Red Hat Reference Architecture Series

Red Hat Cloud Infrastructure:

Managing Red Hat Enterprise Linux OpenStack Platform 4.0 Using Red Hat CloudForms 3.0

Brett Thurber, Principal Software Engineer

RHCA, RHCVA

Version 1.0

March 2014

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/

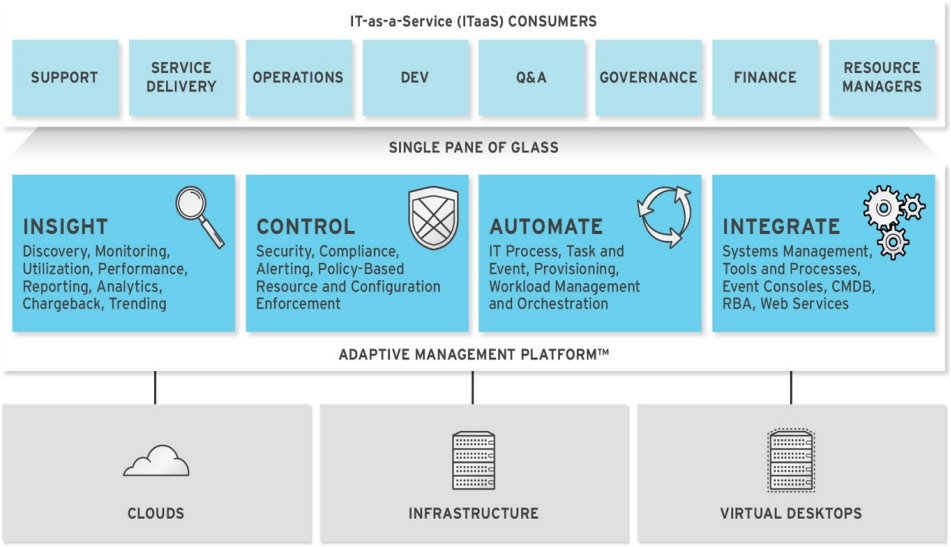
Table of Contents

1 Executive Summary

As the enterprise landscape continues to evolve and change, many IT leaders are faced with critical choices to meet current and future needs. One very important need always finds it's way to the top, how to optimize the management of their environment while reliably providing services to end-users.

Red Hat CloudForms 3.0 provides a set of Infrastructure-as-a-Service (IaaS) capabilities to orchestrate and manage both private and hybrid cloud environments helping to optimize existing infrastructure and plan for environment expansion using a comprehensive management platform. Capabilities include:

- Automation
- Provisioning
- Reporting
- Discovery
- Trending
- Compliance
- Alerting
- Utilization
- 3rd party integration



Although Red Hat CoudForms 3.0 offers almost limitless capabilities, the focus of this reference architecture is to demonstrate management and integration with Red Hat Enterprise Linux OpenStack Platform (RHEL OSP) 4.0, in a private cloud setting, targeting four detailed use cases:

- Import the CloudForms Management Engine appliance into RHEL OSP for discovery and management
- Create and deploy a Linux, Apache, MySQL, and PHP (LAMP) stack from the Service Catalog as a self-service user
- Demonstrate using Control Policies for instance management and automation
- Demonstrate RHEL OSP metering integration with Capacity and Utilization to include chargeback for managed resources

Disclaimer: Some features of this reference architecture may not be supported by Red Hat Global Support.

2 Components Overview

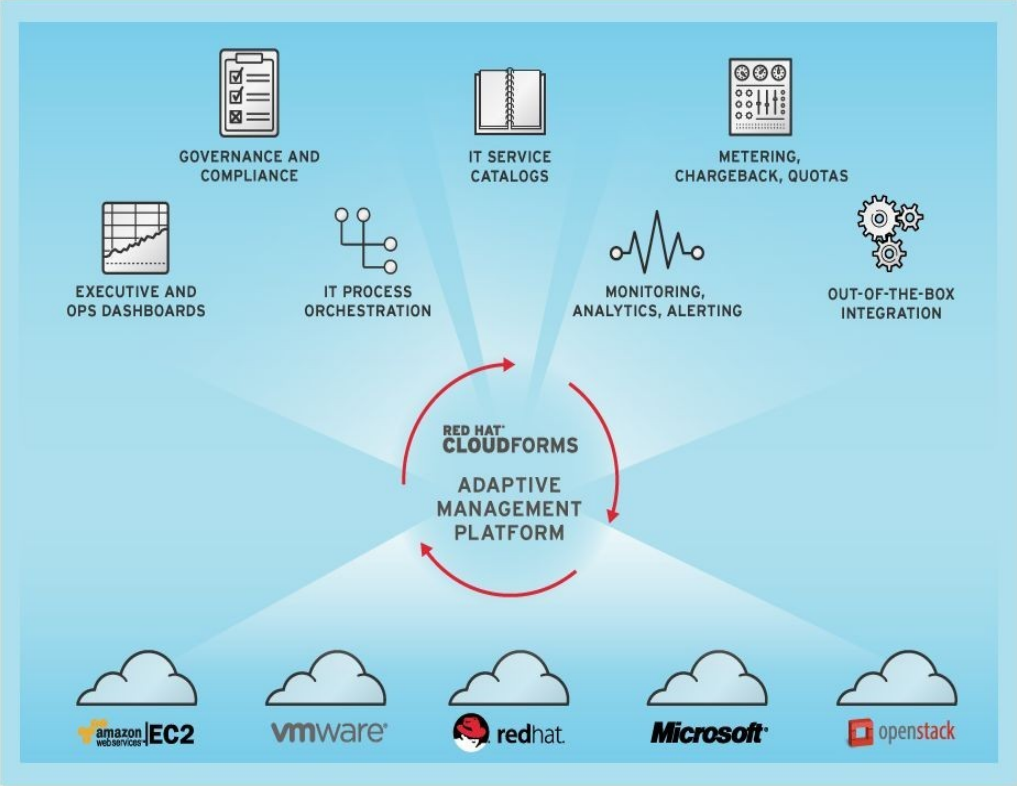
2.1 Red Hat CloudForms 3.0

A Continuum of Management Capability:

Whether you are focused on gaining control of your virtualization environment or seeking to put management capabilities in place to operate a private or hybrid cloud, CloudForms can meet your needs today with a comprehensive management platform to do both. The goal is to future proof your investment and eliminate multiple disparate tool sprawl which introduces problems of integration, multiple interfaces, and rising costs & training needs with multiple vendor point products. CloudForms allows organizations to address virtual environment problems like monitoring, tracking, capacity management, resource utilization/optimization, VM lifecycle management, and policies to govern access and usage, while allowing you to evolve, at your pace, to a private or hybrid cloud model without future management investment. If and when you want to operate a cloud model, CloudForms delivers self-service cataloging with policy-based control to agilely manage requests. We provide a single pane of glass across multiple virtualization providers, public cloud; giving you choice among providers and allowing you to leverage existing platform investments or introduce new more cost effective ones. CloudForms also equips you for quota enforcement, usage, chargeback and cost allocation, allowing you to truly evolve to *IT as a Service* (ITaaS). CloudForms provides all these capabilities with dashboards, reports, policies, approval workflows and alerts, to ensure you remain in control.

Virtual Environment	Private/Hybrid Cloud	Operational
• Monitoring/Tracking • Capacity Management/Planning • Resource Utilization/Optimization • VM Lifecycle Management • Policies to Govern Access/Usage	• Self-Service Portal/Catalog • Controls to Manage Requests • Quota Enforcement/Usage • Chargeback/Cost Allocation • Automated Provisioning	• Dashboards • Reports • Policies • Alerts • Approval Workflows

Figure 2.1-1: CloudForms 3.0 Capabilities provides a depiction of CloudForms 3.0 capabilities and features.



1. **Figure 2.1-1: CloudForms 3.0 Capabilities**

Key Product Messages:

- **Seamless user self-service portals** support service catalogs with role-delegated automated provisioning, quota enforcement and chargeback across Red Hat Enterprise Virtualization and other hypervisor and cloud platforms including VMware, Microsoft, OpenStack, and Amazon EC2.
- **Automated policy enforcement/control** for managed systems to reduce the delay, effort, cost and potential for errors involved in manually enforcing policies and changing system configurations or allocation of resources, while helping to assure security and compliance.
- **Executive management and governance** with comprehensive dashboards and reporting, policy-based standards enforcement, financial management, capacity forecasting, trend analysis, health, and availability of managed environments.
- **Intelligent workload management** to ensure resources are automatically and optimally utilized for service availability and performance. This includes policy-based orchestration of workloads and resources, the ability to simulate allocation of resources for “what/if” planning and continuous insights into granular workload along with consumption levels to allow chargeback, showback, proactive planning, and policy

creation.

- **Capacity Planning** to anticipate and plan for future resource needs based on capacity, trending, data, and analytics. This includes the abilities to classify resources based on configuration, performance, capacity, cost, acceptable use, and locations.
- **Capacity Management** to dynamically and automatically assure the most efficient use of resources. This includes the ability to discover and track resource changes, provision and de-provision resources based on policies and demand, and identify the current condition of resources along with “best fit” for new workloads across compute, storage, and network resources.
- **Federated management** of large and distributed infrastructures from a single pane of glass, enabling enterprises to rapidly scale out their virtual and cloud deployments.
- **Red Hat Enterprise Virtualization certified** and optimized to run on Red Hat Enterprise Linux.

Key Business Messages:

- **Low Acquisition Cost** less than ½ the cost of other management technologies, over a 3 year period. Efficient subscription model allows for more of an annual rental model – no large upfront licensing fees. Tool consolidation and replacement can result in a zero net sum or even saving on current management spend.
- **Fastest Time to Value** installs rapidly as a Virtual Appliance with no agents to license or maintain. Value is seen in hours versus days/weeks through auto-discovery of your environment. Operations specific use case implementations happen in days not months. One management system to learn vs. multiple tools/interfaces which result in higher support costs and greater potential for vendor price hikes. Can integrate as needed with larger management systems (BMC, CA, HP, Microsoft, ServiceNow).
- **Increased Automation and Continuous Optimization** through policy based controls and automated responses allows you to maximize resource efficiency and control of IT capital costs by adaptively increasing utilization. Increased automation supports higher operational efficiency in build, test and automate sequences. Significantly reduces human interaction/errors and gain ability to manage with less people/labor costs.
- **Open/Flexible solution** prevents vendor lock-in and allows for choice of infrastructure by leveraging low cost virtual platforms for your cloud. Leverages existing IT investments and supports seamless introduction of new lower cost platform alternatives. Choice among VMware, Red Hat, & Microsoft for virtualization platforms, Amazon as public cloud, and Red Hat Enterprise Linux OpenStack Platform for private clouds.
- **More efficient users and customers** through self service with web-based portals accompanied by fine-grained access control and support for request management, tracking and approval.

2.2 What is OpenStack?

OpenStack[1] is an open source project for building a private or public Infrastructure-as-a-Service (IaaS) cloud running on standard hardware. The typical analogy is that OpenStack gives you public cloud-like capabilities in your datacenter.

OpenStack is a cloud operating system that controls all the infrastructure—compute, storage, and networking—resources throughout a datacenter, all managed through a central dashboard.

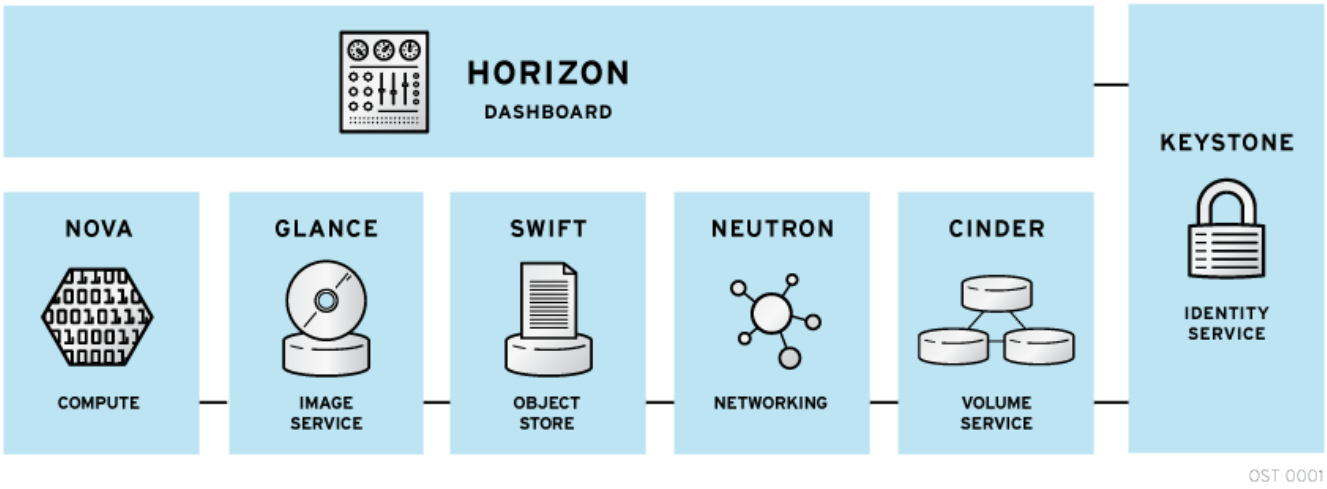


Figure 2.2.1-1: Red Hat Enterprise Linux OpenStack Platform Resources

2.2.1 Red Hat Enterprise Linux OpenStack Platform 4.0

Red Hat Enterprise Linux OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable public or private OpenStack cloud. It delivers a cloud platform built on Red Hat Enterprise Linux, optimized for and integrated with Red Hat's OpenStack technologies, giving you the agility to scale and quickly meet customer demands without compromising on availability, security, or performance. Additionally, when combined with the Red Hat Storage Server[2] add-on, you gain an integrated and distributed highly available cloud storage platform across public and private clouds.

Red Hat Enterprise Linux OpenStack Platform 4 is built from the combination of Red Hat Enterprise Linux 6.5 and the latest Red Hat OpenStack technology, which is an enterprise-hardened version of the community “Havana” release. This version boasts all the community core features and functions, as well as some additional innovations by Red Hat.

Specifically, Red Hat Enterprise Linux OpenStack Platform 4 now includes OpenStack orchestration (project “Heat”), which provides template-based orchestration for provisioning infrastructure resources. It also includes OpenStack telemetry (project “Ceilometer”), which collects and stores metering and usage data and makes that data available via APIs to custom billing systems. Additionally, Red Hat Enterprise Linux OpenStack Platform 4 now

1 http://www.openstack.org/
2 http://www.redhat.com/products/storage-server/

utilizes the highly scalable Foreman provisioning tool for enterprise deployments, which includes bare-metal provisioning for new compute nodes. Finally, in an effort to provide a more seamless infrastructure to customers, it is now integrated with additional Red Hat infrastructure and cloud management tools, including Red Hat CloudForms for system-wide management and Red Hat Storage to provide optional object, block, and image storage services.

Key Benefits:

- **Integrated and optimized:** OpenStack is dependent on the underlying Linux kernel for performance, security, hardware enablement, networking, storage, and other basic servers. Red Hat Enterprise Linux OpenStack Platform delivers an OpenStack distribution with the performance, security, and stability of Red Hat Enterprise Linux, allowing you to focus on delivering the services your customers want instead of the underlying operating platform.

- **Take advantage of broad application support.** Red Hat Enterprise Linux running as guest virtual machines provides a stable application development platform with many independent software vendor (ISV) certifications so you can rapidly build and deploy your cloud applications.

- **Improve service-level agreements (SLAs)** with faster IT service delivery. Red Hat Enterprise Linux OpenStack Platform delivers a Red Hat OpenStack distribution with the performance, security, scalability, and reliability of Red Hat Enterprise Linux.

- **Avoid vendor lock-in** by moving to open technologies while maintaining your existing infrastructure investments.

- **Largest partner ecosystem:** Red Hat has assembled the worlds largest ecosystem of certified partners for OpenStack compute, storage, networking, ISV software, and services for Red Hat Enterprise Linux OpenStack Platform deployments, ensuring the same level of broad support and compatibility customers enjoy today in the Red Hat Enterprise Linux ecosystem.

- **Bring security to the cloud.** Rely on the SELinux military-grade security and container technologies of Red Hat Enterprise Linux to prevent intrusions and protect your data when running in public or private clouds.

- **Enterprise hardened:** Red Hat tests, fixes, and certifies each OpenStack release running on Red Hat Enterprise Linux OpenStack Platform for broad hardware and software compatibility and performance.

- **Enterprise life cycle:** Red Hat Enterprise Linux OpenStack Platform provides a stable branch release of OpenStack and Linux which can be supported by Red Hat for extended life cycles beyond the six-month release cycle of OpenStack. Security fixes, bug fixes, performance enhancements, and some features can be back-ported from future releases without disrupting production environments.

3 Environment

The following figure depicts the major components used in the reference environment to include:

Two CloudForms Management Engine appliances, one Red Hat Enterprise Linux OpenStack Platform environment utilizing iSCSI storage for Cinder volumes, one Red Hat Enterprise Virtualization environment utilizing iSCSI storage for Data domains, one Satellite Server to support DNS services and systems management, and a Windows Active Directory server supporting secure LDAP services for users and groups integrated within CloudForms.

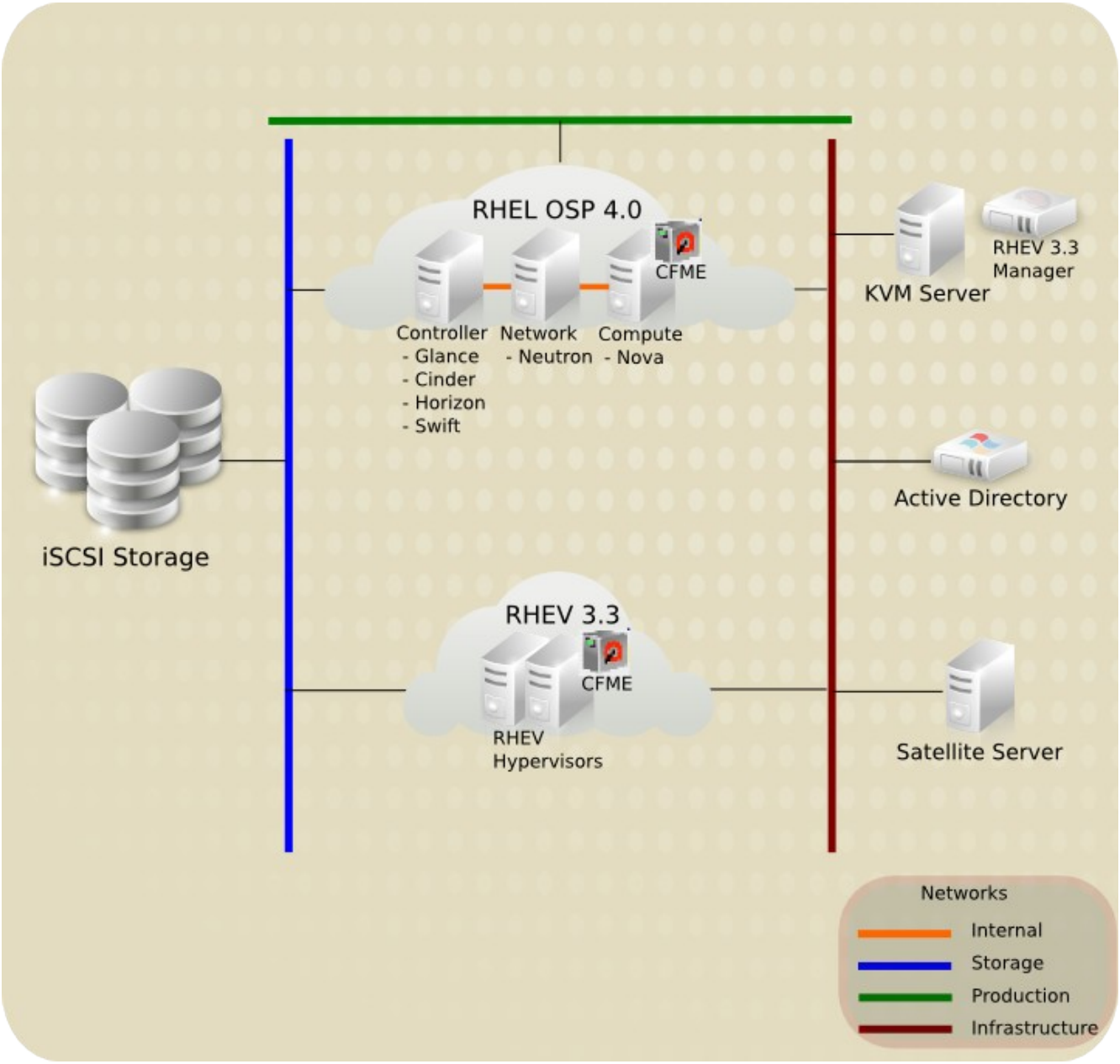


Figure 3-1: Reference Architecture Components

3.1 Software

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

3.1.1 CloudForms Management Engine

The following table lists the software version used for CloudForms 3.0.

System	Software Version	Role(s)
CloudForms Management Engine (cf-cfme-rhev, cf-cfme-rhelosp)	5.2.2.3.20140303163147_1db30fd	Orchestration and Management

Table 3.1.1-1: CFME – Software Versions

3.1.2 Microsoft Windows

The following table lists the software version used for Windows Server 2008 R2.

System	Software Version	Role(s)
Windows Server 2008 (cf-win-ad)	6.1.7601	Active Directory LDAP Authentication

Table 3.1.2-1: Windows Server – Software Versions

3.1.3 Red Hat Satellite Server

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

System	Software Version	Role(s)
Satellite Server (sysman-rhel6)	5.6.0	DNS, DHCP, Provisioning

Table 3.1.3-1: Satellite Server – Software Versions

3.1.4 Red Hat Enterprise Linux OpenStack Platform

The following table lists the software version used for RHEL OSP 4.0.

Systems	Software	Version
Controller (cf-rhos-glance) Glance, Cinder, Horizon, Ceilometer	qpid-cpp-server	0.14-22
	mysql-server	5.1.73-3
	openstack-packstack	2013.2.1-0.22
	openstack-keystone	2013.2.1-1
	openstack-nova-{api,cert,common,conductor,scheduler,console}	2013.2.1-2
	openstack-nova-novncproxy	2013.2.1-2
	openstack-glance	2013.2.1-3
	openstack-cinder	2013.2.1-5
	openstack-dashboard	2013.2.1-1
	openstack-utils	2013.2-3
	openstack-selinux	0.1.3-2
	openstack-ceilometer-{common,collector,central,alarm,api}	2013.2.1-2
Compute (cf-rhos-nova-1) Nova	openstack-nova-{common,compute}	2013.2.1-2
	openstack-utils	2013.2-3
	openstack-selinux	0.1.3-2
	openvswitch	1.11.0-1
	openstack-neutron-openvswitch	2013.2.1-4
	openstack-neutron	2013.2.1-4
Network (cf-rhos-neutron) Neutron	openstack-neutron-openvswitch	2013.2.1-4
	openstack-neutron	2013.2.1-4
	openstack-utils	2013.2-3
	openstack-selinux	0.1.3-2
	openvswitch	1.11.0-1

Table 3.1.4-1: RHEL OSP – Software Versions

3.1.5 Red Hat Enterprise Virtualization

The reference environment utilizes a single Red Hat Enterprise Virtualization 3.3 Management System and two Hosts.

Systems	Configuration	Software Versions	Role(s)
Red Hat Enterprise Virtualization (cf-rhevm-32)	RHEV 3.3 environment • Two RHEV 6.5 Hypervisor	• RHEV Manager (3.3): • 3.3.0-0.45.el6ev • RHEV Hypervisor - 6.5 - 20140112.0.el6 • VDSM Version: vdsm-4.13.2-0.6.el6ev	Virtualization

Table 3.1.5-1: RHEV Environment – Software Versions

3.1.6 Red Hat Network

The following channels are used for each Red Hat Product.

Product	Parent Channel	Child Channel(s)
Red Hat CloudForms	rhel-x86_64-server-6	rhel-x86_64-server-6-cf-me-3
Red Hat Enterprise Virtualization	rhel-x86_64-server-6	rhel-x86_64-server-6-rhevm-3.3
		rhel-x86_64-server-6-rhevh
		rhel-x86_64-rhev-agent-6-server
		jbappplatform-6-x86_64-server-6-rpm
Red Hat Enterprise Linux	rhel-x86_64-server-6	rhn-tools-rhel-x86_64-server-6
		rhel-x86_64-server-supplementary-6
		rhel-x86_64-server-rh-common-6
Red Hat Satellite Server	rhel-x86_64-server-6	redhat-rhn-satellite-5.6-server-x86_64-6
Red Hat Enterprise Linux OpenStack Platform	rhel-x86_64-server-6	rhel-x86_64-server-6-ost-4

Table 3.1.6-1: RHN Product Channels

3.2 Systems

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

3.2.1 Server Hardware

All six 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	4096 MB
Network	1 bridged virtIO
Disk	Disk 1 – 10 GB (OS) Disk 2 – 200GB (/var/satellite) Disk 3 – 20 GB (/rhnsat) Disk 4 – 30 GB (/var/lib/pgsql)

Table 3.2.2-2: Satellite Server – Virtual Machine Configuration

Microsoft Windows Server 2008 R2

Component	Details
CPU	1
Memory	4096 MB
Network	1 bridged virtIO
Disk	Disk 1 – 20 GB (OS)

Table 3.2.2-3: Windows Server – Virtual Machine Configuration

RHEV Manager

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

3.2.3 RHEL OSP Instances

The following image is used within the Red Hat Enterprise Linux OpenStack Platform environment.

LAMP Server

Component	Details
CPU	1
Memory	2048 MB
Network	Single network interface
Disk	Disk 1 – 20 GB (OS)

Table 3.2.3-1: LAMP – Instance Configuration

CloudForms Management Engine

Component	Details
CPU	2
Memory	4096 MB
Network	Single network interface
Disk	Disk 1 – 40 GB (OS) Disk 2 – 20 GB (CFME database)

Table 3.2.3-2: CFME – Instance Configuration

Note: The CloudForms Management Engine hosted within the RHEL OSP environment utilizes the *m1.medium* flavor as the hardware profile.

3.3 Storage

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

System	Disk Size
RHEL OSP Controller Node (Cinder)	488 GB
RHEV Hypervisor	488 GB
KVM Server (VM disks)	400 GB

Table 3.3-1: Storage Configuration

4 Preparing the Infrastructure

This section describes the procedures used in preparing the infrastructure. This does not include installing each component from the ground up however discusses the specific configurations used for the reference environment.

4.1 RHEL OSP Configuration

Importing and creating the CloudForms Management image and instance within a RHEL OSP environment requires several steps. Refer to *CloudForms 3.0: Management Engine 5.2 Installation Guide*[3] for required steps.

Post image upload and instance configuration for the reference environment include:

- Configuring networking
- Creating security groups
- Generating keypairs
- Cinder storage configuration

Note: At the time of this writing CloudForms 3.0 supports only the *admin* tenant. Additional tenant support will be included in future releases.

4.1.1 External Networking

The reference environment utilizes three physical networks.

- Storage – provides access to Cinder storage services
- Infrastructure – management network for infrastructure services
- Production – provides external access to production resources

4.1.2 Internal Networking

For the reference environment, two internally managed networks are configured for the RHEL OSP environment. One network provides internal access only (also known as **private**) among instances while a second network provides access to a production network (also known as **external**).

