms to install katello agent
  subscription-manager repos --enable=rhel-*-rh-common-rpms
<% end %>

echo "Installing Katello Agent"
yum -t -y -e 0 install katello-agent
chkconfig goferd on

echo "Installing RHEL OSP Installer"
yum -t -y -e 0 install rhel-osp-installer
<% end %>
```

Appendix G: Troubleshooting

G.1 Time Synchronization

For the reference environment it is critical to synchronize all systems to a central time server. Example */etc/ntpd.conf*:

```
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5),
ntp_mon(5).

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
```

```
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 10.16.255.2

#broadcast 192.168.1.255 autokey  # broadcast server
#broadcastclient            # broadcast client
#broadcast 224.0.1.1 autokey      # multicast server
#multicastclient 224.0.1.1        # multicast client
#manycastserver 239.255.254.254         # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8

# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats

# Disable the monitoring facility to prevent amplification attacks using
ntpdc
# monlist command when default restrict does not include the noquery flag.
See
# CVE-2013-5211 for more details.
# Note: Monitoring will not be disabled with the limited restriction flag.
disable monitor
```

Note: All RHEL 7 based systems are converted to utilize the Network Time Protocol daemon.

G.2 Log files

CloudForms

File	Purpose
evm.log	Main CFME log file. Very verbose and all events are written to this file.
automate.log	Automate events are logged to this file.
audit.log	Security related events are logged to this file.
miqconsole.log	CFME appliance events are logged to this file. Ex. power off or on.
miq_ntpdate.log	Time synchronization events are logged to this file.
policy.log	Policy related events are logged to this file.
prince.log	Events related to report conversions are logged to this file.
production.log	Rails and application events are logged to this file.
rhevm.log	RHEV related events are logged in this file.
top_output.log	Periodic top output is logged to this file.
vim.log	VMware related events are logged in this file.
vmdb_restart.log	Events related to restarting the VMDB database are logged in this file.
vmstat_output.log	Periodic virtual memory statistics are logged to this file.

Table G.2-1: CloudForms Log Files

Satellite 6

https://access.redhat.com/solutions/1155573

RHEL OSP

/var/log/<service_name> (nova, glance, heat, cinder, etc.)

OpenShift Enterprise

https://access.redhat.com/documentation/en-US/OpenShift_Enterprise/2/html/Troubleshooting_Guide/index.html

RHEV

https://access.redhat.com/solutions/17587

IdM

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/server-config.html#logging