The internal (**private**) network is configured to provide DHCP services to running instances and is connected to a RHEL OSP managed internal router to provide access to the production (**external**) network.

The production (**external**) network is connected to a RHEL OSP managed internal router to provide access back into the internal (**private**) network.

3 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Installation_Guide/index.html

Additionally, a range of floating IP's is configured to assign to the CFME instance (*cf-cfme-rhelosp*) and LAMP instances as needed. Each instance uses a single **private** connection and the assigned floating IP allows the instance to communicate on the **public** network.

Refer to *Red Hat Enterprise Linux OpenStack Platform 4: End User Guide*[4] and *Red Hat Enterprise Linux OpenStack Platform 4: Installation and Configuration Guide*[5] for additional information regarding installing and configuring network services for RHEL OSP.

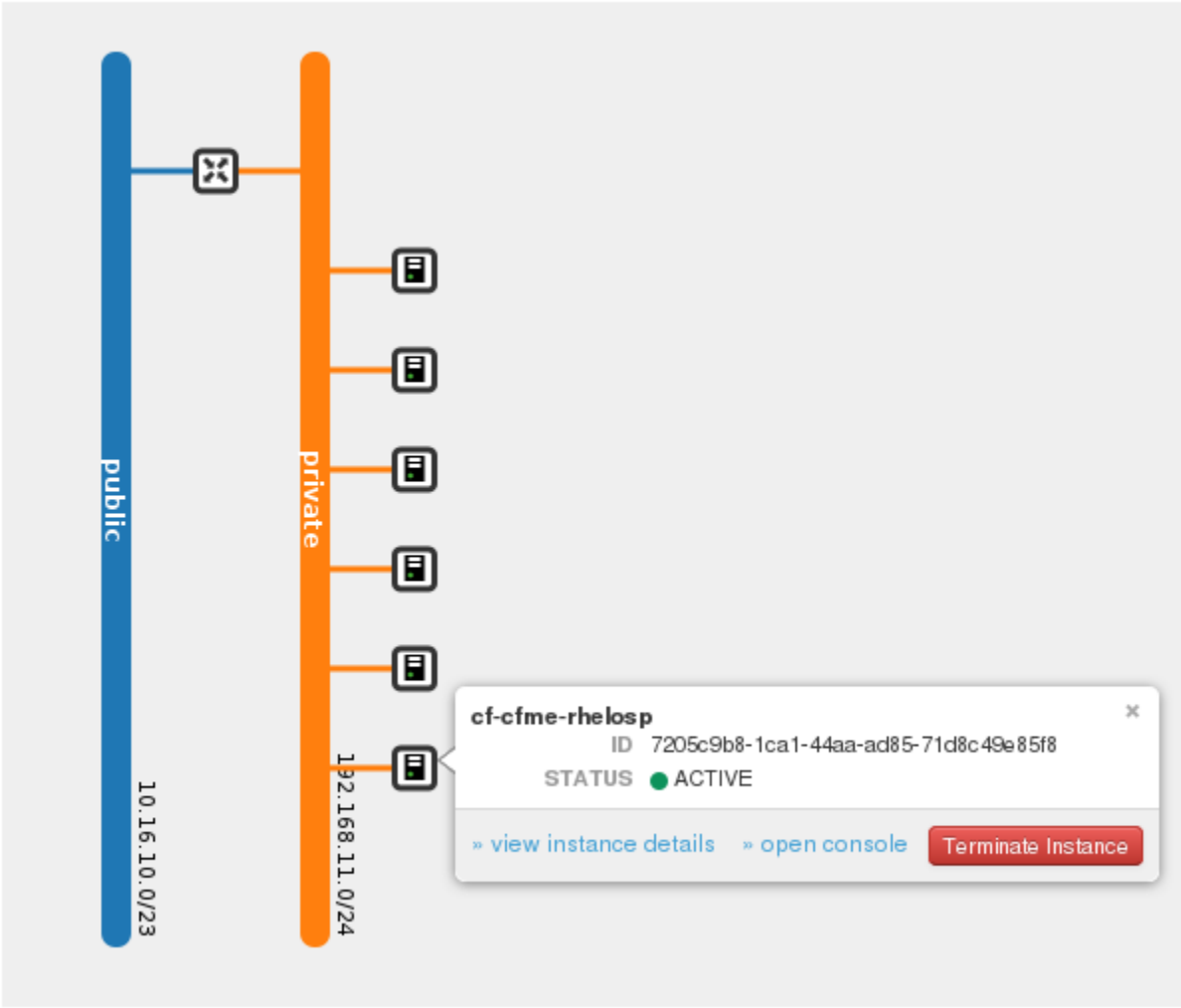


Figure 4.1.2-1: RHEL OSP Network Topology

4 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/End_User_Guide/dashboard_create_networks.html

5 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/chap-Installing_the_OpenStack_Networking_Service.html

4.1.3 Security Groups

For the reference environment, two security groups are used within the RHEL OSP environment. One security group dedicated to the CFME appliance and another security group dedicated to the LAMP instances. Refer to *Red Hat Enterprise Linux OpenStack Platform 4: End User Guide*[6] for instructions on creating and modifying Security Groups.

CFME security rules

Direction	Protocol	Port
Ingress and Egress	TCP	22
Egress	TCP	25
Egress	UDP	53
Ingress and Egress	TCP	80
Ingress and Egress	TCP	389
Ingress and Egress	TCP	443
Ingress and Egress	TCP	636
Ingress and Egress	TCP	5000
Ingress and Egress	TCP	5432
Ingress and Egress	TCP	5672
Ingress and Egress	TCP	8773
Ingress and Egress	TCP	8774
Ingress and Egress	TCP	8777
Ingress and Egress	TCP	9292
Ingress and Egress	TCP	9696
Ingress and Egress	ICMP	N/A

Table 4.1.3-1: CFME Security Group Rules

LAMP security rules

Direction	Protocol	Port
Ingress and Egress	TCP	22
Ingress and Egress	TCP	80

Table 4.1.3-2: LAMP Security Group Rules

6 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/End_User_Guide/index.html

4.1.4 Keypairs

Keypairs are an additional security option for deployed instances within RHEL OSP. Used in combination with SSH, they provide secure and direct access to running instances. For the reference environment a single keypair is created and assigned to deployed LAMP instances.

The following is an example of accessing a running instance using an assigned keypair:

```
$ ssh -i cf-lamp-rhos4.pem root@<ip_address>
```

Additional information regarding Keypairs can be found in *Red Hat Enterprise Linux OpenStack Platform 4: End User Guide*[7].

4.1.5 Cinder Storage

Volume management within RHEL OSP is provided by Cinder[8]. For the reference environment iSCSI storage is used as the back end storage hosting the volume group used named *cinder-volumes.* Total size for the volume group is 480 GB.

CloudForms Management Engine requires that a volume be created and attached to the CFME instance as part of the installation and configuration of the Virtual Management Database (VMDB) database when choosing the internal database option[9].

> **Note:** Refer to **Section Error: Reference source not found Error: Reference source not found** for specific CFME configuration steps within a RHEL OSP environment.
>
> The following bug was encountered when utilizing iSCSI storage for hosting the Cinder volume group. https://bugzilla.redhat.com/show_bug.cgi?id=1023212

7 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/End_User_Guide/cli_configure_instances.html

8 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/Image_-_Glance.html

9 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Installation_Guide/Configuring_a_Database_for_CloudForms_Management_Engine.html

4.2 Microsoft Active Directory

Microsoft Active Directory is used to provide secure LDAP user and group authentication for the reference environment. For complete end-to-end configuration refer to *Red Hat Cloud Infrastructure: Managing a RHEV 3.2 Infrastructure Using Red Hat CloudForms 2.0,* section *4.2 Microsoft Active Directory*[10].

4.3 Red Hat Satellite Server

Red Hat Satellite Server is utilized within the reference environment to provide provisioning, name resolution (DNS), IP address management (DHCP), and packages for installation and updates in support of infrastructure machines.

4.4 Importing and Configuring CloudForms Management Engine within Red Hat Enterprise Linux OpenStack Platform 4

Obtaining and importing the CloudForms Management Engine appliance is performed using the following steps:

- Obtain the appropriate CFME appliance image for RHEL OSP
- Import the CFME appliance image into RHEL OSP
- Create a new CFME instance with the VMDB database

4.4.1 Obtain CFME Image

The CFME appliance image for RHEL OSP is obtained via https://access.redhat.com.

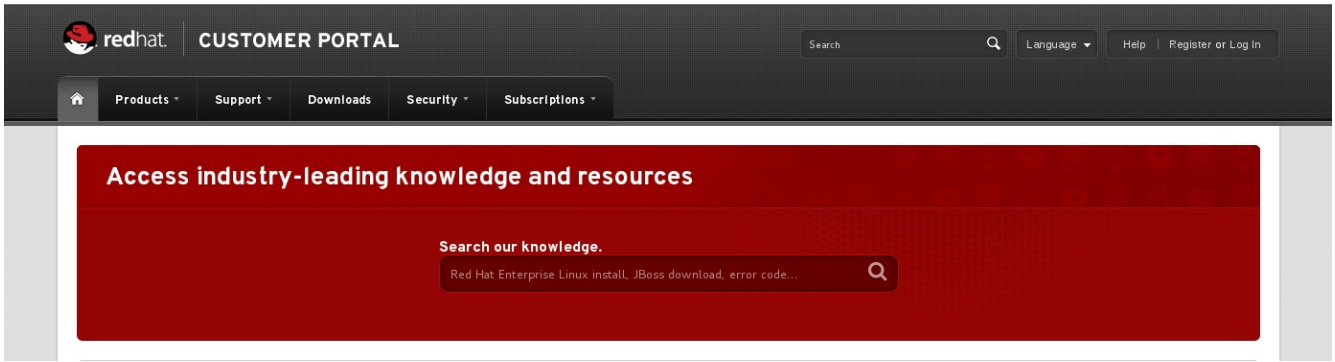


Figure 4.4.1-1: Red Hat Customer Portal

10 http://www.redhat.com/resourcelibrary/reference-architectures/managing-a-rhev-3-dot-2-infrastructure-using-red-hat-cloudforms-2

Once logged in, select the **Downloads** tab and choose **Red Hat Enterprise Linux – Download Software** button.

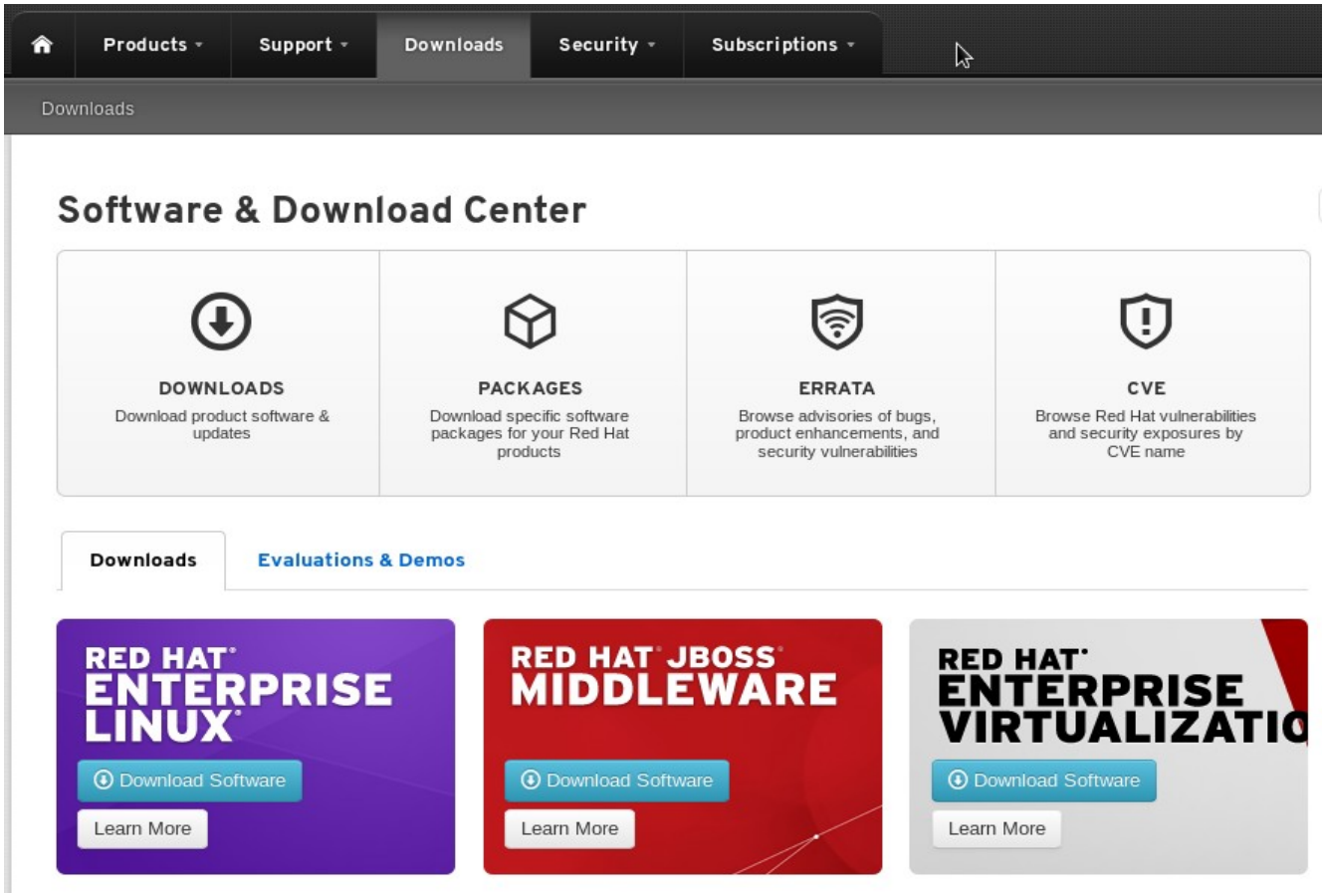


Figure 4.4.1-2: CloudForms Download

Expand *Red Hat Enterprise Linux Server (v. 6 for 64-bit x86_64)* and select *Red Hat CloudForms Management Engine (v5.2)*.

Figure 4.4.1-3: CloudForms Appliance Location

Click *CFME OpenStack Virtual Appliance* and choose a download location.

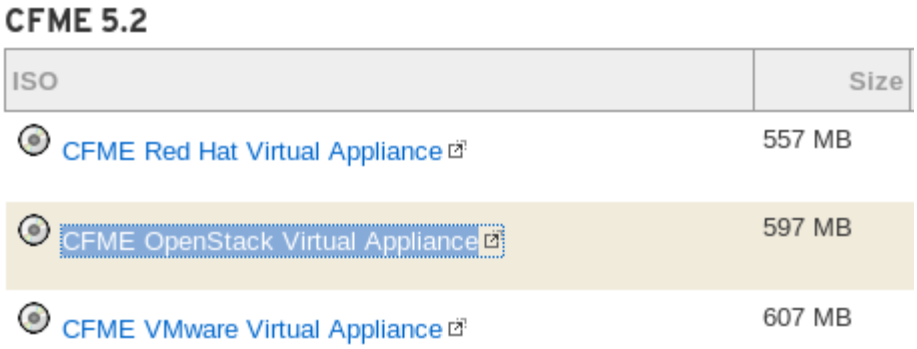


Figure 4.4.1-4: CloudForms Appliance Image

4.4.2 Importing CFME into RHEL OSP

Importing the CFME image is no different from the process used for the LAMP image described in **Section 4.5.3 Importing the LAMP image into RHEL OSP**.

Login to the RHEL OSP dashboard (http://ip_address/dashboard) with the *admin* account. Under the **Admin** tab select **Images**. Click the **Create Image** button in the upper right.

Provide a **Name**, **Image Source**, and **Format**. For the reference environment **Public** is not checked. Only the *admin* user and project have access to the image.

Click **Create Image** to initiate the image import.

Upon completion the image will appear under **Images**.

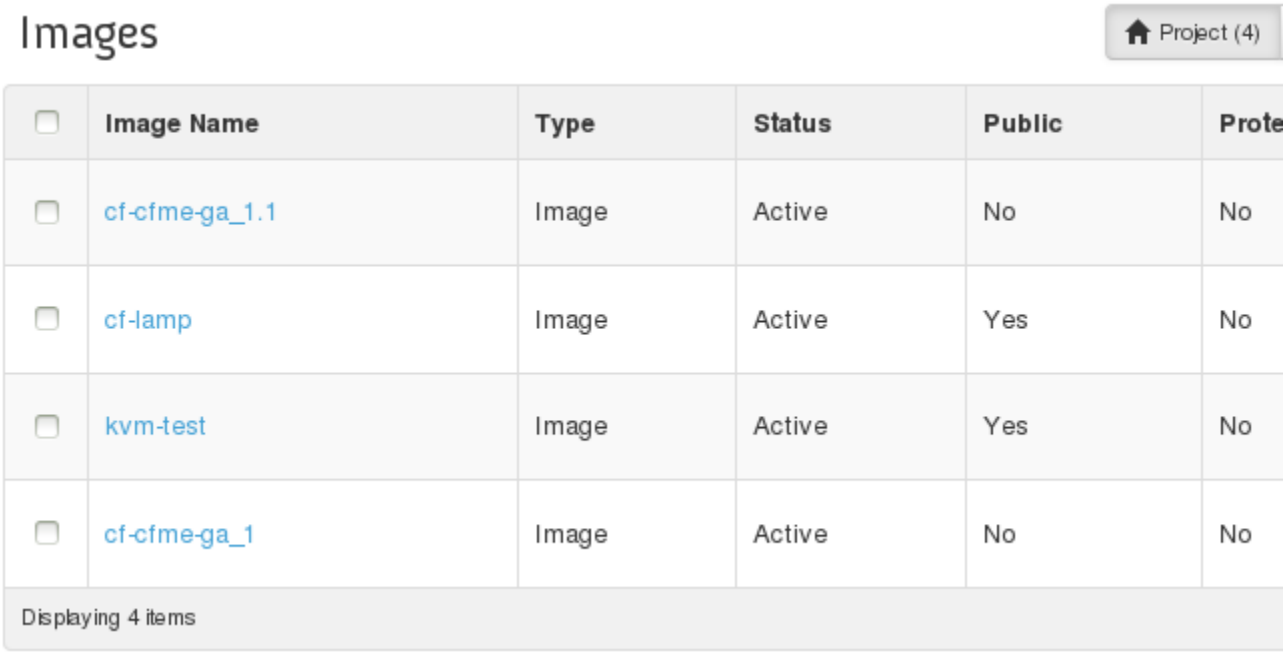


Figure 4.4.2-1: CFME Image Import Completion

4.4.3 Create a New CFME Instance with the VMDB Database

With the image upload complete, create a new CFME instance. Login to the RHEL OSP dashboard (http://ip_address/dashboard) with the *admin* account, select the **Project** tab on the left window pane and click **Instances**.

On the right window pane click the **Launch Instance** button. A series of dialog tabs appear. For the reference environment the following inputs are provided:

Tab	Option	Value
Details	Availability Zone	nova
	Instance Name	cf-cfme
	Flavor[11]	m1.medium
	Instance Count	1
	Instance Boot Source	Boot from image
	Image Name	cf-cfme-ga_1.1
Access and Security	Keypair	Select a keypair(none)
	Admin Pass	(blank)
	Confirm Admin Pass	(blank)
	Security Groups	cfme
Networking	Selected Networks	Private
Post-Creation	Customization Script	(blank)

Table 4.4.3-1: CFME Instance Creation Options

Click the **Launch** button when complete to create the CloudForms Management Engine instance.

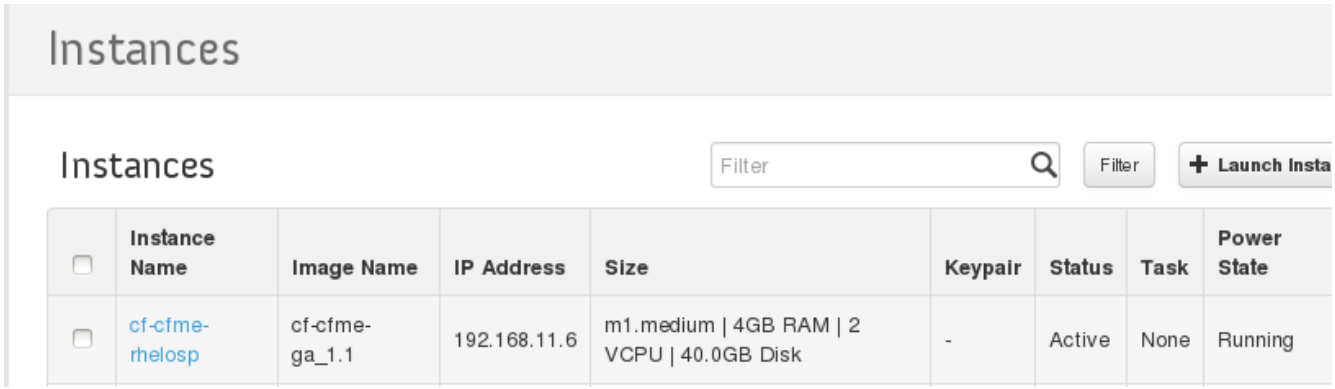


Figure 4.4.3-1: CFME Instance Running

11 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Installation_Guide/sect-Uploading_the_Appliance_on_OpenStack.html#Creating_a_Custom_Flavor_for_CloudForms_Management_Engine

As *admin*, navigate to the **Admin** tab, **Volumes** on the left window pane. On the right window pane click the **Create Volume Type**. Provide a **Name** and click the **Create Volume Type** button.

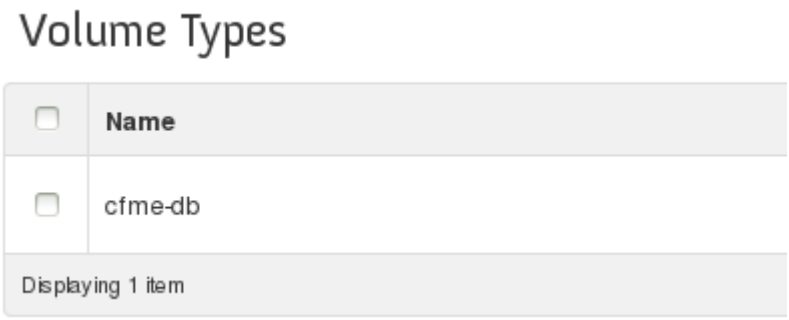


Figure 4.4.3-2: Volume Type Creation

Once the instance is running and a **Volume Type** created, make a volume to attach to the instance. Navigate to **Volumes** under the **Project** tab in the left window pane. Click the **Create Volume** button on the right window pane.

The following inputs are provided for the new volume:

Option	Value
Volume Name	vmdb
Description	VMDB database
Type	cfme-db
Size	20 GB
Volume Source	No source, empty volume.

Table 4.4.3-2: Volume Creation Settings

Attach the volume to the CFME instance by clicking the **Edit Attachments** button.

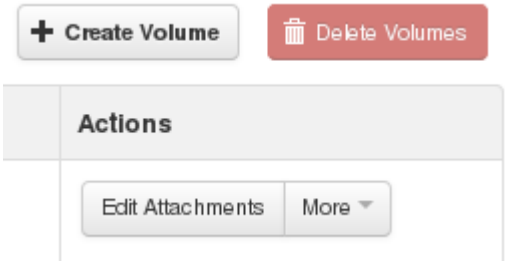


Figure 4.4.3-3: VMDB Volume Attachment

Select the newly created CFME instance (*cf-cfme-rhelosp*) under **Attach to Instance** and enter a **Device Name**. For the reference environment */dev/vdb* is entered as the **Device Name**.

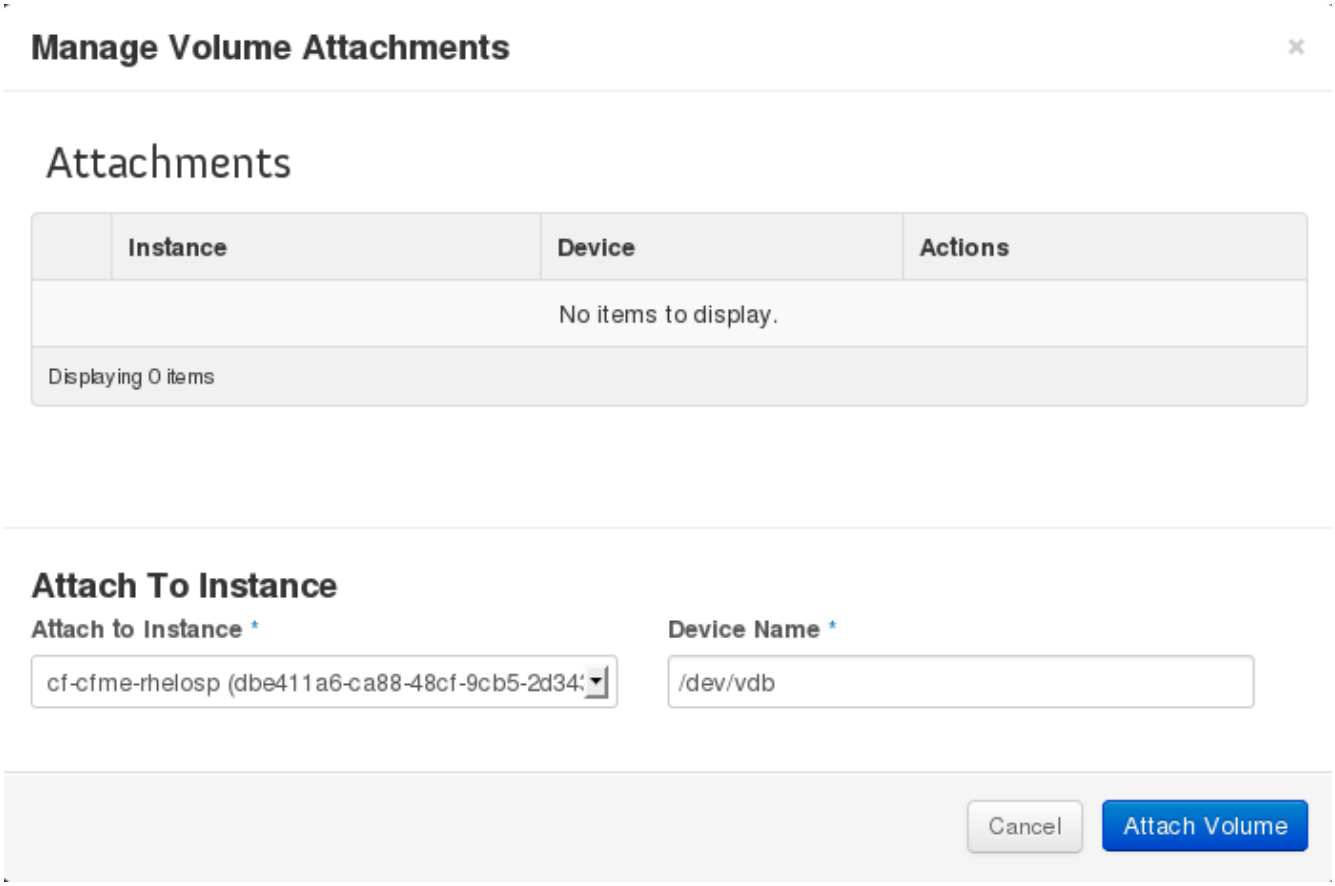


Figure 4.4.3-4: Instance Attach Volume

Complete the volume attachment by clicking **Attach Volume**.

Access the console for the CFME instance by navigating to **Instances** in the left window pane, clicking the **More** drop down menu next to the instance and select **Console**.

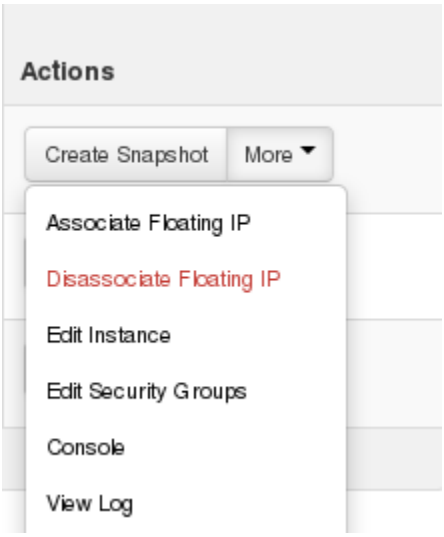


Figure 4.4.3-5: CFME Console Access

Login to the CFME appliance using the default credentials for the *admin* user. At the summary page press the `Enter` key to access configuration options. From the menu choices type the number `10) Configure Database` and press the `Enter` key once more.

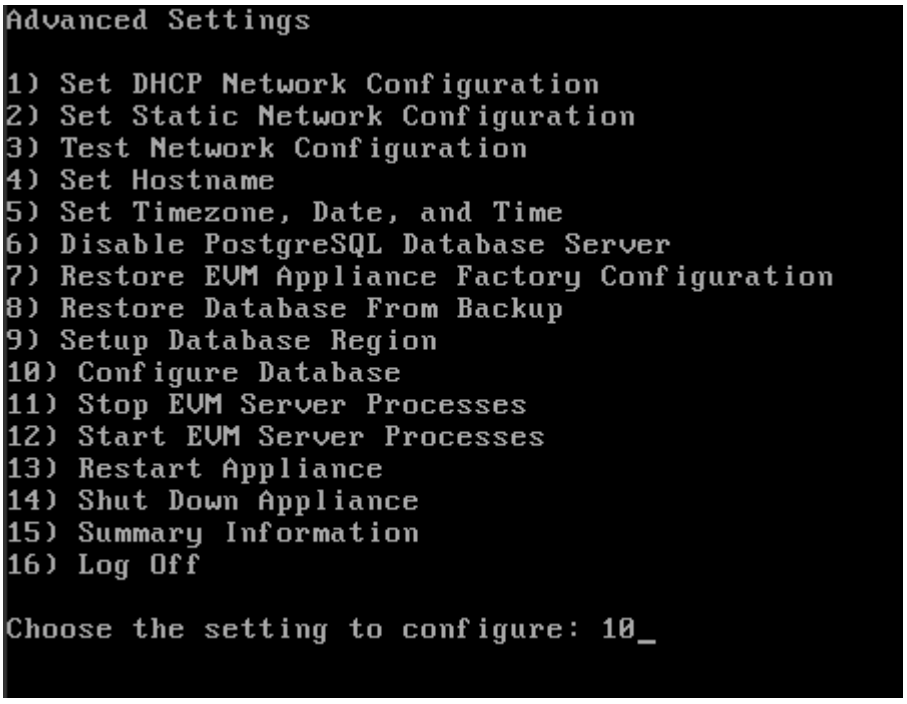


Figure 4.4.3-6: VMBD Configuration

Choose the database location. For the reference environment **Internal** is the option selected. Type the number `1) Internal` and press the `Enter` key.

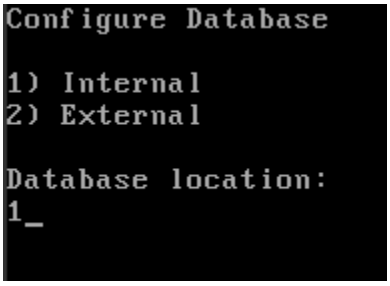


Figure 4.4.3-7: VMDB Location

Select the disk to install the VMDB to and press the `Enter` key. For the reference environment this represents the volume attached to the instance, */dev/vdb*.

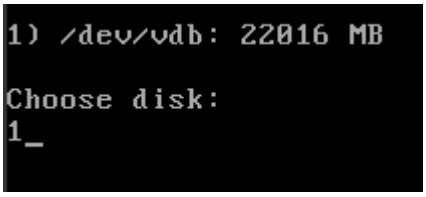


Figure 4.4.3-8: VMDB Disk

Enter the desired database region number. For the reference environment the number 0 is chosen.

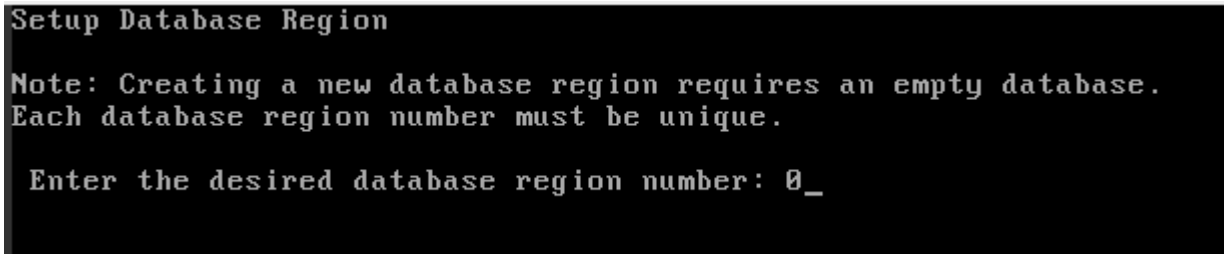


Figure 4.4.3-9: VMDB Region Number

A warning screen appears asking to proceed indicating any existing databases will be destroyed. Enter `Y` for “yes” and press the `Enter` key. A status screen appears indicating the progress for the VMDB installation.

```
Activating the configuration using the following settings...

Host:     127.0.0.1
Username: root
Database: vmdb_production
Region:   0

Initialize postgresql disk starting
Initialize postgresql disk complete
Initialize postgresql starting
Initialize postgresql complete
Create region starting
_
```

Figure 4.4.3-10: VMDB Creation Status

Upon completion the status screen indicates the configuration activated successfully and to “Press any key to continue” which returns the user back to the CFME appliance status screen.

To complete the CFME instance configuration for the reference environment, assign a floating IP address to the instance. Navigate to **Instances** on the left window pane under the **Project** tab. On the right window pane next to the instance, click the **More** drop down menu and select **Associate Floating IP**.

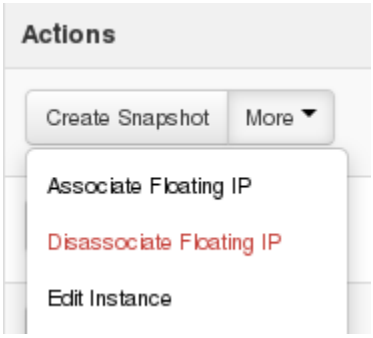


Figure 4.4.3-11: Instance Floating IP

Chose an available IP address from the drop down menu in the **IP Address** field. Click the **Associate** button to complete the assignment.

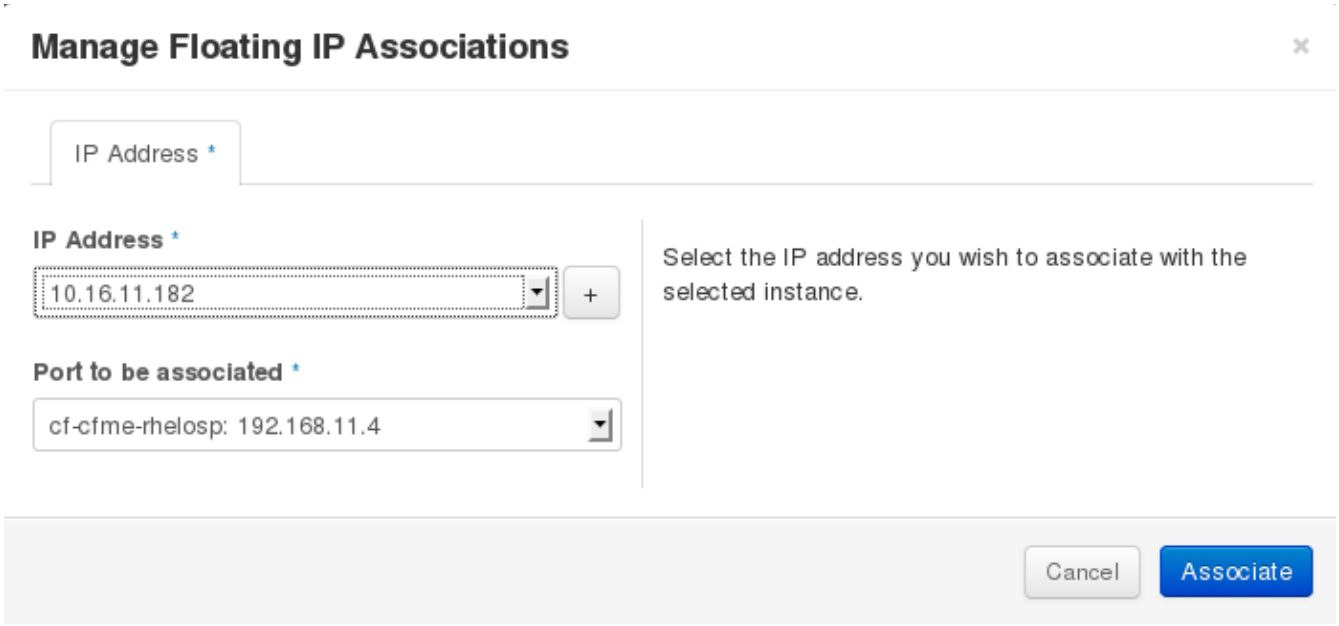


Figure 4.4.3-12: Instance Floating IP Assignment

The purpose of assigning a floating IP address allows the CFME instance the ability to communicate on the Production network.

After a few minutes the CFME console is available by navigating to the floating IP address assigned on the Production network.

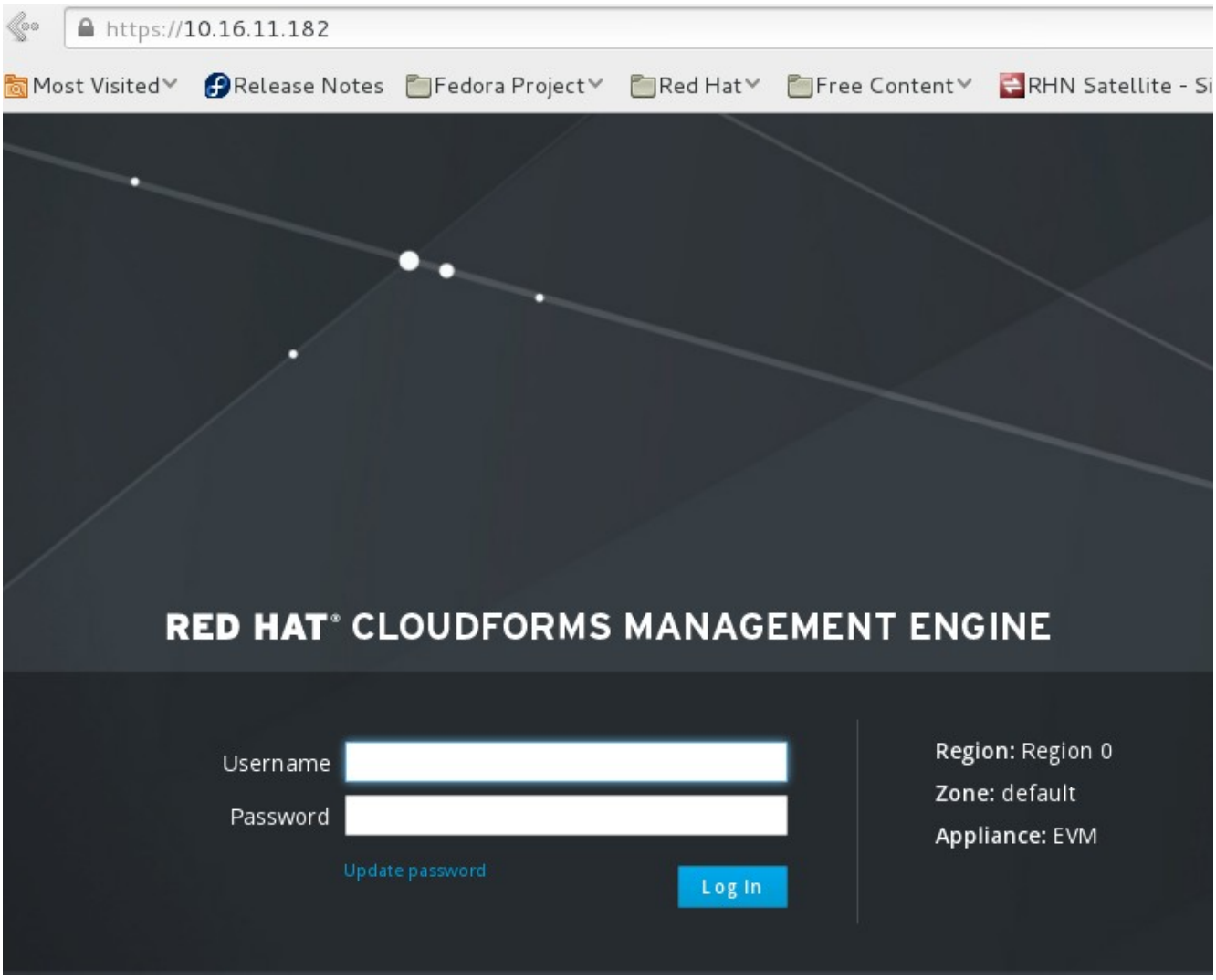


Figure 4.4.3-13: CFME Appliance Login

4.4.4 Adding a Cloud Provider

After CFME appliance deployment within the RHEL OSP environment, add the RHEL OSP cloud provider.

Login to the CFME appliance using the default credentials for the *admin* user account. Navigate to **Clouds**, **Providers**. Click the **Configuration** button and choose **Add a New Cloud Provider**.

Figure 4.4.4-1: Add Cloud Provider

Enter a **Name**, select the **Type**, provide a **Hostname** and **IP Address** in the **Basic Information** input box. The **API Port** defaults to 5000 and should not be changed.

In the **Credentials Input** box use the *admin* account for the RHEL OSP environment on the **Default** tab. Settings under the **AMQP** tab do not require input unless they are changed from the defaults during the RHEL OSP install. If default settings are changed refer to the settings used for the RHEL OSP install. Click the **Validate** button to verify correct settings. Click the **Add** button to complete process for adding a new provider.

> **Note:** qpid credentials can be found in */etc/nova/nova.conf* on the RHEL OSP controller node.

For the reference environment the following settings are used:

Input Box	Option	Value
Basic Information	Name	RHEL OSP 4
	Type	OpenStack
	Hostname	cf-rhos.refarch.bos.redhat.com
	IP Address	10.16.11.14
	API Port	5000
	Zone	Default
Credentials	User ID	admin
	Password	**[REDACTED]**
	Verify Password	**[REDACTED]**

Table 4.4.4-1: Cloud Provider Settings

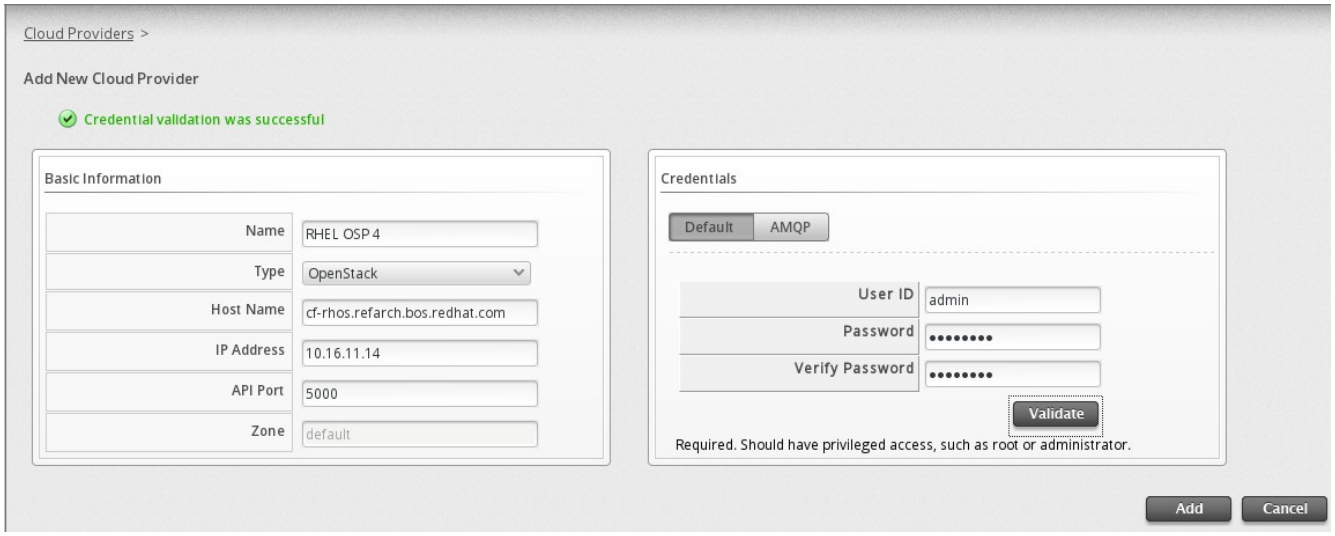


Figure 4.4.4-2: Cloud Provider Settings

Refresh Relationships and Power States for the newly added provider. Navigate to **Clouds**, **Providers** and place a check mark next to the provider. Click the **Configuration** button and choose **Refresh Relationships and Power States**. Click **OK** on the pop up window to proceed.

Figure 4.4.4-3: Provider Refresh

After a few moments instances running on the provider are displayed under **Clouds**, **Instances**.

4.4.5 CloudForms Management Engine Customization

For the reference environment, several items are configured on the CFME appliances to provide enhanced functionality and integration. Items include configuring:

- Regions and Zones
- Role Resiliency
- Lightweight Directory Access Protocol (LDAP) authentication
- Tags
- Capacity and Utilization Settings

4.4.5.1 Regions and Zones

CloudForms 3.0 provides the capability to partition managed environments into **Regions** and **Zones**. Regions can represent a geographic location where zones may represent cities or separate environments within the region.

The top level region contains a centralized VMDB database for reporting. Subordinate regions can exist under the top level region which replicate to the top level region however they do not

replicate databases to each other.

Zones are used to isolate traffic within a region. For example, one zone may contain a RHEL OSP based provider and another zone contain a RHEV based management system all within the same region.

Additional information regarding regions and zones can be found in the *CloudForms 3.0 Management Engine 5.2 Settings And Operations*[12] guide.

For the reference environment a single region and the d*efault* zone are used.

4.4.5.2 Role Resiliency

Several possibilities exist for CFME resiliency to include clustering an external PostgreSQL instance, by distributing server roles across multiple CFME appliances, or a combination of both. Distributing server roles consist of setting primary, secondary, and tertiary role assignments. For the reference environment role failover[13] is the method used for resiliency.

- Primary - There can only be one primary per zone per role. When an appliance is started, the system looks to see if any role is set to primary. If that is the case, the role is activated on that appliance and de-activated from the secondary.
- Secondary - This is the default priority. There can be multiple secondary CFME Appliances for a role. When an appliance is started, if no primary is found in the zone, the first appliance to start takes the role.
- Tertiary - If all appliances with primary roles or secondary roles are down, one of the tertiary would be activated.

Prior to assigning roles, designate a CFME appliance as the master VMDB instance and configure each additional CFME appliance to point to the designated master.

Access the CFME Console on a subordinate appliance using the *admin* account, hover over the **Configure** tab and select **Configuration**.

On the right window pane click the **Database** tab. Within the **Database** input box next to **Type**, click the pull down menu and change the setting from *Internal Database on this EVM Appliance* to *External Database on Another EVM Appliance*. Enter the hostname or IP address for the designated master in the **Hostname** field. Click the **Validate** button to verify settings and click **Save** to complete.

12 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Settings_And_Operations/chap-Configuration.html#sect-Settings

13 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Settings_And_Operations/sect-Diagnostics.html

For the reference environment *cf-cfme-rhev* is set to the designated master with *cf-cfme-rhelosp* configured as a subordinate.

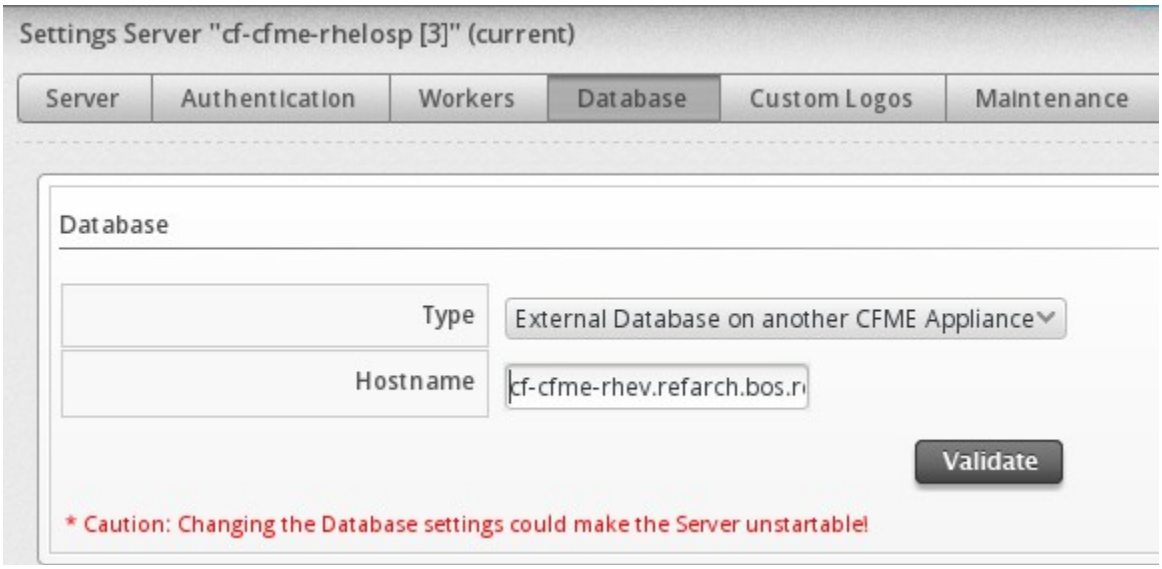


Figure 4.4.5.2-1: VMDB Settings

To assign server roles, access the CFME Console using the *admin* account, hover over the **Configure** tab and select **Configuration**.

On the right window pane under the **Server** tab place a check mark next to the desired roles within the **Server Control** input box.

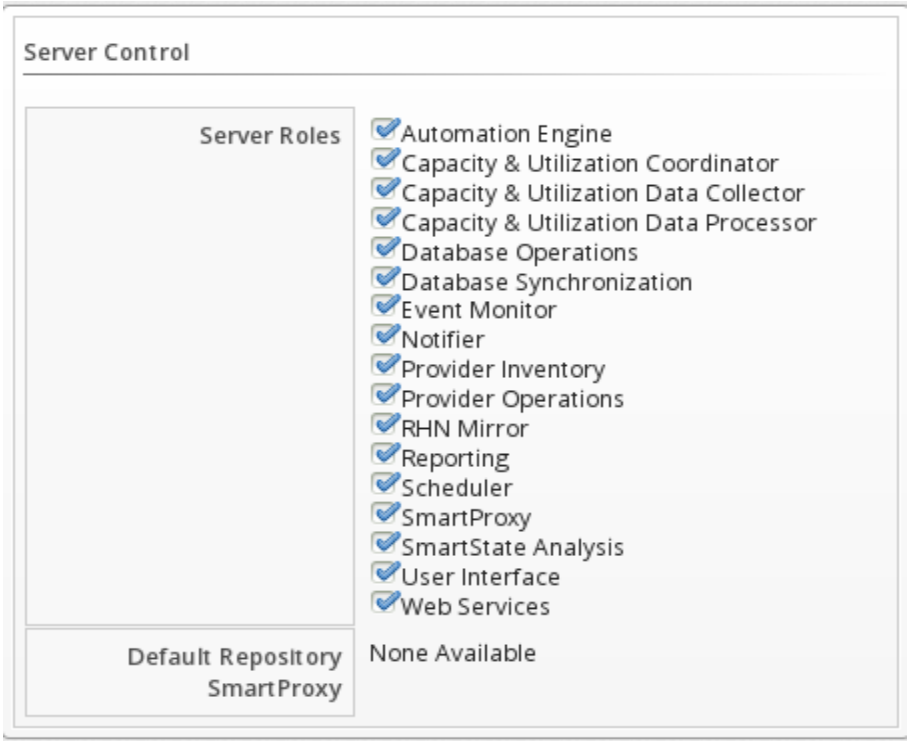


Figure 4.4.5.2-2: Role Assignments

Note: Not all roles are enabled by default and there may be some circumstances that require only certain roles be enabled on a CFME appliance. The requirements may include scaling and performance for the CFME appliance based on environment needs. Roles enabled by default are:

- Database Operations
- Event Monitor
- Reporting
- Scheduler
- SmartState Analysis
- User Interface
- Provider Inventory
- Provider System Operations
- Web Services

The following server roles support failover:

Roles	Type
Notifier	Primary, Secondary, Tertiary
Capacity and Utilization Coordinator	Primary, Secondary, Tertiary
Database Synchronization	Primary, Secondary, Tertiary
Scheduler	Primary, Secondary, Tertiary
Provider Inventory	Primary, Secondary, Tertiary

Table 4.4.5.2-1: Failover Roles

Additional roles exist on each CFME appliance which do not support failover. These roles work in conjunction with additional CFME appliances located in the same zone to support increased capacity.

Roles	Type
Automation Engine	Distributed
Capacity and Utilization Data Collector	Distributed
Capacity and Utilization Data Processor	Distributed
Database Operations	Distributed
Provider Operations	Distributed
Event Monitor	Distributed
Reporting	Distributed
SmartProxy	Distributed
SmartState Analysis	Distributed
User Interface	Distributed
Web Services	Distributed

Table 4.4.5.2-2: Distributed Roles

To define failover role priorities, access the CFME Console using the *admin* account, hover over the **Configure** and select **Configuration**.

On the left window pane accordion menu select **Diagnostics**. Click the zone where the CFME appliance(s) reside.

Under **Roles by Servers** a status display provides a listing of each role and priority assigned to each CFME appliance.

Select a role, click **Configuration** and either select promote or demote for the role. Primary role assignment is displayed in bold.

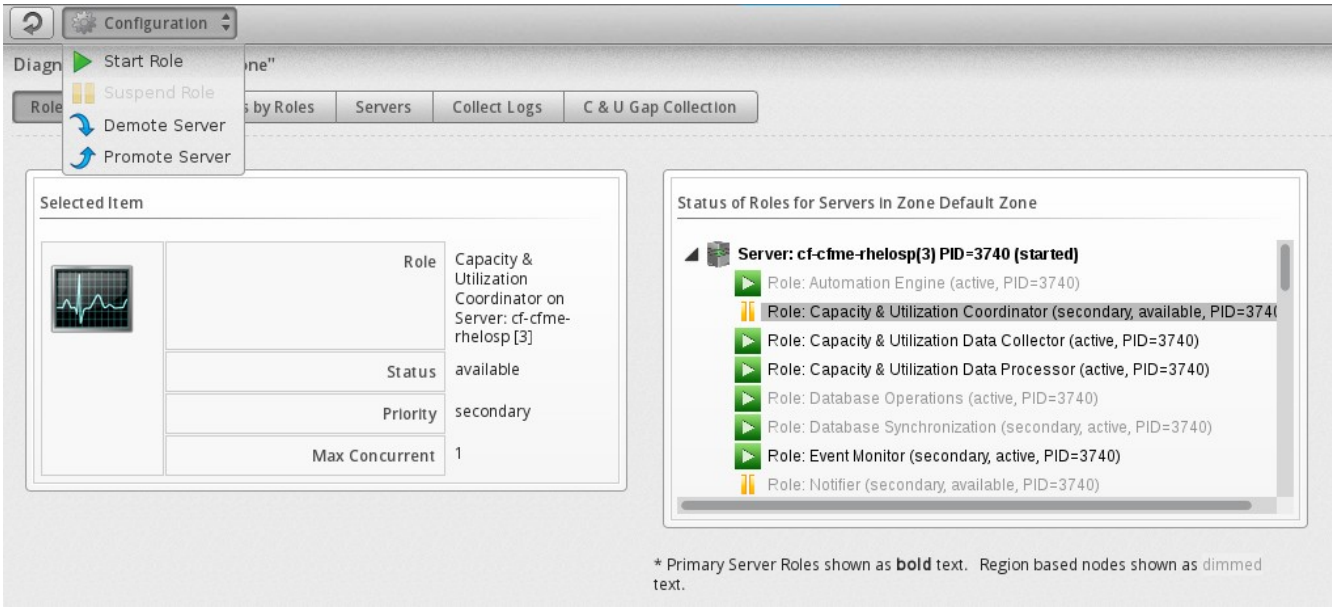


Figure 4.4.5.2-3: Role Priorities

Note: Roles that are grayed out are managed at the **Region** level.

For the reference environment the following roles are assigned for each CFME appliance to closely balance role responsibility:

Role	Primary	Secondary
*Notifier	cf-cfme-rhev	cf-cfme-rhelosp
Capacity and Utilization Coordinator	cf-cfme-rhelosp	cf-cfme-rhev
*Database Synchronization	cf-cfme-rhev	cf-cfme-rhelosp
*Scheduler	cf-cfme-rhev	cf-cfme-rhelosp
Provider Inventory	cf-cfme-rhelosp	cf-cfme-rhev
Event Monitor	cf-cfme-rhelosp	cf-cfme-rhev
°Virtual Environment to Storage Bridge	N/A	N/A
°Storage Inventory	N/A	N/A
°Storage Capacity and Utilization Coordinator	N/A	N/A
RHN Mirror[14]	cf-cfme-rhev	N/A
Automation Engine	cf-cfme-rhev, cf-cfme-rhelosp	
Capacity and Utilization Data Collector		
Capacity and Utilization Data Processor		
Database Operations		
Provider Operations		
Reporting		
SmartProxy		
SmartState Analysis		
User Interface		
Web Services		

Table 4.4.5.2-3: Assigned Roles

*Regional roles.

°Roles available for use with NetApp storage configurations.

14 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Installation_Guide/chap-Registering_and_Updating_CloudForms_Management_Engine.html

4.4.5.3 Secure LDAP Authentication

As discussed in **Section 4.2 Microsoft Active Directory**, Microsoft Active Directory is used to provide user authentication and group mapping for the reference environment.

To enable secure LDAP authentication, login to the CFME Console with the *admin* account, hover over **Configure** and select **Configuration**.

Select the **Authentication** tab at the top. On the bottom window pane there is a single box labeled **Authentication** with a **Mode** set to *Database*. This is the default authentication level.

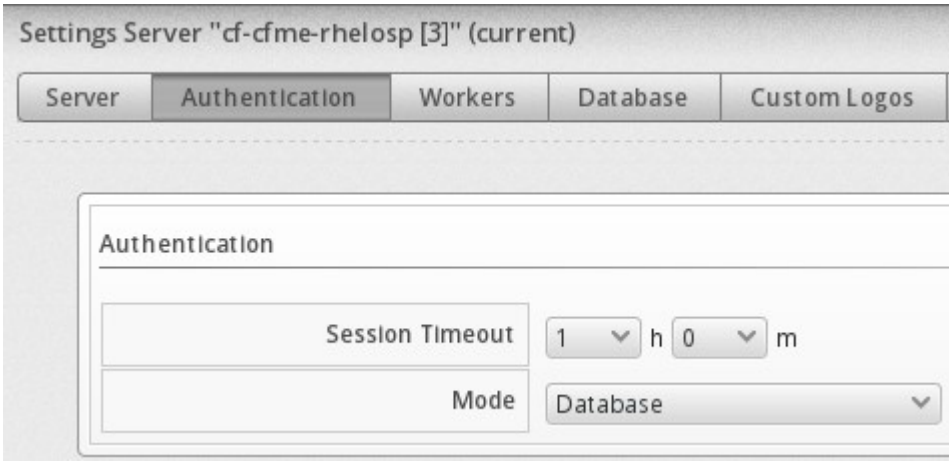


Figure 4.4.5.3-1: Authentication Mode – Database

To configure secure LDAP, set the **Mode** to *LDAPS*.

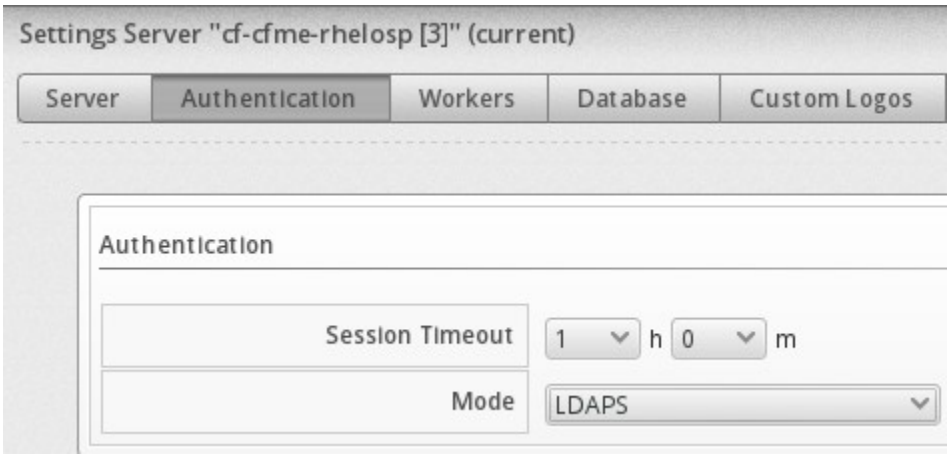


Figure 4.4.5.3-2: Authentication Mode – LDAPS

Once the **Mode** is set to *LDAPS*, additional input boxes are made available to provide details for the LDAP environment. For the reference environment the following inputs are provided:

Settings	Field	Value
LDAP	LDAP Host Names	cf-win-ad.refarch.bos.redhat.com
	LDAP Port	636
	User Type	User Principal Name
	User Suffix: <user@>	refarch.bos.redhat.com
Role	Get User Groups from LDAP	<checked>
	Get Roles from Home Forrest	<checked>
	Base DN	DC=refarch,DC=bos,DC=redhat,DC=com
	Bind DN	administrator@refarch.bos.redhat.com
	Bind Password	**[REDACTED]**

Table 4.4.5.3-1: LDAP Settings

With the desired values provided, click the **Validate** button to verify settings.

Figure 4.4.5.3-3: Authentication – Validate LDAPS Settings

Upon validation a success message is displayed at the top of the window pane.

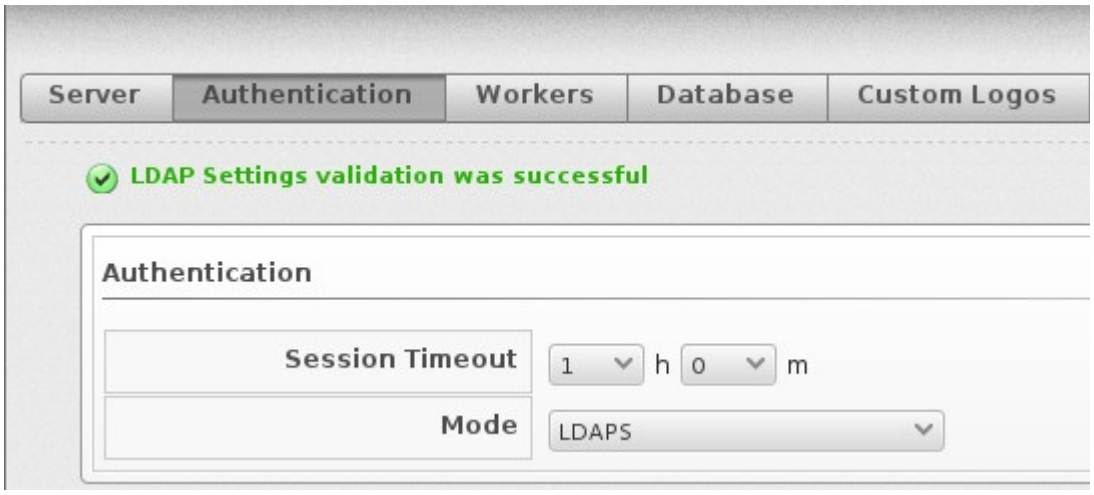


Figure 4.4.5.3-4: Authentication – Validate Success

Click **Save** at the bottom to complete the configuration.

4.4.5.4 Tags

Using tags within CloudForms 3.0 provides the capability to organize and manage resources from users, groups, and roles to management systems, hosts, virtual machines and more.

Tags are either system defined or custom created. For the reference environment a single custom tag is created and assigned to managed resources.

To create a custom tag, login to the CFME Console with the *admin* account, hover over the **Configure** tab and select **Configuration**. On the left window pane under **Settings**, select the **Region**.

On the right window pane click the **Red Hat Tags** tab. Within the **Choose a Category** input box, click the drop down menu next to the **Category** field to choose a category to add a custom tag. For the reference environment, *Environment* is the category selected.

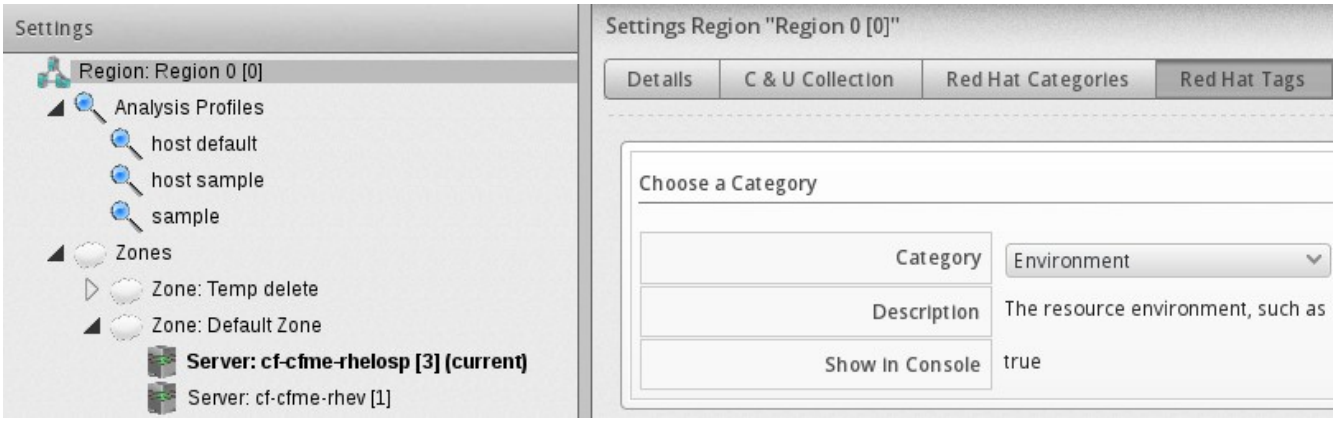


Figure 4.4.5.4-1: Custom Tag

In the **Environment Entries** input box, click the green plus and provide a **Name** and **Display Name**. To complete click the server icon to the left of the input fields to add.

Figure 4.4.5.4-2: Add Custom Tag

Upon completion the new tag is listed. For the reference environment the new tag created is *Environment > cloudforms*.

Figure 4.4.5.4-3: Custom Tag Complete

Note: The *Name* field must be a single word, lower case.

4.4.5.5 Self-Service Users

CloudForms self-service users are mapped to LDAP users. For the reference environment the following steps are completed to map LDAP users to CloudForms Roles and Groups.

- Create a custom role and assign features
- Create a new group and assign role(s) and users

Create Custom Role

To create a custom role, login to the CFME Console with the *admin* account, hover over the **Configure** tab and select **Configuration**. On the left window pane select **Access Control**. Scroll down and highlight **Roles**. On the right window pane click the **Configuration** button and choose **Add a new Role**.

In the **Role Information** input box provide a **Name** and choose an option for **VM & Template Access Restriction**. For the reference environment the following variables are used:

Field	Setting
Name	CloudForms–user_self_service
VM & Template Access Restriction	None

Table 4.4.5.5-1: Role Creation

In the **Product Features** input box the following features are enabled for the custom role:

Category	Feature Selection
Everything/Services/My Services/All Services	View All Services
Everything/Services/Catalogs Explorer/Catalog Items	View Catalog Items
Everything/Services/Catalogs Explorer/Catalogs	View Catalogs
Everything/Services/Catalogs Explorer	Service Catalogs (all items underneath)
Everything/Services/Requests	View (all items underneath)
Everything/Services	Workloads (all items underneath)
Everything/Clouds/Cloud Providers	View
Everything/Clouds/Availability Zones	
Everything/Clouds/Flavors	
Everything/Clouds/Security Groups	
Everything/Clouds/Instances/Accordions	Images, Images by Provider, Instances, Instances by Provider
Everything/Settings and Operations/My Settings/Modify	Default Views, Visual
Everything/Settings and Operations/My Settings	About

Table 4.4.5.5-2: Role Features

Click the **Add** button to complete.

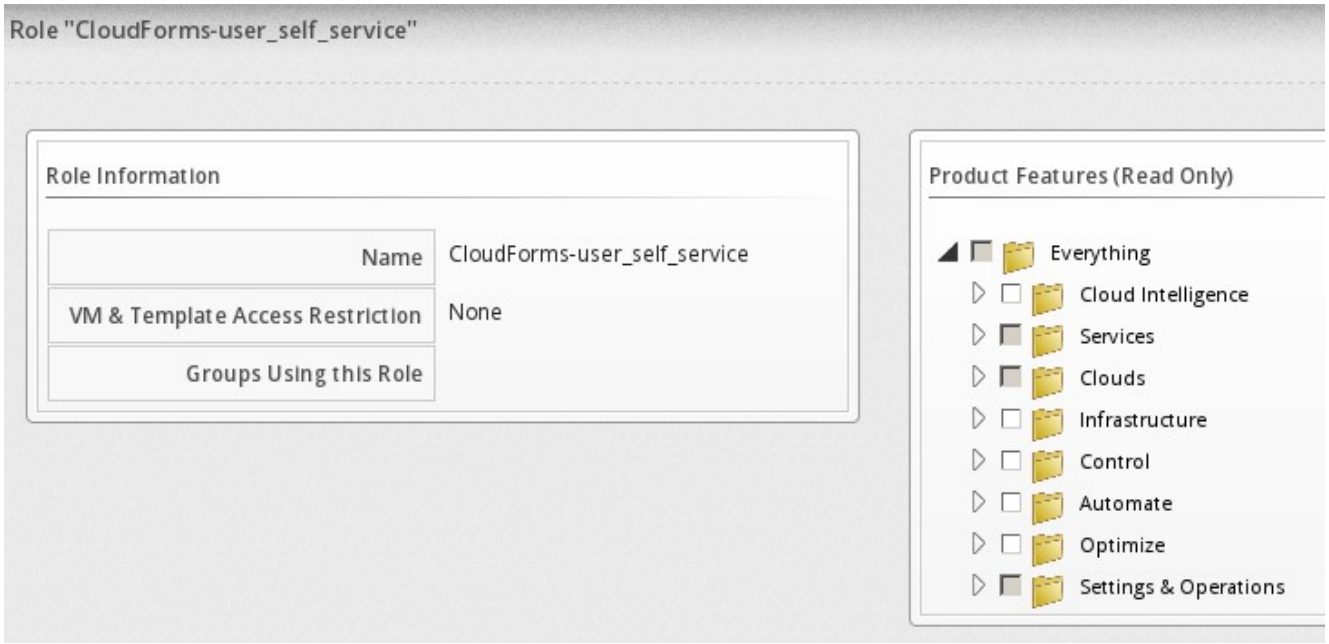


Figure 4.4.5.5-1: Custom Role Creation

Create Group

On the left window pane under **Access Control**, select **Groups**. On the right window pane click the **Configuration** button and choose **Add a new Group**.

Under the Group Information input box provide a **Name** and place a check mark next to **Look Up LDAP Groups**. A new input box named **LDAP Group Look Up** will appear below the **Group Information** input box. Input the necessary values for **User to Look Up**, **User id**, and **Password**. Click the **Retrieve** button to complete.

Upon success a new input choice is made available in the **Group Information** input box called **LDAP Groups for User**. Next to this option select the available LDAP group the LDAP user is a member of. Provide a **Description** and assign a **Role** to the new group.

For the reference environment the following inputs are provided:

Input	Option	Value
Group Information	LDAP Groups for User	dev
	Description	dev
	Role	CloudForms-user_self_service
LDAP Group Lookup	User to Look Up	cf-dev
	User Id	administrator
	Password	**[REDACTED]**

Table 4.4.5.5-3: Group Creation Inputs

Click the **Add** button to create the new group.

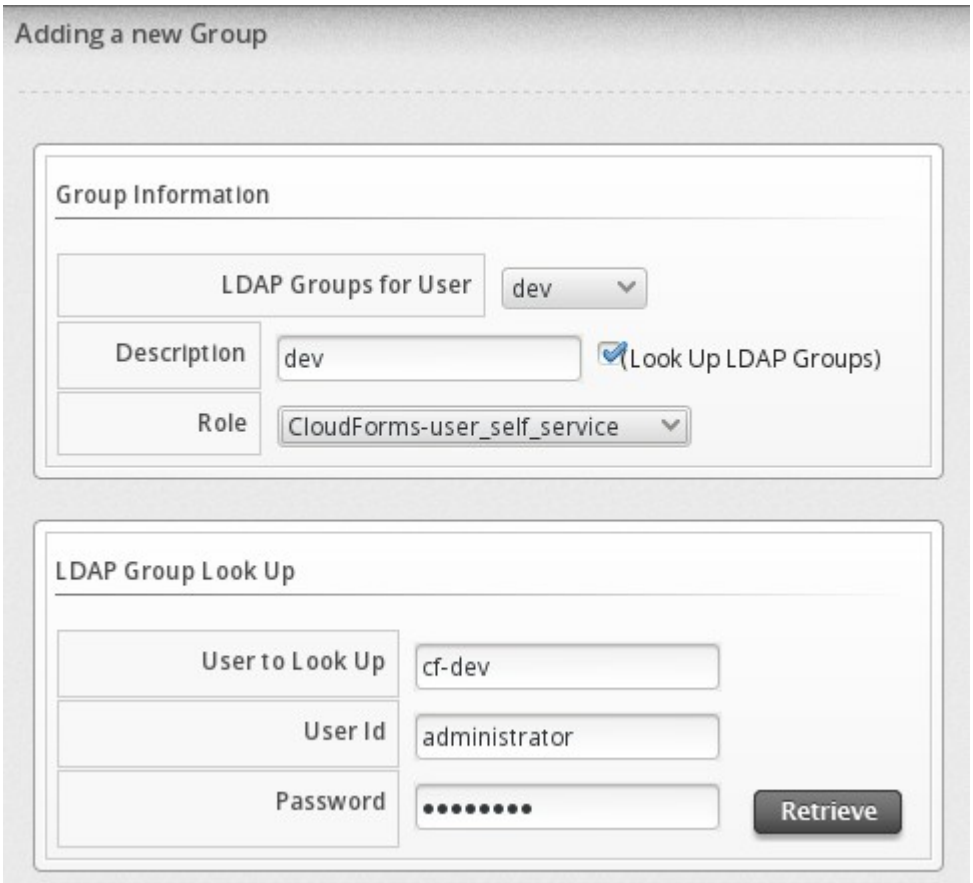


Figure 4.4.5.5-2: Group Creation

With the new group created, assign the previously created Red Hat Tag to the group. On the right window pane highlight the newly created group under **Groups**. On the right window pane click the **Policy** button and select **Edit 'Red Hat' Tags for this Group**.

In the **Tag Assignment** input box, next to **Select a customer tag to assign**, choose the desired value and tag from the pull down options. For the reference environment, *Environment/CloudForms* is the chosen tag assigned to the *dev* group. Click **Save** to complete.

Figure 4.4.5.5-3: Tag Assignment

To complete the self-service user creation, login to the CFME Console with the self-service user account. This maps the LDAP user to the assigned CFME group and assigned role. Verify accessibility to the assigned *Features* within the chosen role.

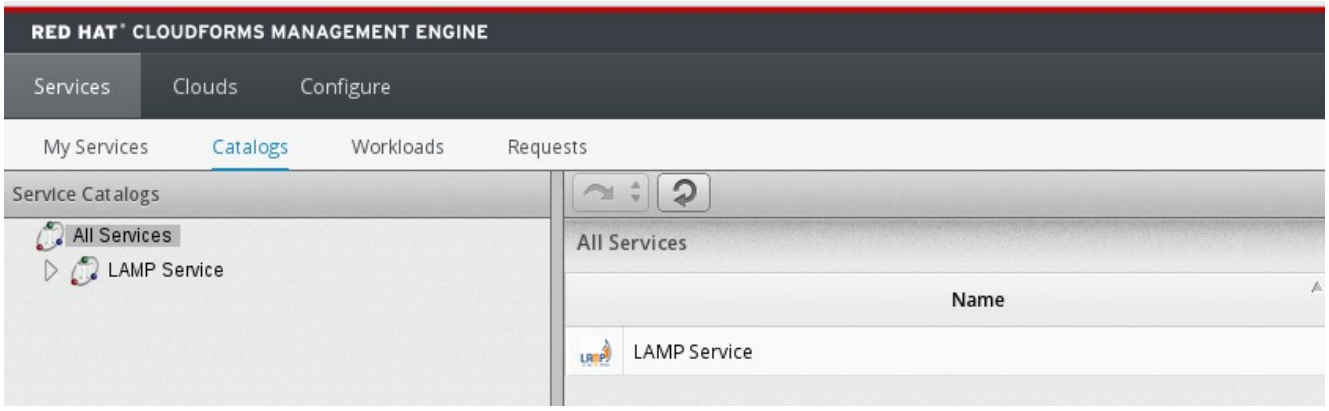


Figure 4.4.5.5-4: Self-Service User Login

4.4.5.6 Capacity and Utilization Settings

Enabling Capacity and Utilization collection within CloudForms for a managed RHEL OSP provider involves the following:

- Enable Capacity and Utilization server roles within CloudForms Management Engine
- Enable Capacity and Utilization collection for all Clusters

Capacity and Utilization Roles

To enable Capacity and Utilization server roles within CFME, login as the *admin* user and navigate to **Configure**, **Configuration**. Under **Settings** in the left window pane, with the desired server selected under the assigned Zone, on the right window pane select the **Server** tab.

In the **Server Control** input box place a check mark next to *Capacity & Utilization Coordinator*, *Capacity & Utilization Data Collector*, and *Capacity & Utilization Data Processor*. Click **Save** in the bottom right to persist the changes.

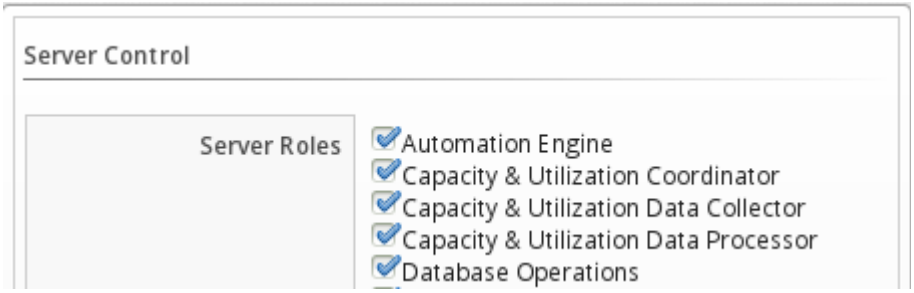


Figure 4.4.5.6-1: Capacity and Utilization Server Roles

Capacity and Utilization Collection

To enable Capacity and Utilization collection for clusters, as *admin*, navigate to **Configure**, **Configuration**. On the left window pane under **Settings**, click on the top most Region. On the right window pane select the **C & U Collection** tab. In the **Clusters** input box place a check mark next to **Collect for All Clusters**. Click the **Save** button in the bottom right to make changes persistent.

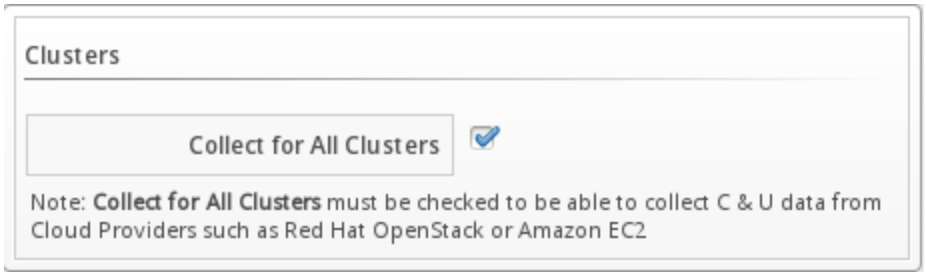


Figure 4.4.5.6-2: Capacity and Utilization Collect for All Clusters

Ensure the Capacity and Utilization workers are running by navigating to **Configure**, **Configuration**. On the left window pane select **Diagnostics**. On the right window pane select the **Workers** tab and look for two *C & U Metrics Collector for Openstack* workers. Status should be *started*.

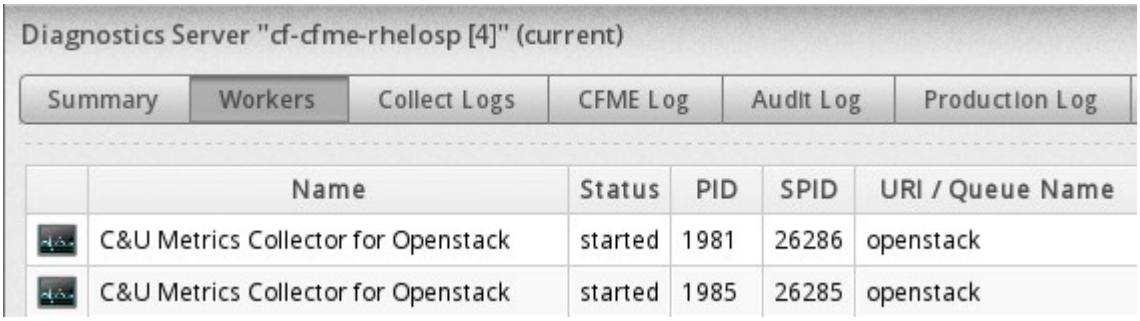


Figure 4.4.5.7-1: Capacity and Utilization Worker Status

4.4.5.7 Customization Templates

A single customization template[15] is utilized in the reference environment. This customization template configured within CloudForms utilizes the cloud-init function within the RHEL OSP environment for the LAMP image used to prevent duplicate hostnames and configure the LAMP services to use the assigned, unique hostname and ip address.

To create a customization template, as *admin*, navigate to **Infrastructure**, **PXE**. On the left window pane click **Customization Templates** and select **RHEL 6** under **All Customization Templates – System Image Types**.

On the right window plane click the **Configuration** button and select **Add a New Customization Template**. In the **Basic Information** input box provide a **Name**, **Description**,

15 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Lifecycle_and_Automation_Guide/chap-Provisioning_Requests.html#sect-Customization_Templates_for_Virtual_Machine_and_Instance_Provisioning

Image Type, **Type**, and **Script** contents. For the reference environment the following inputs are provided:

Input Box	Option	Value
Basic Information	Name	LAMP
	Description	LAMP Service
	Image Type	RHEL-6
	Type	Cloudinit
	Script	See **Appendix E CloudForms Custom Methods and Scripts** *(cloudinit.sh)*

Table 4.4.5.7-1: Customization Template Settings

Customization template details displayed below.

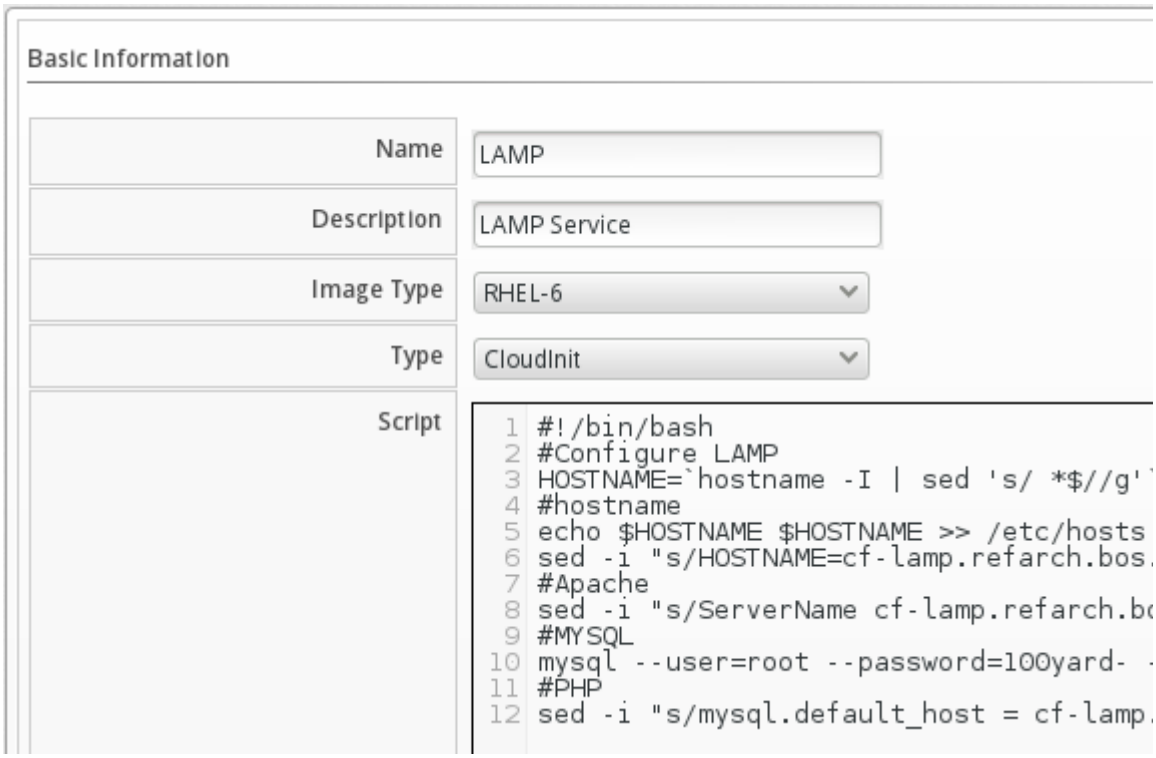


Figure 4.4.5.7-2: Customization Template Creation

Click the **Add** button to complete the customization template creation.

4.5 Creating a RHEL OSP LAMP Image

For the reference environment a single LAMP image is created and imported into the RHEL OSP environment. Steps include:

- Installing and configuring LAMP within a virtual machine
- Preparing and exporting a qcow2 image
- Importing the LAMP image into RHEL OSP

4.5.1 Installing LAMP

The process for installing and configuring LAMP, using DVD Store[16] to exercise the stack, is found in the following previously published Red Hat reference architecture: *Deploying the LAMP Stack on Red Hat Enterprise Linux 5*[17].

The following items differ from the installation steps outlined in the mentioned reference architecture:

- Operating System – Red Hat Enterprise Linux 6.5
- Latest supported version of PHP, Apache, and MySQL
- `setsebool -P httpd_can_network_connect_db=1` – resolves error: php Can't connect to MySQL server
- As user *web*, set up DVD Store 2 (DS2) database:
  - Copy tarballs (ds2.tar.gz and ds2_mysql.tar.gz) from http://linux.dell.com/dvdstore to */home/web/* and extract with tar -xzf – This creates */home/web/ds2* and subdirectories
  - Either leave contents of */home/web/data_files* alone (for small [10 MB] database) or copy over medium (1 GB) or large (100 GB) files
  - To build and load database, as *web*: sh mysqlds2_create_all.sh

4.5.2 Preparing and Exporting Image

With the LAMP stack and DVD Store installation and configuration complete, the next step is to prepare and export the virtual machine into a qcow2 image. For the reference environment the virtual machine is hosted within a Red Hat Enterprise Virtualization 3.3 environment.

Prepare the virtual machine by performing the following steps:

- Install the **cloud-init** package (used to support user-data[18] configuration at instance boot)

```
# yum install -y cloud-init
```

- Disable iptables

```
# chkconfig iptables off
```

16 http://en.community.dell.com/techcenter/extras/w/wiki/dvd-store.aspx
17 https://access.redhat.com/site/articles/215423
18 http://docs.openstack.org/user-guide/content/user-data.html

- Remove existing network interface configuration

```
# rm -rf /etc/udev/rules.d/70-persistence-net.rules
```

- Remove the HWADDR entry from */etc/sysconfig/network-scripts/ifcfg-eth0*
- Verify *openssh* is installed and set to start on boot

```
# rpm -qa openssh
openssh-5.3p1-94.el6.x86_64
# chkconfig sshd on
```

- Add the following to */etc/sysconfig/network*: NOZEROCONF=true
- Fill the remaining free space within the virtual machine disk with 0's and remove the file when complete

```
# dd if=/dev/zero of=zerofile bs=1M
# rm -rf zerofile
```

Note: The **cloud-init** package is available in the *rhel-x86_64-server-rh-common-6* channel.

With the virtual machine still powered on, **ssh** to the RHEV-H machine where the virtual machine is running.

```
# ssh root@cf-rhevh
[root@cf-rhevh ~]#
```

Capture the disk image location of the running virtual machine.

```
# virsh --readonly -c qemu:///system domblklist cf-lamp
Target Source
hdc -
vda
/rhev/data-center/913278b3-acce-4b3a-b9ee-f34a44b9e740/620a3241-6bfa-477b-a7
be-405bfa73be1f/images/4c090ddd-30b9-4f3d-a185-a60897218a43/63e46eb0-a767-43
70-b982-f91fe63a2e28
```

Find the symbolic link associated to the virtual machine disk.

```
# ls -l
/rhev/data-center/913278b3-acce-4b3a-b9ee-f34a44b9e740/620a3241-6bfa-477b-a7
be-405bfa73be1f/images/4c090ddd-30b9-4f3d-a185-a60897218a43/63e46eb0-a767-43
70-b982-f91fe63a2e28

lrwxrwxrwx. 1 vdsm kvm 78 2014-01-03 03:54
/rhev/data-center/913278b3-acce-4b3a-b9ee-f34a44b9e740/620a3241-6bfa-477b-a7
be-405bfa73be1f/images/4c090ddd-30b9-4f3d-a185-a60897218a43/63e46eb0-a767-43
70-b982-f91fe63a2e28 ->
/dev/620a3241-6bfa-477b-a7be-405bfa73be1f/63e46eb0-a767-4370-b982-f91fe63a2e
28
```

Power the virtual machine off and activate the logical volume associated with the virtual machine.

```
# lvchange -ay
/dev/620a3241-6bfa-477b-a7be-405bfa73be1f/63e46eb0-a767-4370-b982-f91fe63a2e
28
```

Export the image to a share. For the reference environment a central NFS share is used.

```
# qemu-img convert -p -O qcow2
/dev/620a3241-6bfa-477b-a7be-405bfa73be1f/63e46eb0-a767-4370-b982-f91fe63a2e
28 /pub/projects/cf-2.1/cf-lamp.qcow2
```

Upon completion the newly created qcow2 image is ready to import into the RHEL OSP environment.

4.5.3 Importing the LAMP image into RHEL OSP

To import the newly created LAMP qcow2 image into RHEL OSP, login to the RHEL OSP dashboard (http://ip_address/dashboard) with the *admin* account. Under the **Admin** tab select **Images**. Click the **Create Image** button in the upper right.

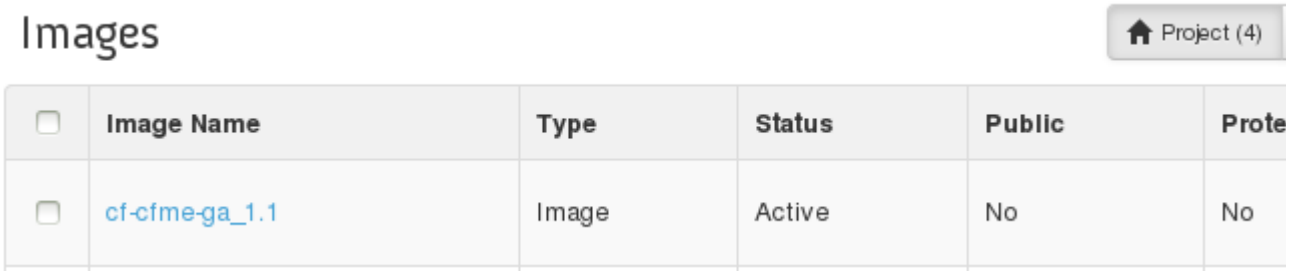


Figure 4.5.3-1: Create Image

Provide a **Name**, **Image Source**, and **Format**. For the reference environment a check mark is placed next to **Public** to allow the image to be used by all tenants and users.

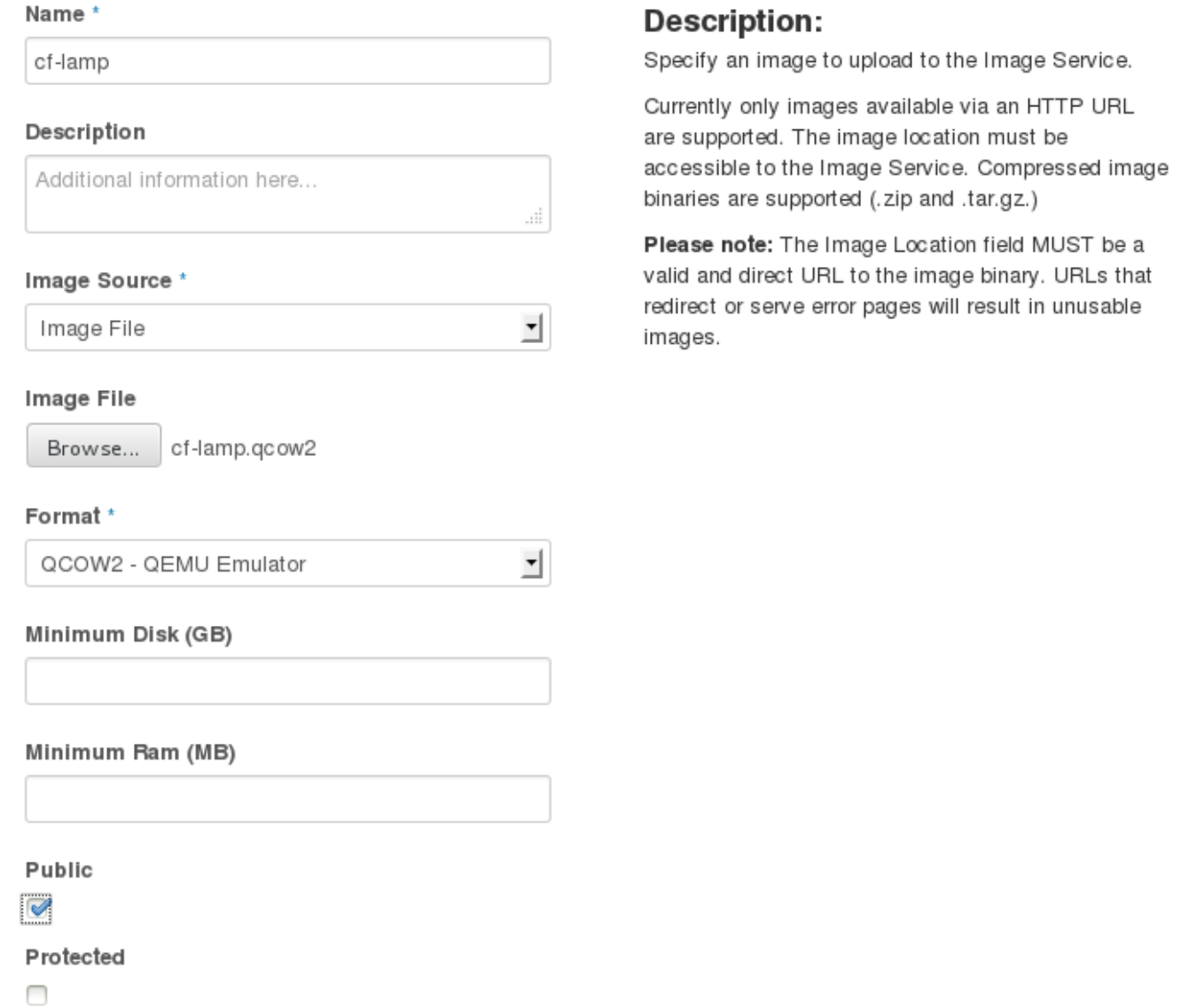


Figure 4.5.3-2: Image Details

Click **Create Image** to initiate the image import.

> **Note: Minimum Disk** and **Minimum RAM** are left blank meaning instances created using the image are not limited by the *Flavor*[19] chosen at time of creation.

19 http://docs.openstack.org/trunk/openstack-ops/content/flavors.html

Upon completion the image is listed under **Images** and is ready for use.

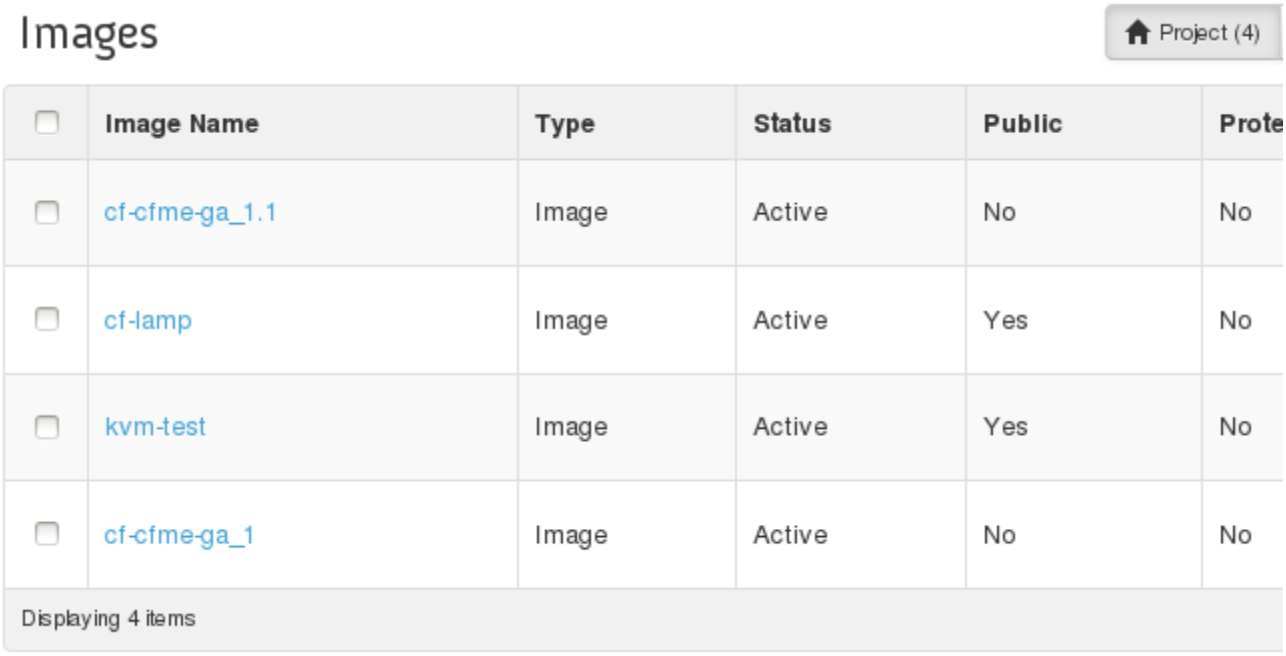


Figure 4.5.3-3: Image List

4.6 Security

RHEL/RHEV/RHEL OSP

`selinux` is enabled and set to enforcing on all systems where applicable. **`iptables`** is enabled on all infrastructure systems and necessary ports open where applicable. Refer to **Appendix C iptables**.

Microsoft Windows Server

Microsoft Windows firewall is enabled for *Domain*, *Private*, and *Public* profiles. Standard rules are used.

5 Service Catalog Creation and Self-Service Users

CloudForms features the concept of Service Catalogs to deploy a service or set of services. Such services may include a web server, a database, or an application for example. These services may run independently or packaged as a bundle for deployment.

For the purposes of the reference environment, a single LAMP service is utilized for development activities made available to the *cf-dev* self-service user.

Details for creating a Service Catalog and associated dependencies are found in the *CloudForms 3.0: Management Engine 5.2 Lifecycle and Automation Guide*[20].

For the reference environment the following steps are taken:

- Automate customization
- Create a service dialog
- Create a service catalog
- Ordering an Application as a Self Service User

5.1 Automate Customization

Creating a service catalog involves associating the service creation with a custom automate method. To create a custom automate method, as *admin*, navigate to **Automate**, **Explorer**.

In the left window pane navigate to **Datastore/Factory/Service**. On the right window pane verify the **Instances** tab is highlighted, click **Configuration** and select **Add a New Instance**.

In the **Main Info** input box provide a **Name**, **Display Name**, and **Description**. In the **Fields** input box provide a **Value**. The **Value** entry next to **(execute)** is the name of the method used. For the reference environment the following inputs are provided:

Input Box	Option	Value
Main Info	Name	lampsrv
	Display Name	lampsrv
	Description	LAMP Server
Fields	Value	lampsrv

Table 5.1-1: Automate Instance Settings

20 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Lifecycle_and_Automation_Guide/chap-Catalogs_and_Services.html

lampsrv instance details displayed below.

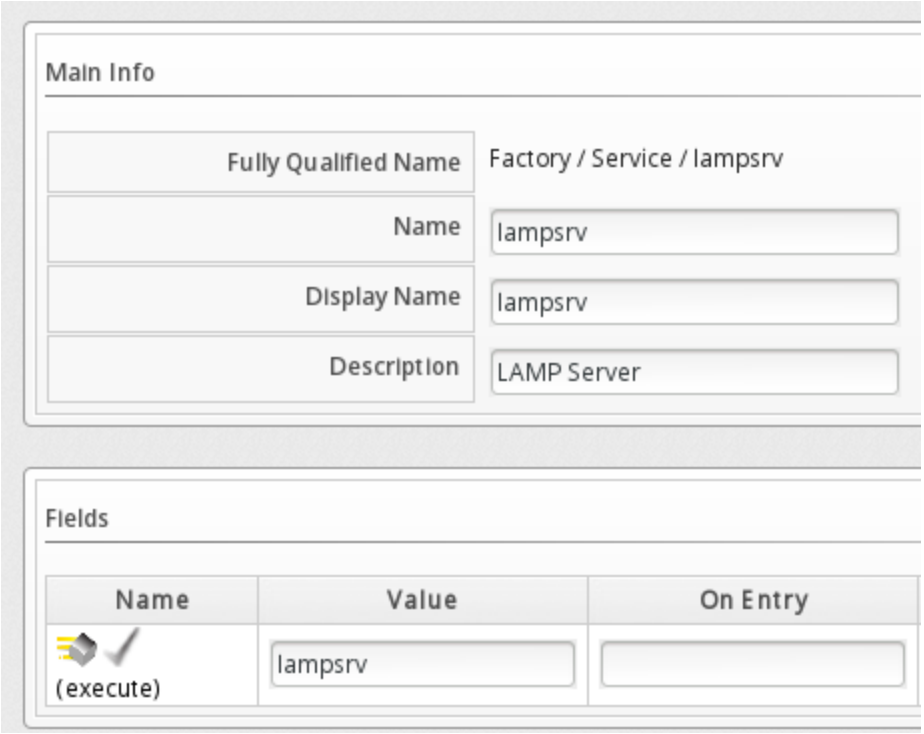


Figure 5.1-1: lampsrv Automate Instance

Click **Add** to complete the instance creation.

Next create a new method by selecting the **Methods** tab in the right window pane. Click **Configuration** and select **Add a New Method**.

In the **Main Info** input box provide a **Name**, **Display Name**, and **Data**. The **Location** is left at the default entry of *inline*.

For the reference environment the following inputs are provided:

Input Box	Option	Value
Main Info	Name	lampsrv
	Display Name	lampsrv
	Data	**Appendix E CloudForms Custom Methods and Scripts**

Table 5.1-2: Automate Instance Settings

Click the **Validate** button to verify the code used for the method in the **Data** code box. If there are formatting issues with the code an error is displayed. If the code checks out, a success message appears.

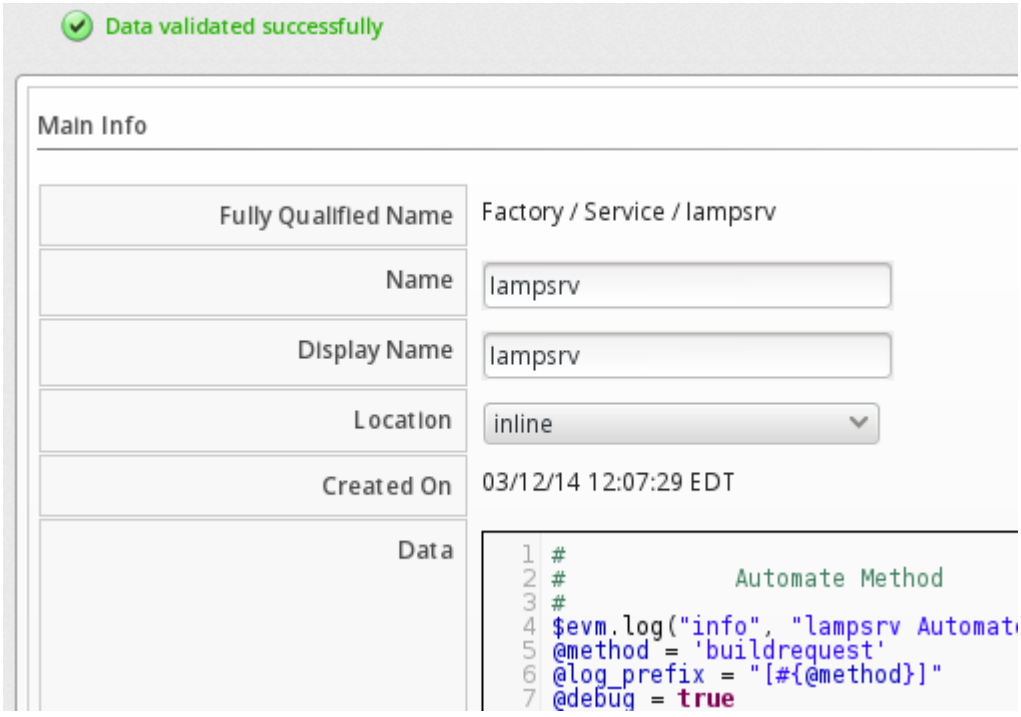


Figure 5.1-2: lampsrv Automate Method

Click **Add** to complete the method creation.

For the reference environment the *PreProvision* method is modified to allow instance creation using the custom *lampsrv* automate method. To modify the *PreProvision* method, in the left window pane, navigate to **Datastore/Factory/Virtual Machine (VM)** and select the *PreProvision* method. In the right window pane click **Configuration** and select **Edit this Method**. The **bold** items are added to the existing method:

```
                        <content removed for brevity>

  ###################################
  #
  # Method: process_openstack
  # Notes: Process OpenStack specific provisioning options
  #
  ###################################
  def process_openstack(prov )
    cloud_network = prov.get_option(:ws_values)[:cloud_networks]
    cloud_network = cloud_network.to_i
    log(:info, "Cloud network [#{cloud_network.inspect}]")
    prov.set_network_adapter(0, {:network_id => cloud_network})
  end # end process_openstack


  # Get provisioning object
  prov = $evm.root["miq_provision"]
  log(:info, "Provision:<#{prov.id}>
Request:<#{prov.miq_provision_request.id}> Type:<#{prov.type}>")

  # Build case statement to determine which type of processing is required
  case prov.type
  when 'MiqProvisionRedhatViaIso', 'MiqProvisionRedhatViaPxe';
process_redhat(prov)
  when 'MiqProvisionVmware';
process_vmware(prov)
  when 'MiqProvisionAmazon';
process_amazon(prov)
  when 'MiqProvisionOpenstack';
process_openstack(prov)
  else log(:info, "Provision Type:<#{prov.type}> does not match, skipping
processing")
  end

                        <content removed for brevity>
```

To complete the automate customization, create an entry point into the automate model. On the left window pane navigate to **Datastore/System/Automation Requests (Requests)**.

On the right window pane verify the **Instances** tab is highlighted, click the **Configuration** button and select **Add a New Instance**.

In the **Main Info** input box provide a **Name**, **Display Name**, and **Description**. In the **Fields** input box provide a **Value** for **(rel1)**. The **Value** entry next to **(rel1)** is the path to the custom service method. For the reference environment the following inputs are provided:

Input Box	Option	Value
Main Info	Name	lampsrv
	Display Name	lampsrv
	Description	LAMP method entry
Fields	(rel1)	/Factory/Service/lampsrv

Table 5.1-3: Request Instance Settings

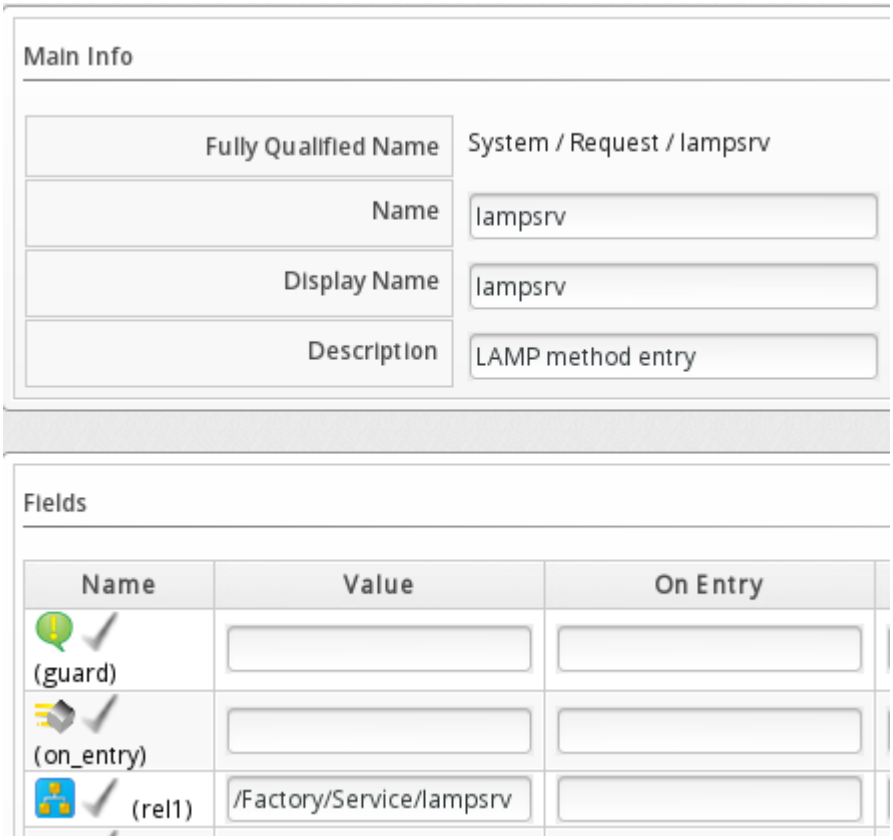


Figure 5.1-3: Service Entry Instance

Click **Add** to complete the service entry creation.

5.2 Create a Service Dialog

A custom service dialog associates the custom automate method to the service catalog.

To create a service dialog navigate to **Automate**, **Customization**. On the left window pane select **Service Dialogs**. On the right window pane click **Configuration** and select **Add a new Dialog**.

In the **Dialog Information** input box provide a **Label**, **Description**, and place a check mark next to *Submit* and/or *Cancel* for **Buttons**. For the reference environment the following inputs are provided:

Option	Value
Label	Description
Description	Instance Description
Buttons (checked)	Submit
	Cancel

Table 5.2-1: Dialog Information Settings

Click the green plus button and select **Add a New Tab to this Dialog**. In the **Tab Information** input box provide a **Label** and **Description**. For the reference environment the following inputs are provided:

Option	Value
Label	Service Description
Description	Service Description

Table 5.2-2: Tab Information Settings

Click the green plus button and select **Add a New Box to this Tab**. In the **Box Information** input box, provide a **Label** and **Description**. For the reference environment the following inputs are provided:

Option	Value
Label	Description
Description	Notes for Service

Table 5.2-3: Box Information Settings

Click the green plus button and select **Add a New Element to this Box**. In the **Element Information** input box provide a **Label**, **Name**, **Description**, and **Type**. For the reference environment the following inputs are provided:

Option	Value
Label	Service Description
Name	Description
Description	Notes for Service
Type	Text Area Box

Table 5.2-4: Element Information Settings

Upon selecting *Text Area Box* for **Type**, a new input box appears for **Options**. For the reference environment **Default Value** is left blank and a check mark is placed next to **Required**.

Figure 5.2-1: New Service Dialog

Click **Add** to complete the service dialog creation.

Create a custom button group by selecting **Buttons** on the left window pane and highlight **VM and Instance**. On the right window pane click **Configuration** and select **Add a new Button Group**.

In the **Action** input box provide a **Button Group Text**, **Button Group Hover Text**, and select a **Button Group Image** from the drop-down selection.

For the reference environment the following inputs are provided:

Option	Value
Button Group Text	LAMP
Button Group Hover Text	LAMP Service
Button Group Image	
Display on Button	checked

Table 5.2-5: Button Group Action Settings

Click **Add** to complete the button group creation.

In the left window pane select the newly created button group. On the right window pane click **Configuration** and select **Add a new Button**.

In the **Action** input box provide a **Button Text**, **Button Hover Text**, select a **Button Image**, and select a **Dialog** from the drop-down selections.

For the reference environment the following inputs are provided:

Option	Value
Button Text	LAMP Service
Button Hover Text	LAMP Service
Button Image	
Display on Button	checked
Dialog	Description

Table 5.2-6: Button Action Settings

In the **Object Details** input box select the **/System/Process/** path, **Message**, and **Request**. This is the entry point into the automate model that initiates the instance creation. For the reference environment the following inputs are provided:

Option	Value
/System/Process	Request
Message	create
Request	lampsrv

Table 5.2-7: Button Object Details Settings

Button details displayed below.

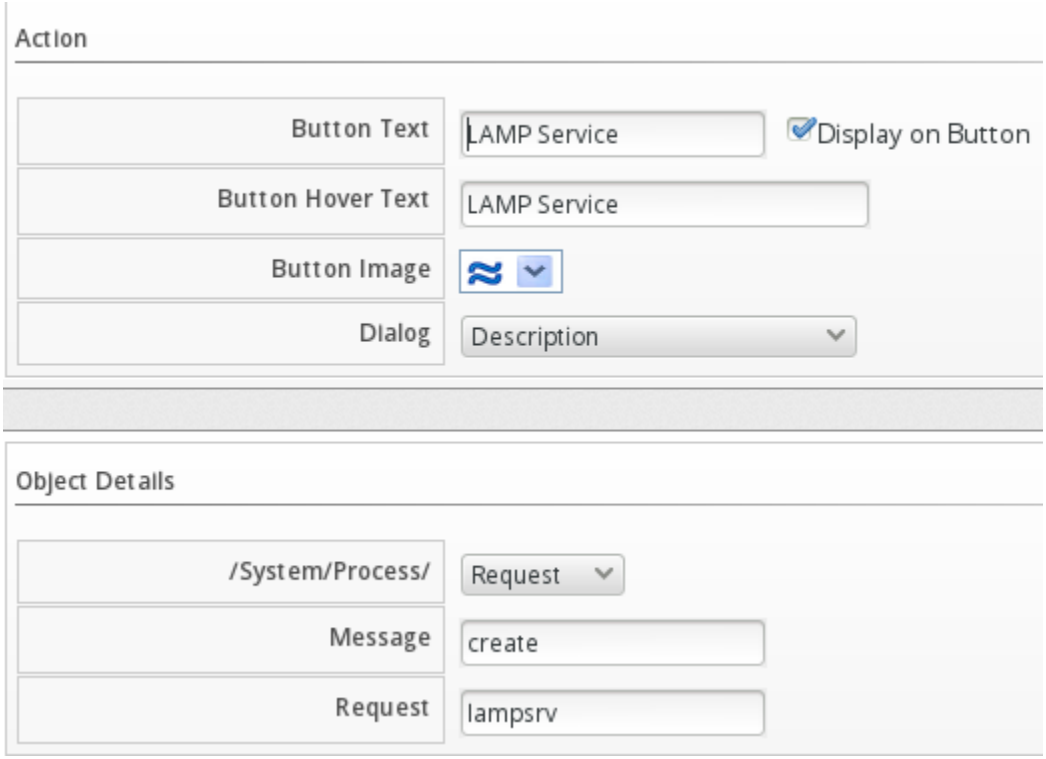


Figure 5.2-2: Button Details

Click **Add** to complete the button group creation.

Note: Default values are accepted for the remaining input boxes.

5.3 Create a Service Catalog

The next step involves creating the service catalog, linked to the automate method, and service dialog.

As *admin*, navigate to **Services**, **Catalogs**. On the left window pane select **Catalogs**. On the right window pane click **Configuration** and select **Add a New Catalog**. In the **Basic Info** input box provide a **Name** and **Description**. In the **Assigned Buttons** input box highlight the **UnAssigned** button created previously and click the arrow to move it to **Selected**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Basic Info	Name	LAMP Service
	Description	LAMP Service
Assign Buttons	Selected	LAMP Service

Table 5.3-1: Catalog Settings

Click **Add** to complete the catalog creation.

The next step is to create a catalog item. On the left window pane select **Catalog Items**. On the right window pane click **Configuration** and select **Add a New Catalog Item**.

In the **New Catalog Item** input box choose a **Catalog Item Type**. For the reference environment *OpenStack* is selected.

A new window is displayed. With the **Basic Info** tab selected, in the **Basic Info** input box provide a **Name / Description**. Place a check mark next to **Display in Catalog**. Addition input fields appear. Select **Catalog**, **Dialog**, and provide **Provision Entry Point** and **Retirement Entry Point** paths. For the reference environment the following inputs are provided:

Input Box	Option	Value
Basic Info	Name / Description	LAMP Service / LAMP Service
	Display in Catalog	checked
	Catalog	LAMP Service
	Dialog	Description
	Provision Entry Point	System/Request/lampsrv
	Retirement Entry Point	Factory/Service/retireservicevms

Table 5.3-2: Catalog Item Settings

Catalog Item details displayed below.

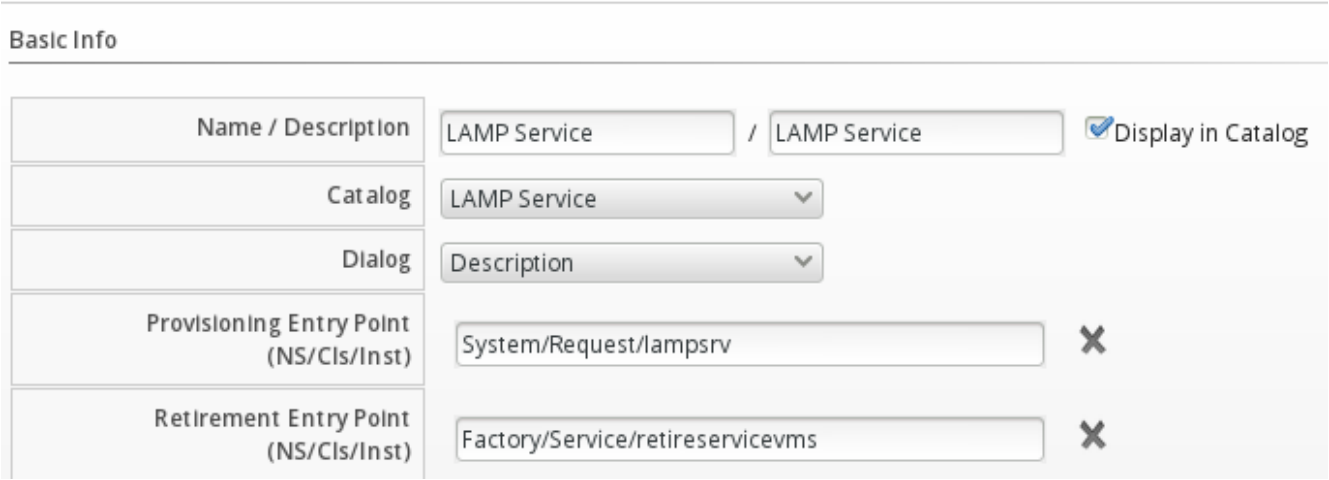


Figure 5.3-1: Catalog Items

In the right window pane, select the **Details** tab. In the **Basic Info** input box, within the **Long Description** text box, provide a detailed description for the catalog item. For the reference environment the following input is provided:

Input Box	Option	Value
Basic Info	Long Description	This is a LAMP service catalog item for use by the cf-dev user.

Table 5.3-3: Catalog Item Details Settings

At the top of the right window pane select the **Request Info** tab. In the **Request Info** input box there are five tabs: **Catalog**, **Environment**, **Properties**, **Customize**, and **Schedule**.

Select the **Catalog** tab. In the **Select** input box choose the image for the catalog item next to **Name.** In the **Naming** input box provide an **Instance Name**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Select	Name	cf-lamp

Table 5.3-4: Catalog Item Request Info - Catalog

Select the **Environment** tab. In the **Placement** input box place a check next to **Choose Automatically**.

Select the **Properties** tab. In the **Properties** input box select and **Instance Type** and **Guest Access Key Pair**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Properties	Instance Type	m1.small
	Guest Access Key Pair	cf-lamp

Table 5.3-5: Catalog Item Request Info - Catalog

Select the **Customize** tab. Scroll down to the **Customize Template** input box and select a **Script Name**. For the reference environment *LAMP* is the customization script chosen.

In the **Selected Template Contents** input box, contents of the *LAMP* script are displayed in the **Script Text** text box.

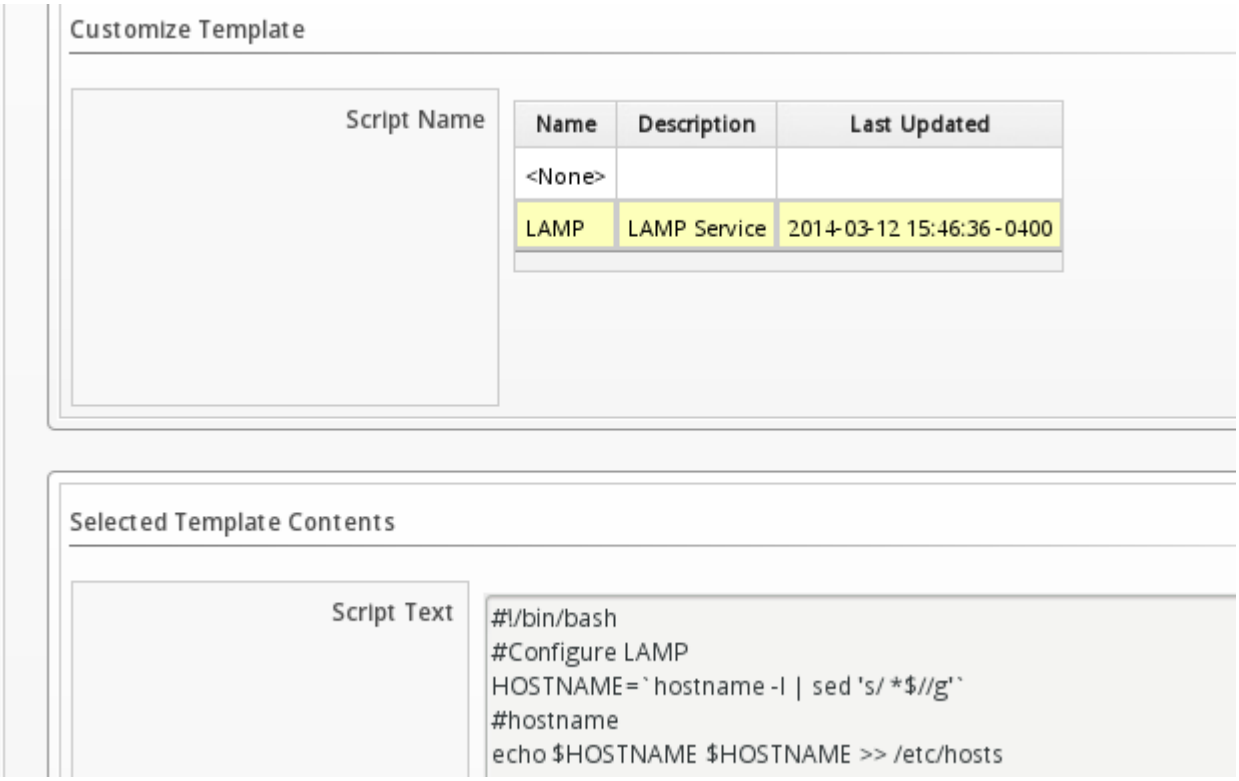


Figure 5.3-2: Catalog Item Request Info - Customize

The remainder of the input boxes under the **Customize** tab are left unchanged from defaults.

Select the **Schedule** tab. In the **Lifespan** input box choose a **Time until Retirement**. If selecting anything other than *Indefinite* an additional option appears for **Retirement Warning**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Lifespan	Time until Retirement	1 month
	Retirement Warning	1 week

Table 5.3-6: Catalog Item Request Info - Catalog

Click the **Add** button to complete the catalog item creation.

Verify the service catalog creation by selecting **Service Catalogs** on the left window pane. On the right window pane an entry for the service catalog should appear along with the ability to **Order** the catalog item.

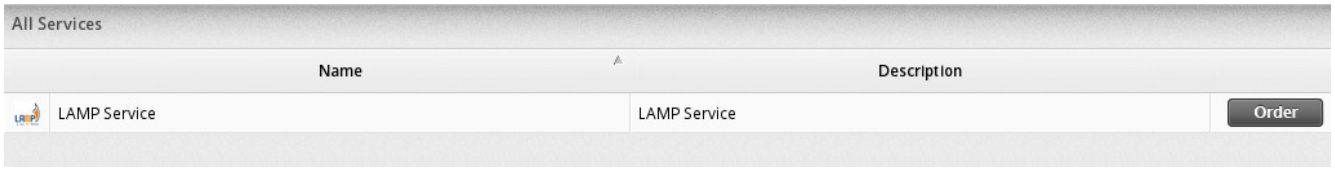


Figure 5.3-3: Service Catalog

5.4 Ordering an Application as a Self Service User

As the *cf-dev* user, login to the CFME Console and navigate to **Services**, **Catalogs**. On the right window pane the previously created catalog item, *LAMP Service*, is available. Click **Order** to place an order for the *LAMP Service* catalog item.

Figure 5.4-1: Self-Service Order

In the **Description** input box provide a description of the service in the **Service Description** text box.

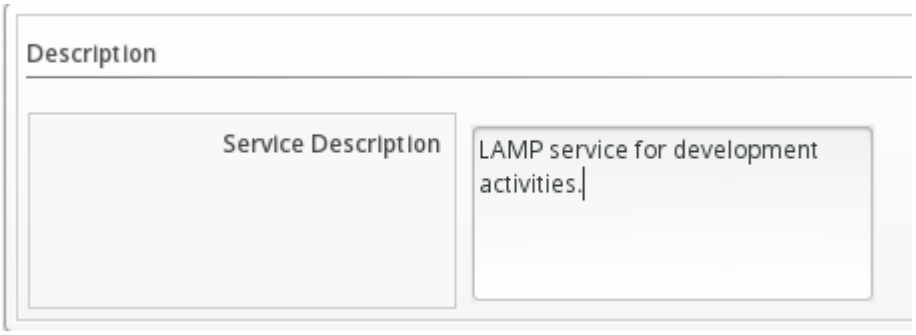


Figure 5.4-2: Self-Service Order – Service Description

Click **Submit** to complete the service order.

The self-service user is automatically redirected to **Services**, **Requests**. The status of the service order request is displayed.

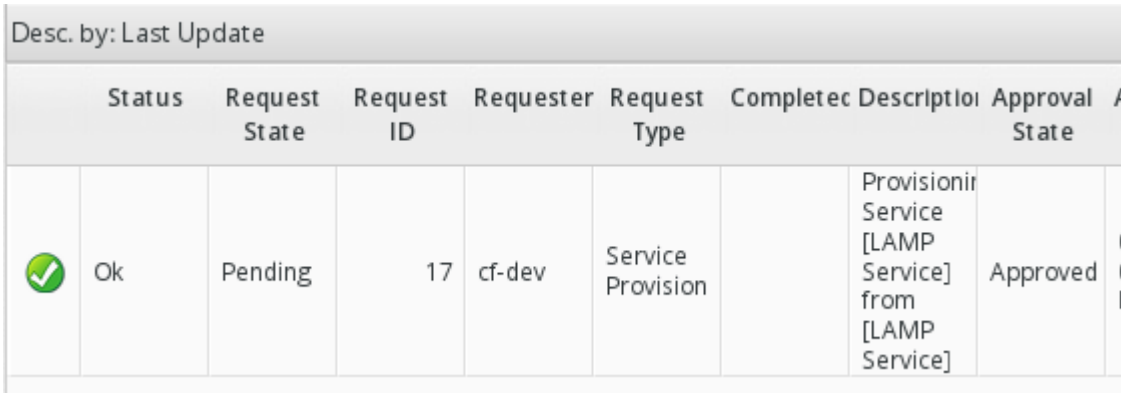


Figure 5.4-3: Self-Service Order Status

Upon provisioning completion, the deployed service appears under **Services**, **My Services**.

Figure 5.4-4: Self-Service – My Services

Additionally, the *cf-dev* self-service user can navigate to **Clouds**, **Instances** and locate the newly provisioned LAMP service hosted on the RHEL OSP cloud provider.

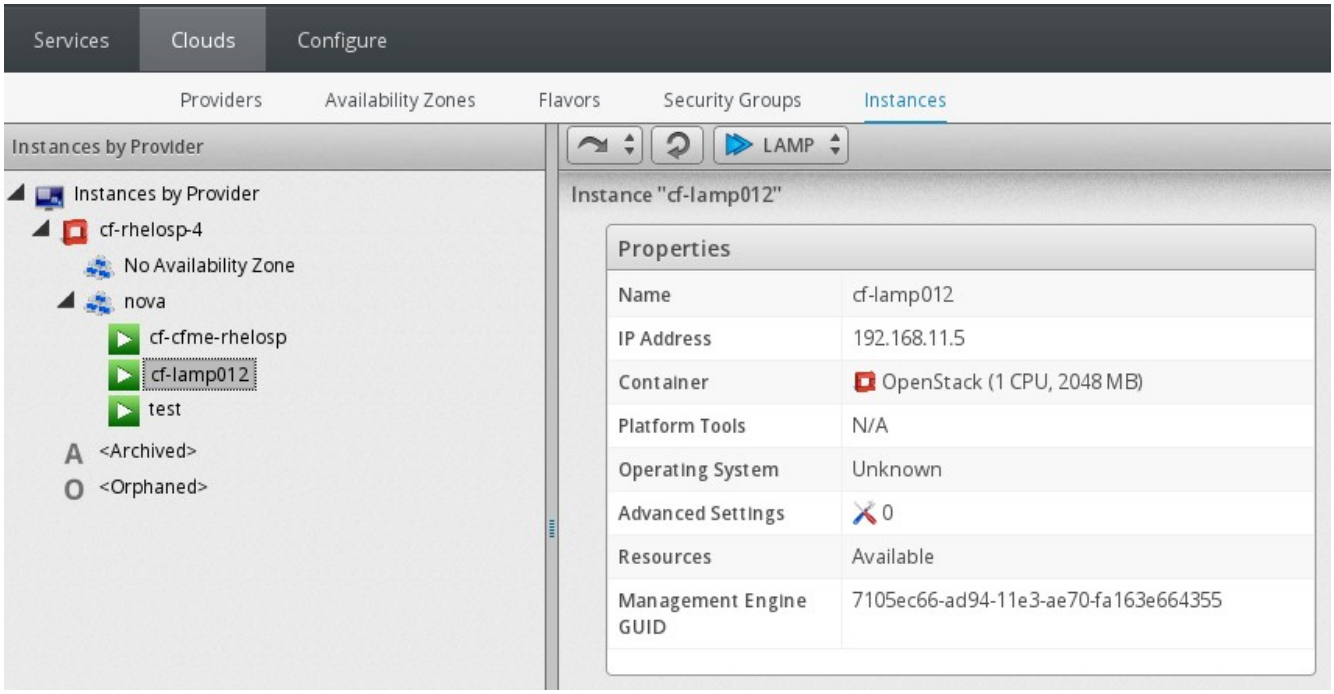


Figure 5.4-5: Self-Service Cloud Instance

Checking the RHEL OSP environment, the instance is present and available.

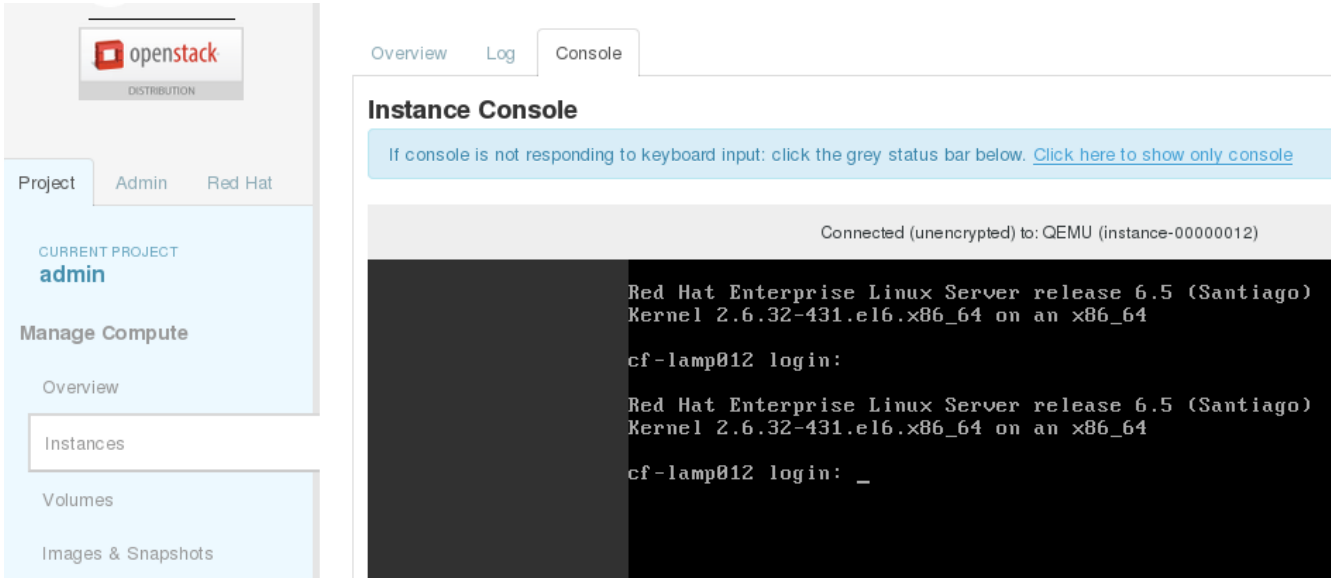


Figure 5.4-6: RHEL OSP Instance

The self-service user *cf-dev* is able to access the LAMP stack for the created service.

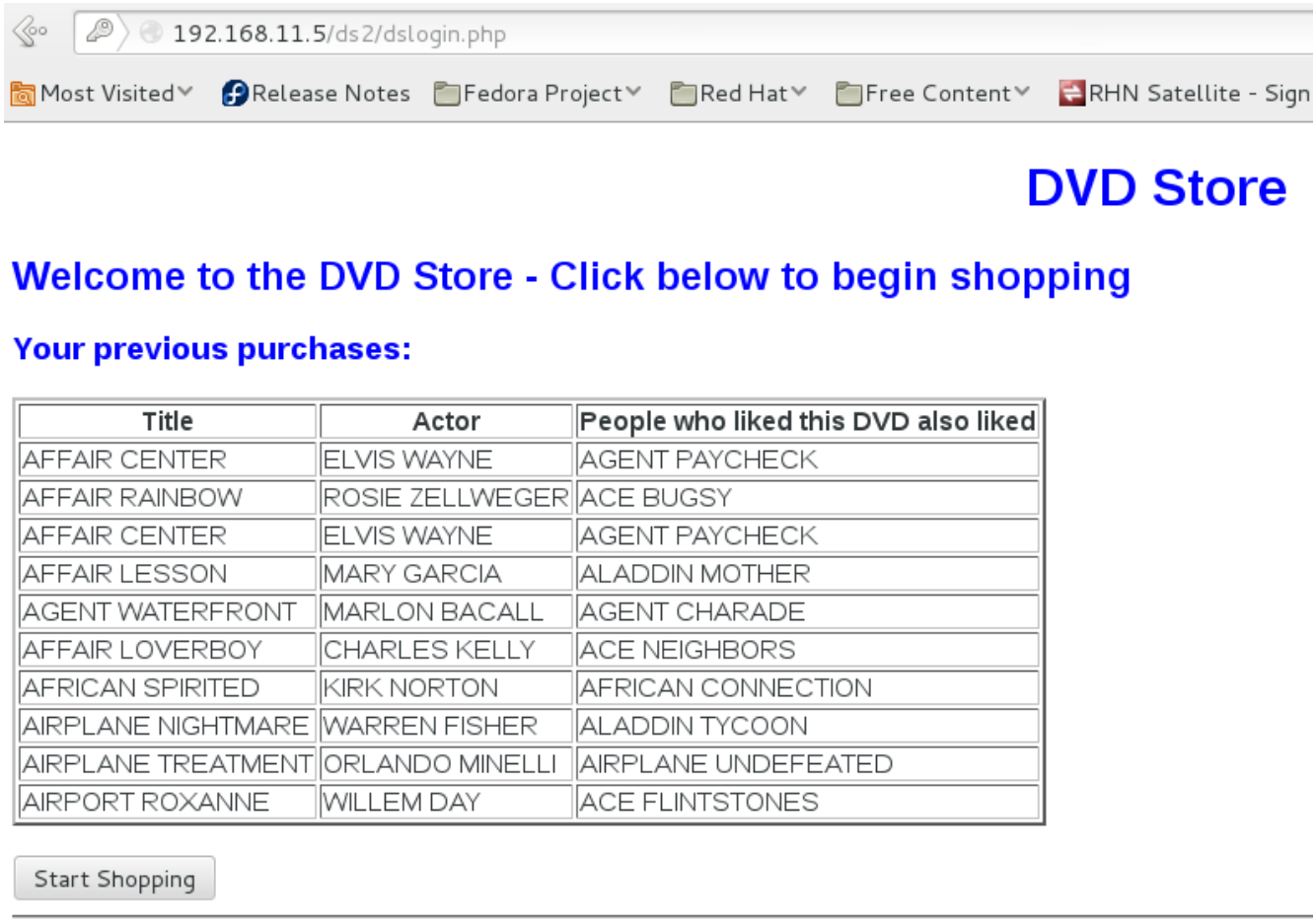
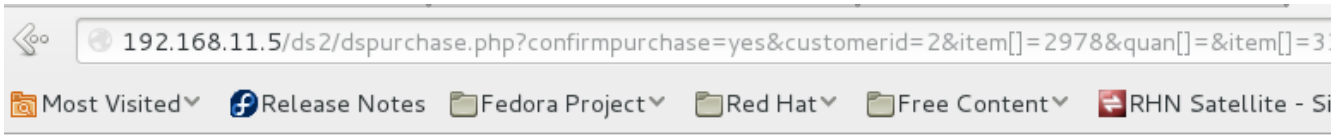


Figure 5.4-7: LAMP Stack Functional Testing - Login

LAMP stack functionality verification.



DVD Store

Purchase complete

Item	Quantity	Title	Actor	Price
1	1	ADAPTATION WISDOM	CHARLES JOVOVICH	$22.99
2	1	AFFAIR GROSSE	PARKER COSTNER	$24.99
3	1	AFFAIR WIZARD	KATHARINE BALE	$22.99
4	1	AFRICAN ROSES	MARLENE DEGENERES	$25.99
5	1	AIRPORT DINOSAUR	ELLEN SMITH	$19.99
6	1	ALABAMA BAREFOOT	HUMPHREY NEESON	$26.99
7	1	ALADDIN CARIBBEAN	SIDNEY MCCONAUGHEY	$23.99
			Subtotal	$167.93
			Tax (8.25%)	$13.85
			Total	$181.78

$181.78 charged to credit card (Discover), expiration 2011/06

Order Completed Successfully --- ORDER NUMBER: 12003

Figure 5.4-8: LAMP Stack Functional Testing - Purchase

As the *cf-dev* self-service user provisioned a new service, the cloud administrator receives e-mail notification[21] for the associated instance.

```
Hello,

Your request to provision a virtual machine was approved and completed on
Monday, March 17, 2014 at 01:25AM.

Virtual machine cf-lamp012 will be available in approximately 15 minutes.

For Windows VM access is available via RDP and for Linux VM access is
available via putty/ssh, etc. Or you can use the Console Access feature
found in the detail view of your VM. As the designated owner you will
receive expiration warnings at this email address: bthurber@redhat.com

If you are not already logged in, you can access and manage your virtual
machine here https://192.168.11.2/vm_or_template/show/26'

If you have any issues with your new virtual machine please contact Support.

Thank you,
Virtualization Infrastructure Team
```

Note: Automation Engine and Notifier server roles, along with SMTP settings, must be configured for e-mail notification and service automation functionality. Refer to *CloudForms 3.0: Management Engine 5.2 Settings and Operations*[22] guide.

21 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Lifecycle_and_Automation_Guide/Setting_Provisioning_Notification_Email_Addresses.html

22 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Settings_And_Operations/index.html

6 High Availability Using Control Policies

Red Hat CloudForms provides control policies[23] that enable actions against a set of events that may occur. These policies can be assigned to individual instances and virtual machines or assigned to providers. Actions which take place range from virtual machine management operations, tagging resources, executing external scripts, to invoking custom automation.

For the reference environment a control policy is used to ensure a LAMP application is made available in the event an unexpected retirement or suspend action occurs to an existing instance running a LAMP application.

The following steps are taken:

- Create a custom action
- Create a custom condition
- Create a control policy and assign the custom condition and action
- Assign and verify control policy functionality

6.1 Create a Custom Action

To create a custom action, login to the CFME Console as the *admin* user and navigate to **Control**, **Explorer**. On the left window pane click **Actions**. On the right window pane click **Configuration** and select **Add a new Action**.

In the **Basic Information** input box provide a **Name** and select an **Action Type**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Basic Information	Name	LAMP Provision
	Action Type	Invoke a Custom Automation

Table 6.1-1: Custom Action – Basic Information

Note: Selecting *Invoke a Custom Automation* **Action Type** enables additional input boxes.

In the **Object Details** input box, within the **Custom Automation** input box, provide a **Message** and **Request**. For the reference environment the following inputs are provided:

Input Box	Option	Value
Object Details	Message	create
	Request	lampsrv

Table 6.1-2: Custom Action – Object Details

23 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Control_Guide/chap-Policies.html#sect-Control_Policies

Note: The assigned **Message** and **Request** values invoke the automation model. *create* indicates the automate action to perform and *lampsrv* is the entry point into the automate model. In this case the entry point maps to: *Datastore/System/Request/lampsrv* and can be verified by navigating to **Automate**, **Explorer**.

For the reference environment **Attribute/Value Pairs** are not used therefore the input fields are left blank.

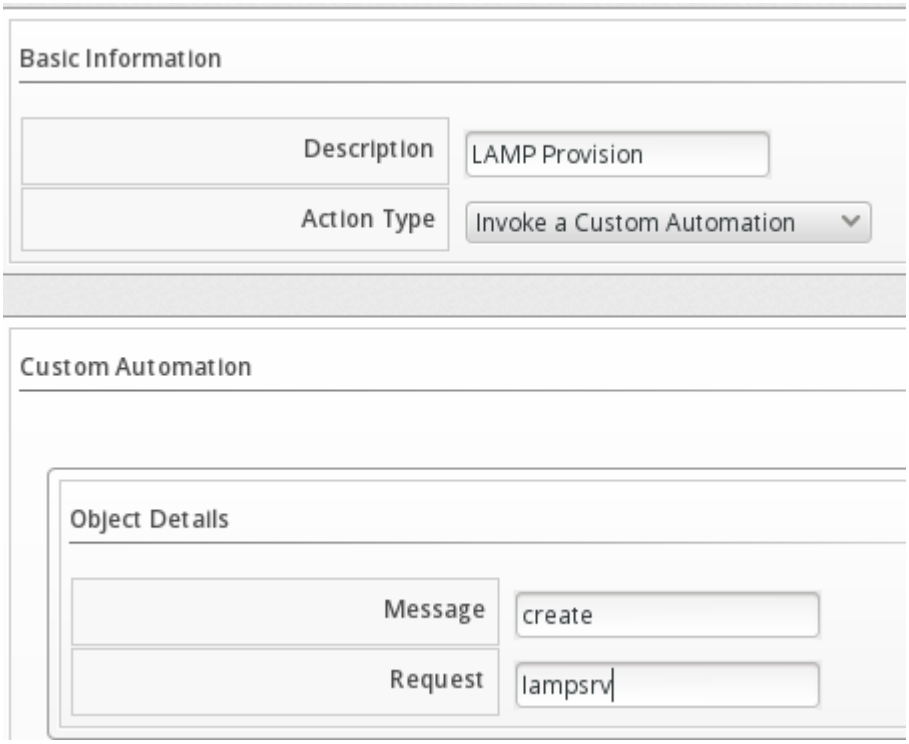


Figure 6.1-1: Custom Action Creation

Click **Add** to complete the custom action creation.

6.2 Create a Custom Condition

To create a custom condition, on the left window pane click **Conditions** and select **VM Conditions**. On the right window pane click **Configuration** and select **Add a New Vm Condition**.

In the **Basic Information** input box provide a **Description**. For the reference environment *Tag* is the provided input.

In the **Scope** input box click the pencil icon. A new selection box appears with a pull down menu. Select *Field* from the pull down options. An additional pull down menu appears. Select *VM and Instance : Name*. In the subsequent pull down select *STARTS WITH*. In the text string box enter *cf-lamp*. Click the check mark to add the scope.

In the **Expression** input box, click ???. A new selection box appears with a pull down menu. Select *Tag* from the pull down options. An additional pull down menu appears. Select *VM and Instance.Red Hat Tags : Environment*.

A final pull down menu appears under **CONTAINS**. Select *CloudForms*. In the **Notes** input box provide a description for the condition. *LAMP instance condition* is the input provided for the reference environment. Click the check mark to add the expression.

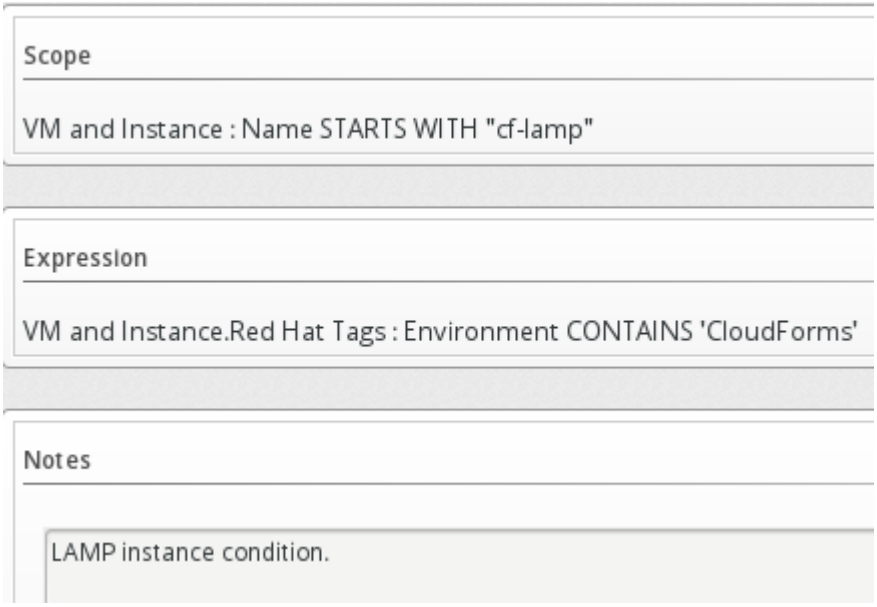


Figure 6.2-1: Custom Condition Creation

Click **Add** to complete the custom condition creation.

6.3 Create a Control Policy

To create a control policy, on the left window pane click **Policies** and select **Vm Control Policies**. On the right window pane click **Configuration** and select **Add a New Vm Control Policy**.

In the **Basic Information** input box provide a **Description** and place a check mark next to **Active**. For the reference environment *LAMP Automate* is the description input provided.

In the **Scope** input box select *Tag* from the pull down options. An additional pull down menu appears. Select *VM and Instance.Red Hat Tags : Environment*.

A final pull down menu appears under **CONTAINS**. Select *CloudForms*. In the **Notes** input box provide a description for the condition. *LAMP Automate policy scope* is the input provided for the reference environment. Click the check mark to add the scope.

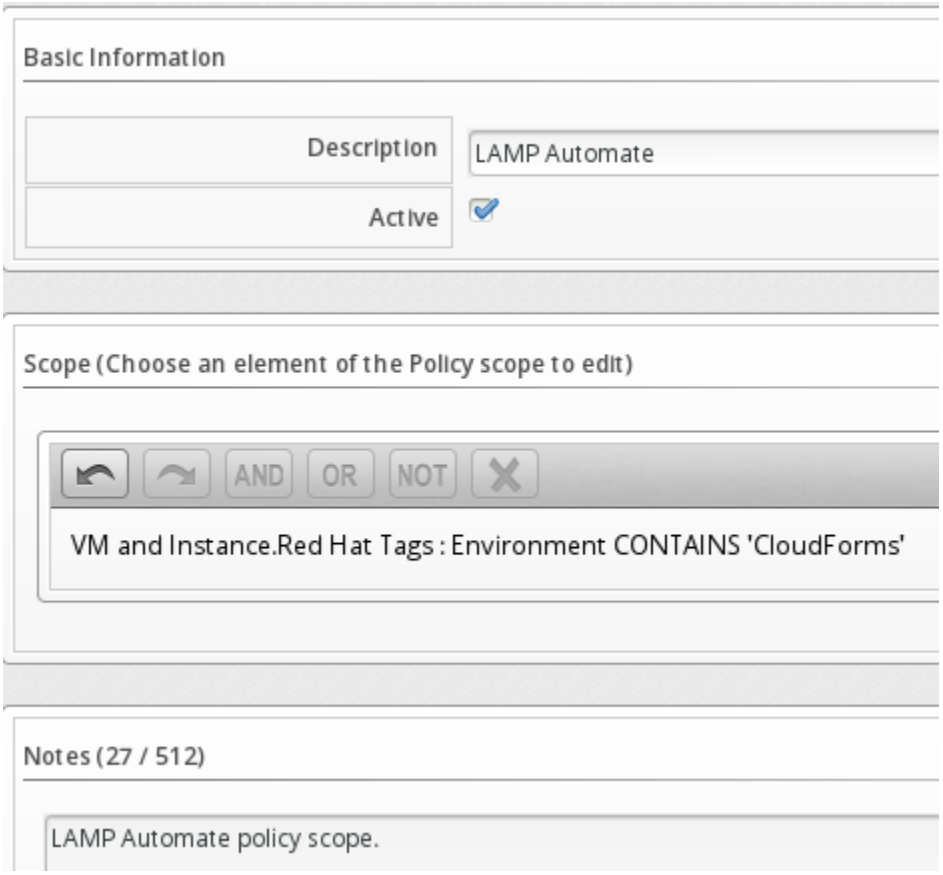


Figure 6.3-1: Control Policy Creation

With the newly created control policy selected on the left window pane, on the right window pane click **Configuration** and select **Edit this Policy's Condition assignments**.

In the **Condition Selection** input box, under **Available VM and Instance Conditions**, highlight the previously created condition called *Tag* and click the right arrow to move it under **Policy Conditions**.

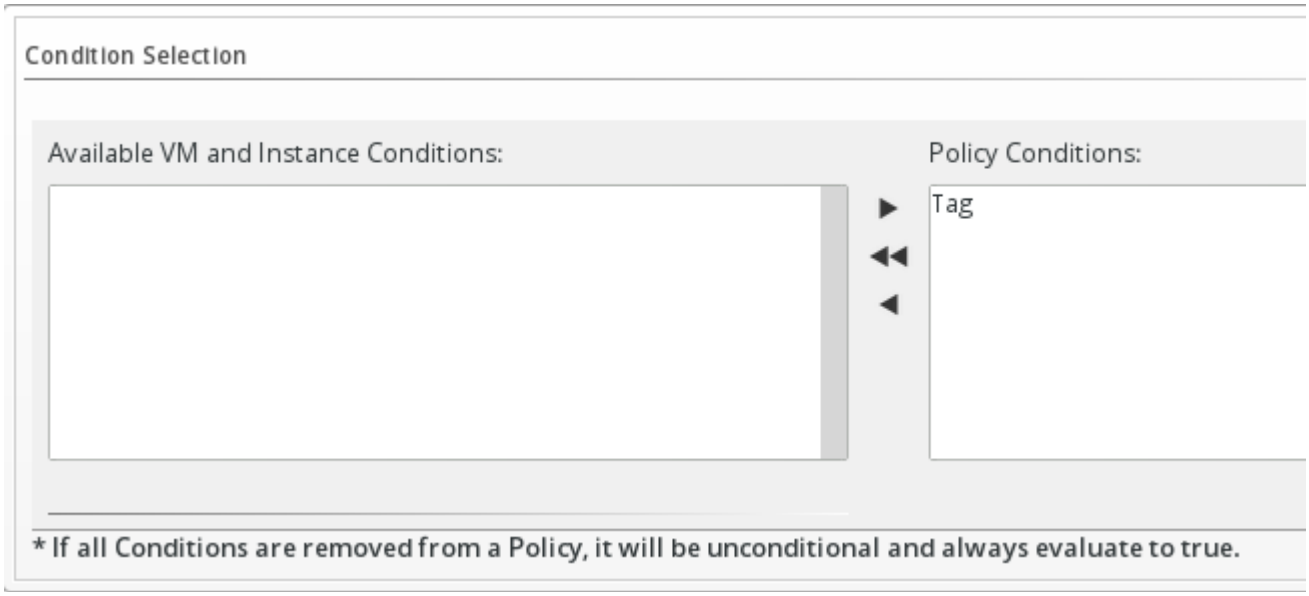


Figure 6.3-2: Control Policy Condition Selection

Click **Save** to complete the policy condition assignment.

With the the **LAMP Automate** policy highlighted in the left window pane, on the right window pane click **Configuration** and select **Edit this Policy's Event assignments**. In the **Event Selection** input box, scroll down and place a check mark next to *VM Retired* in the **VM Lifecycle** input box and a check mark next to *VM Suspend Request* in the **VM Operation** input box.

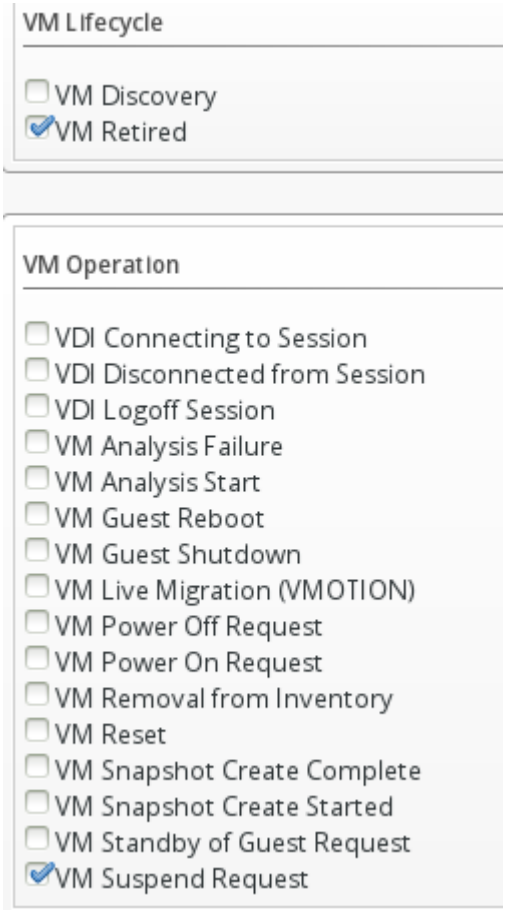


Figure 6.3-3: Control Policy Event Selections

Click **Save** to complete the policy event assignment.

On the left window pane, select a newly assigned policy event, **VM Retired**. On the right window pane click **Configuration** and select **Edit Actions for this Policy Event**.

Under the **Order of Actions if ALL Conditions are True** input box, under **Available Actions**, locate and highlight the previously created action, *LAMP Provision*, and click the right arrow to move it under **Selected Actions**.

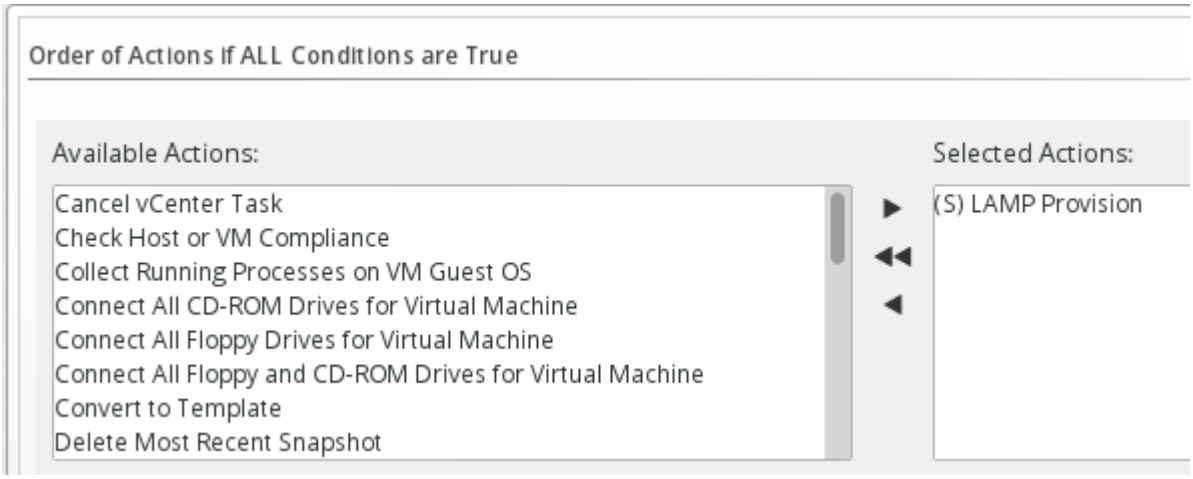


Figure 6.3-4: Control Policy Event Action Assignment

For the reference environment no actions are assigned under the **Order of Actions if ANY Conditions are False** input box.

Click **Save** to complete the action assignment for the event. Perform the same for the **VM Suspend Request** event.

> **Note:** Multiple actions can be assigned to the same event. The order of execution can be performed synchronously or asynchronously[24] and is indicated by an *A* or *S* designation next to each assigned action.

The final step in the control policy creation is to create a policy profile. On the left window pane select **Policy Profiles**. On the right window pane click **Configuration** and select **Add a New Policy Profile**.

In the **Basic Information** input box provide a **Description**. For the reference environment *LAMP HA* is the input provided.

Under **Available Policies** in the **Policy Selection** input box select the newly created policy and click the right arrow to move it under **Profile Policies**.

24 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Control_Guide/chap-Policies.html#To_assign_an_Action_to_an_Event

In the **Notes** input box provide a description for the policy profile. For the reference environment *LAMP HA policy profile* is the input provided.

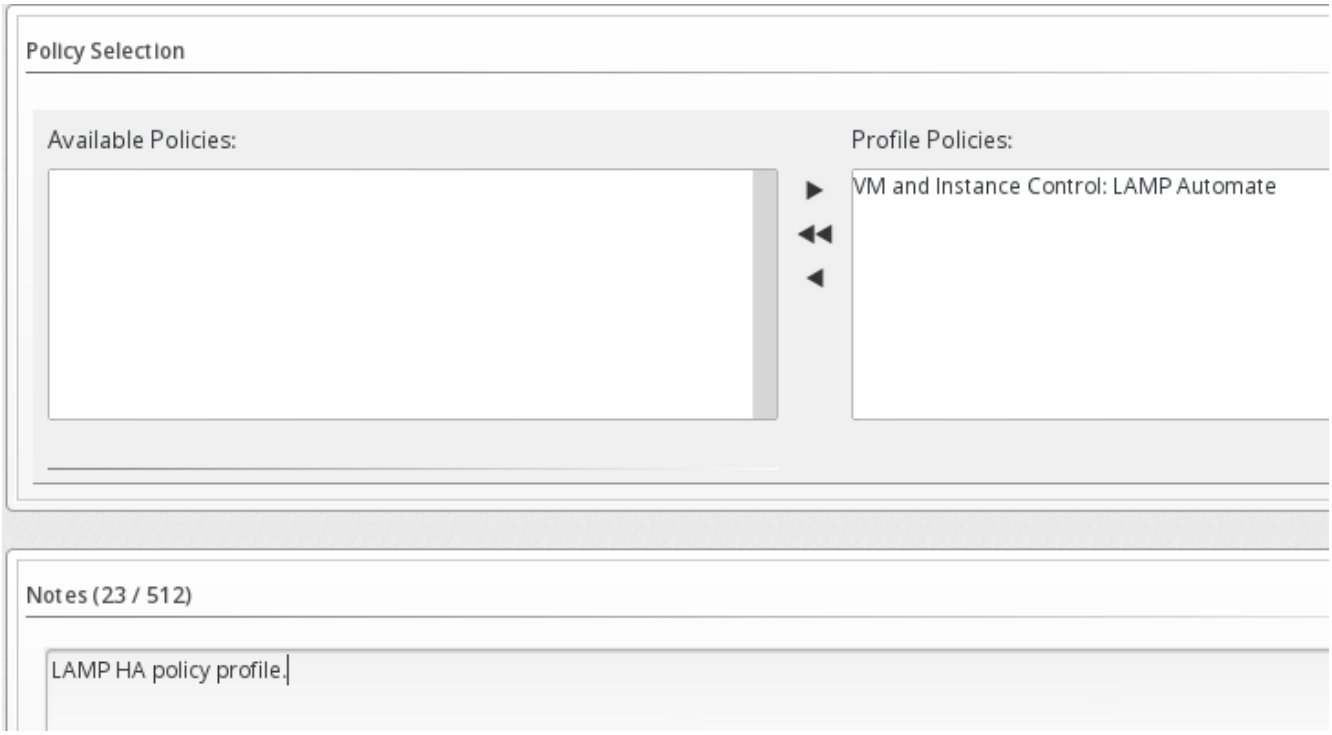


Figure 6.3-5: Control Policy Profile Creation

Click **Add** to complete the policy profile creation.

6.4 Verify and Assign the Control Policy

To simulate the control policy, navigate to **Control**, **Simulation**. On the left window pane select the **Type** and **Event** within the **Event Selection** input box. In the **VM Selection** input box choose a resource. For the reference environment the following inputs are provided:

Input Box	Option	Value
Basic Information	Type	VM Operation
	Event	VM Suspend Request
VM Selection	By Provider	cf-rhelosp-4

Table 6.4-1: Control Policy Simulation Values

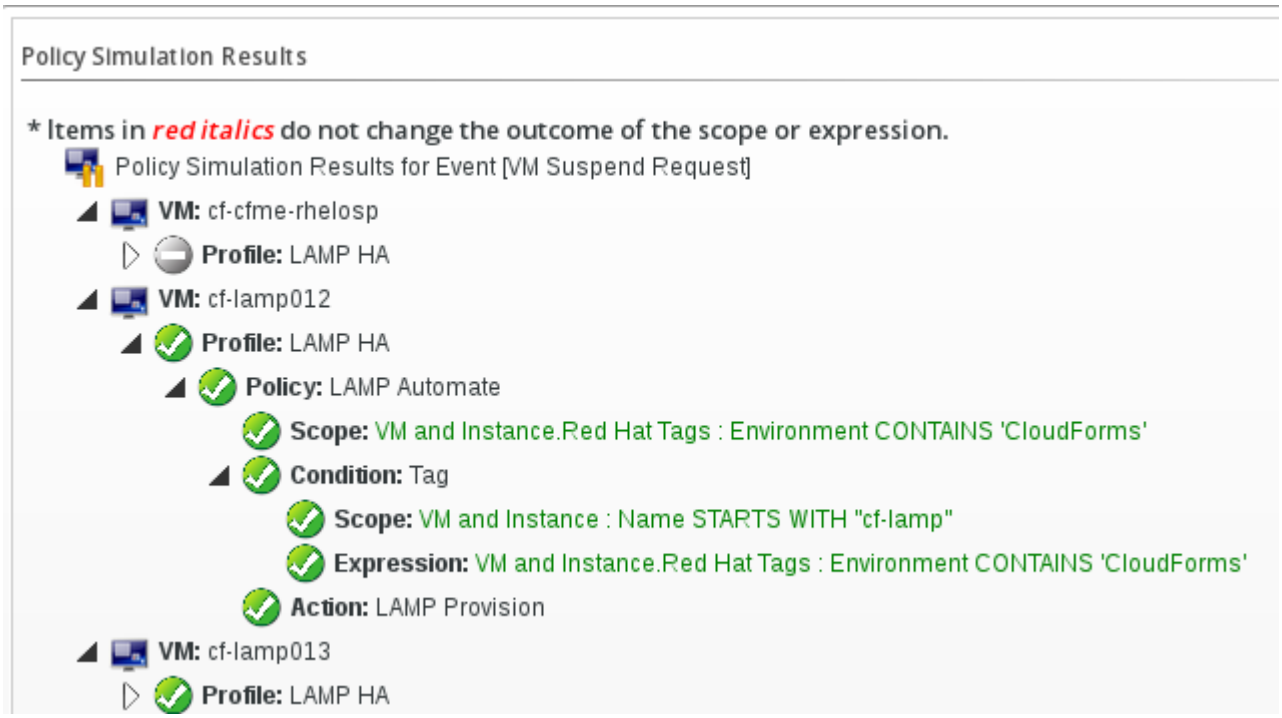


Figure 6.4-1: Control Policy Simulation Results

Assign the control policy to the RHEL OSP provider by navigating to **Clouds**, **Providers**. On the right window pane place a check mark next to the desired provider, click **Policy** and select **Manage Policies**.

In the **Select Policy Profiles** input box place a check mark next to the desired policy profile. For the reference environment the policy profile selected is *LAMP HA*.

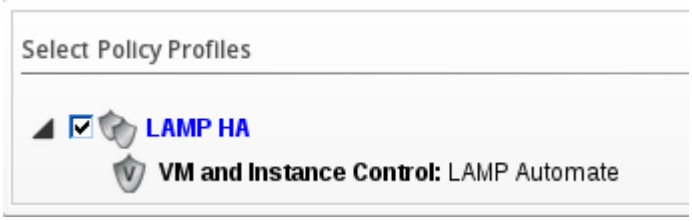


Figure 6.4-2: Control Policy Cloud Provider Assignment

Click **Save** to complete the policy profile assignment.

Test the control policy by navigating to **Clouds**, **Instances**. On the right window pane click on any instance with a name that begins with *cf-lamp*. For the reference environment *cf-lamp012* is selected.

Within the instance details window, click **Power** and select **Suspend**. A message window appears asking for confirmation to suspend the instance. Click **OK** to proceed.

After a few minutes the instance is suspended and the action assigned to the control policy initiates creating a new LAMP instance.

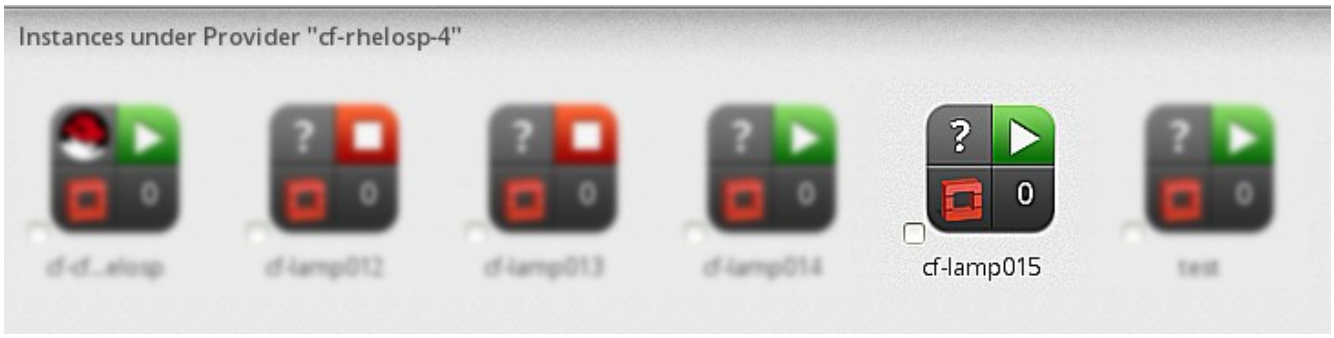


Figure 6.4-3: CloudForms Control Policy Verification

☐	cf-lamp015	cf-lamp	192.168.11.9	m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	cf-lamp	Active
☐	cf-lamp014	cf-lamp	192.168.11.8	m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	cf-lamp	Active
☐	cf-lamp013	cf-lamp	192.168.11.6	m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	cf-lamp	Suspended
☐	cf-lamp012	cf-lamp	192.168.11.5 10.16.11.202	m1.small | 2GB RAM | 1 VCPU | 20.0GB Disk	cf-lamp	Suspended

Figure 6.4-4: RHEL OSP Instance Verification

The cloud administrator is notified of the new instance creation:

```
Hello,

Your request to provision a virtual machine was approved and completed on
Tuesday, March 18, 2014 at 05:35PM.

Virtual machine cf-lamp015 will be available in approximately 15 minutes.

For Windows VM access is available via RDP and for Linux VM access is
available via putty/ssh, etc. Or you can use the Console Access feature
found in the detail view of your VM. As the designated owner you will
receive expiration warnings at this email address: bthurber@redhat.com

If you are not already logged in, you can access and manage your virtual
machine here https://192.168.11.2/vm_or_template/show/29'

If you have any issues with your new virtual machine please contact Support.

Thank you,
Virtualization Infrastructure Team
```

7 Integrating RHEL OSP Metering with CloudForms Capacity and Utilization

Red Hat Enterprise Linux OpenStack Platform 4.0 introduces metering[25] capabilities to monitor running instances. Monitored items include disk, CPU, and network utilization for example.

CloudForms 3.0 provides the ability to tie into RHEL OSP metering via Capacity and Utilization. This integration allows for monitoring resource usage on a hourly, daily, weekly, and monthly basis. Combined with chargeback[26] and reporting, this becomes a very powerful tool to view, manage, and control cost be it on-premise or within a public cloud.

7.1 RHEL OSP Metering Configuration

With RHEL OSP 4, Ceilometer installation is integrated with packstack and Foreman[27]. For the reference environment Ceilometer components[28] run on the following distributed nodes:

Component	Node
ceilometer-agent-compute	cf-rhos-nova-1 (compute)
ceilometer-agent-central	cf-rhos-glance (controller)
ceilometer-collector	cf-rhos-glance (controller)
Mongo database	cf-rhos-glance (controller)
API Server	cf-rhos-glance (controller)

Table 7.1-1: Ceilometer Component Distribution

Additional information regarding installing and configuring Ceilometer for RHEL OSP 4 is found in *Red Hat Enterprise Linux OpenStack Platform 4: Installation and Configuration Guide*[29].

For the reference environment example RHEL OSP installation files are found in **Appendix F: RHEL OSP Files**.

To verify if metering is operational, perform the following commands on the controller node:

```
# source keystonerc_admin
[root@cf-rhos ~(keystone_admin)]#
```

25 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/End_User_Guide/ceilometer_cli_commands.html
26 https://access.redhat.com/site/documentation/en-US/CloudForms/3.0/html/Management_Engine_5.2_Insight_Guide/sect-Chargeback.html
27 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/index.html
28 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/Metering_-_Ceilometer_Technical_Preview.html
29 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/chap-Installing_the_OpenStack_Metering_Service.html

```
# ceilometer meter-list
+--------------------------+------------+----------+-----------------------
-------------------------------------------------+--------------------------
------+----------------------------------+
| Name                     | Type       | Unit     | Resource ID
| User ID                          | Project ID                          |
+--------------------------+------------+----------+-----------------------
-------------------------------------------------+--------------------------
------+----------------------------------+
| cpu                      | cumulative | ns       |
20322ebc-3add-4356-b0fe-b51411562463                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
| cpu                      | cumulative | ns       |
3c14b061-b454-4a60-98ce-90281ac94226                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
| cpu                      | cumulative | ns       |
3ee28cc1-afe8-43fb-a1b8-62f5c72fbff9                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
| cpu                      | cumulative | ns       |
53ec6bef-b700-4fa8-9ca0-f8d7ae01e2e5                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
| cpu                      | cumulative | ns       |
a7815aa7-ba82-4956-b416-78956b10280f                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
| cpu                      | cumulative | ns       |
b55a99dc-76d1-404c-9957-4dce489415f9                                  |
e41aca10858942ed8b04f06d3863549f | b1f17a38b41046db8506418ce4c1c13c |
                            <output truncated>
```

Output includes *cpu*, *disk*, and *network* statistics for each running instance.

7.2 CloudForms Capacity and Utilization

Refer to **Section 4.4.5.6 Capacity and Utilization Settings** for configuration steps regarding CloudForms 3.0 Capacity and Utilization support with RHEL OSP 4.

To verify Capacity and Utilization is working within CFME for RHEL OSP, access the CFME Console using the *admin* account. Hover over **Clouds** and select **Instances**. On the left window pane under **Instances by Provider** navigate to and select an instance running within the RHEL OSP 4 environment.

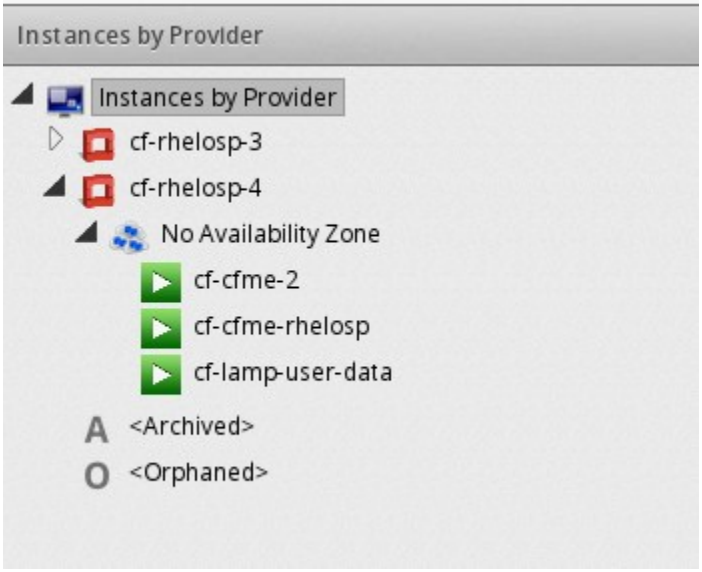


Figure 7.2-1: CFME Managed Instances

On the right window pane within the instance details, click **Monitoring** and select **Utilization**. Within a few moments utilization data is displayed in a graphical format for the running instance.

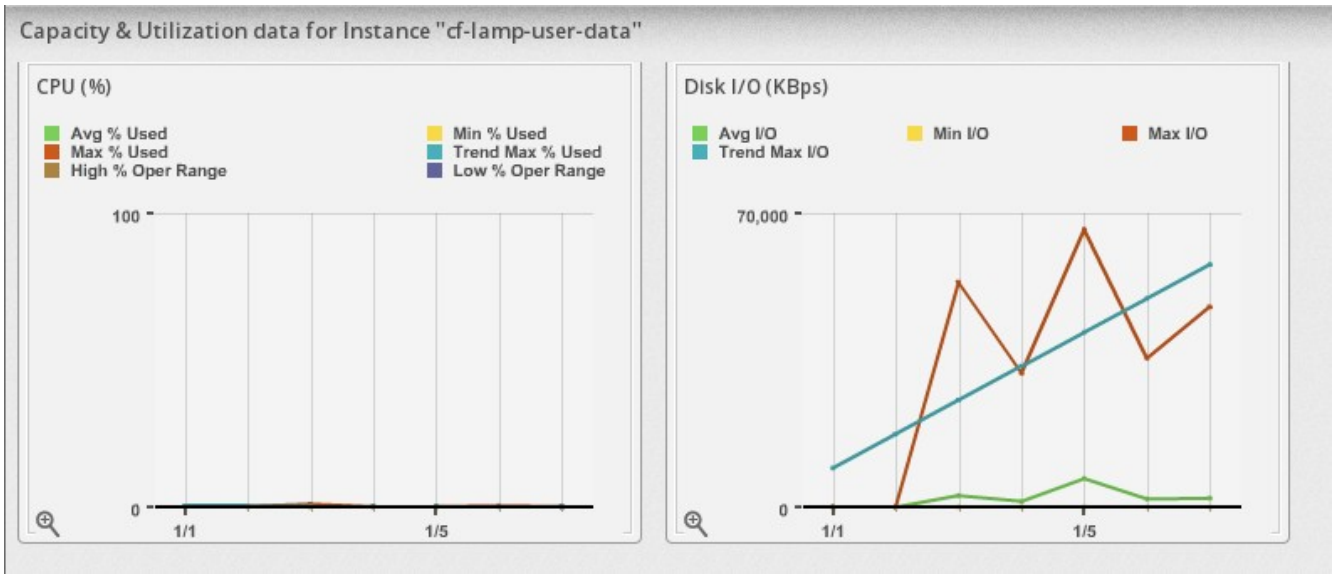


Figure 7.2-2: Instance Capacity and Utilization Data

Note: Utilization data takes time to populate for capacity and utilization. To view the most recent collection select *Most Recent Hour* for the **Interval** to display within the **Options** input box.

7.2.1 Troubleshooting Capacity and Utilization

If no data appears, either enough time has not passed for collection to complete or an issue exists with communication or configuration between CloudForms and RHEL OSP.

Verify Ceilometer is working properly within the RHEL OSP environment and CloudForms is able to communicate with *qpid*[30] on TCP port 5672.

If default credentials are not used for AMQP during the RHEL OSP install, configure the Cloud Provider to use the defined credentials. As *admin*, navigate to **Clouds**, **Providers** and select the RHEL OSP provider. Click **Configuration** and choose **Edit this Cloud Provider**.

Figure 7.2.1-1: Edit Cloud Provider

Inside the **Credentials** input box select the **AMQP** tab. Provide the associated *User ID* and *Password*. Click **Validate** to verify successful entries and **Save** to complete.

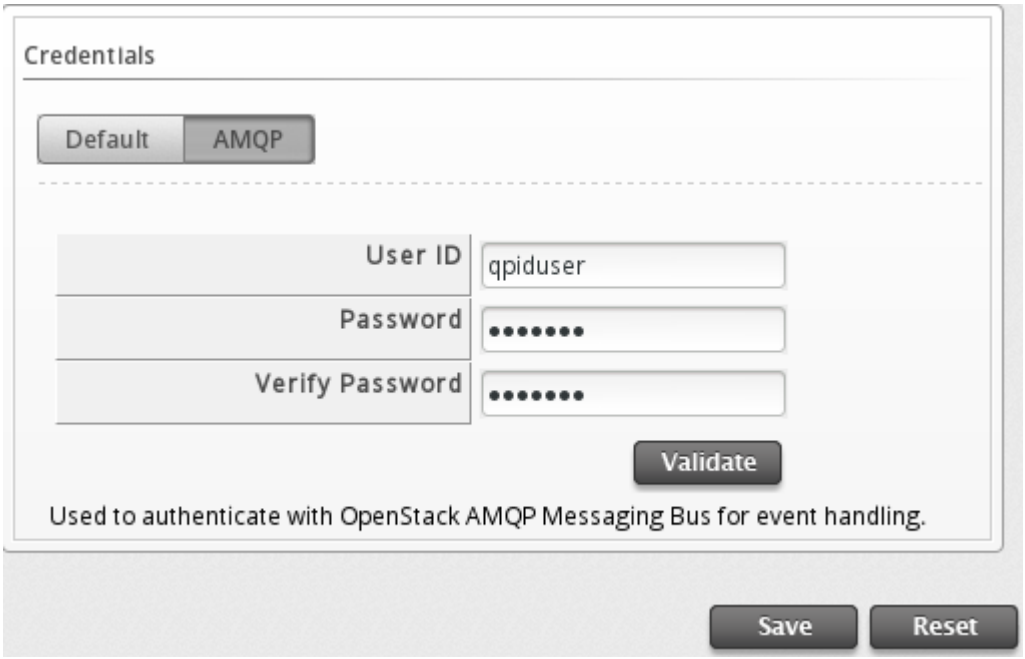


Figure 7.2.1-2:

Additional debugging information can be found in **Appendix G Troubleshooting**.

30 http://docs.openstack.org/developer/nova/devref/rpc.html

7.3 Chargeback and Reporting

The chargeback feature provides the monetary calculation of managed resource charges based on owner or company tag to include memory, storage, network, and CPU utilization. CloudForms Management Engine provides a default set of rates for calculating chargeback costs, however a custom set of rates can be created. To use this feature, capacity and utilization data collection must be enabled as discussed in **Section 4.4.5.6 Capacity and Utilization Settings**. For the reference environment the following items are configured:

- Custom chargeback rates
- Custom reporting against the *CloudForms* environment tag

Note: The *CloudForms* tag creation and assignment is discussed in **Section 4.4.5.4 Tags**.

7.3.1 Chargeback Rates

To create custom chargeback rates login to the CFME console with the *admin* account, hover over **Cloud Intelligence** and select **Chargeback**. Under the accordion menu in the left window pane select **Rates** and highlight **Compute**.

On the right window pane click **Configuration** and select **Add a New Chargeback Rate**.

In the **Basic Info** input box provide a **Description**. In the **Rate Details** input box define rates for each item. For the reference environment the following settings are used:

Input Box	Field	Value	Per Time	Per Unit
Basic Info	Description	CloudForms	N/A	N/A
Rate Details	CPU – Allocated CPU Count	0	Hourly	CPU
	CPU – Used CPU in MHz	.02	Hourly	MHz
	Disk I/O – Used disk I/O in KBps	.005	Hourly	KBps
	Fixed – Compute Cost 1	0	Hourly	N/A
	Fixed – Compute Cost 2	0	Hourly	N/A
	Memory – Allocated Memory in MB	0	Hourly	MB
	Memory – Used Memory in MB	0	Hourly	MB
	Network I/O – Used Network I/O in KBps	.005	Hourly	KBps

Table 7.3.1-1: Custom Compute Chargeback Rate Values

Click **Add** to complete.

Rate Details

Group	Description	Rate	Per Time	Per Unit
CPU	Allocated CPU Count	0	Hourly	Cpu
CPU	Used CPU in MHz	.02	Hourly	MHz
Disk I/O	Used Disk I/O in KBps	.005	Hourly	KBps
Fixed	Fixed Compute Cost 1	0	Hourly	
Fixed	Fixed Compute Cost 2	0	Hourly	

Figure 7.3.1-1: CloudForms Compute Chargeback Rates

Assign the custom chargeback rates to the environment

On the accordion menu in the left window pane select **Assignments** and highlight **Compute**.

On the right window pane in the **Basic Info** input box next to **Assign To**, click the pull down menu and select **Tagged VMs and Instances**. Next to **Tag Category**, click the pull down menu and choose *Environment*. For **Selections** next to **CloudForms**, click the pull down menu and choose *CloudForms* for the **Rate**. Click **Save** to complete.

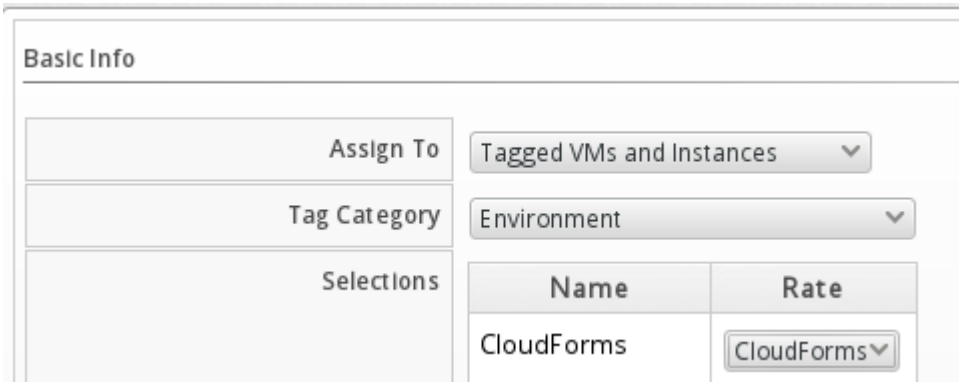


Figure 7.3.1-2: CloudForms Compute Chargeback Assignment

7.4 Reporting

Reporting is an integral piece to the chargeback model and provides a view into infrastructure costs. This is helpful to both the consumer and the owner of resources to understand demands and trends for budgeting needs.

7.4.1 Chargeback Report Configuration

To configure reporting for chargeback, login to the CFME console with the *admin* account, hover over **Cloud Intelligence** and select **Reports**. On the accordion menu in the left window pane select **Reports**.

On the right window pane click **Configuration** and select **Add a new Report**. A series of tabs is presented to include what information the report contains, format of the report, report interval, and more. Complete the settings under each tab before clicking the **Add** button at the bottom. The following settings are used for the reference environment:

Columns

The values provided for the **Columns** tab determine what information the report contains. For the reference environment the following settings are used:

Input Box	Field	Value
Basic Report Info	Menu Name	CloudForms
	Title	Chargeback
Configure Report Columns	Base the report on	Chargebacks
	Selected Fields	CPU Used Cost
		Disk I/O Used Cost
		Network I/O Used Cost
		Total Cost
Report Creation Timeout	Cancel after	<System Default>

Table 7.4.1-1: Chargeback Report Column Values

Chargeback report **Columns** settings.

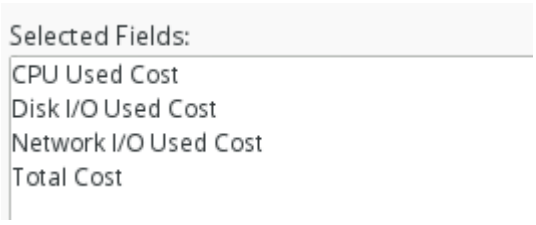


Figure 7.4.1-1: Chargeback Report Column Settings

Formatting

Settings under the **Formatting** tab determine how the information is displayed. For the reference environment defaults are used.

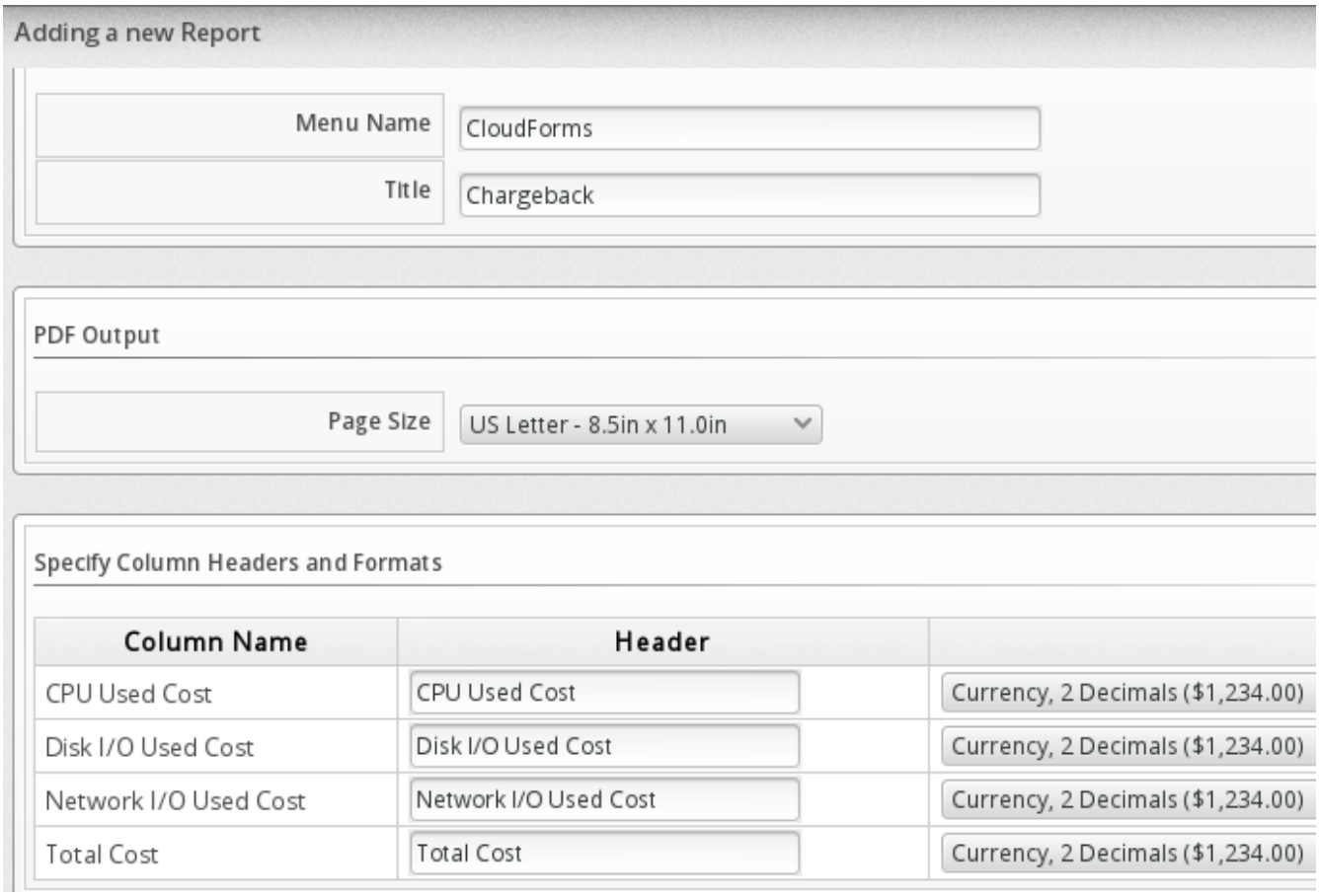


Figure 7.4.1-2: Chargeback Report Formatting Settings

Filter

Settings under the **Filter** tab are used to refine the data captured to include setting how far back captured data is displayed. The following settings are used for the reference environment:

Input Box	Field	Value
Basic Report Info	Menu Name	Chargeback
	Title	CloudForms
Chargeback Filters	Show Costs by	Red Hat Tag
	Tag Category	Environment
	Tag	CloudForms
	Group by	Date
Chargeback Interval	Show Costs by	Day
	Ending with	Yesterday going back 2 weeks
	Timezone	Eastern Time

Table 7.4.1-2: Chargeback Report Filter Values

Chargeback report **Filter** settings:

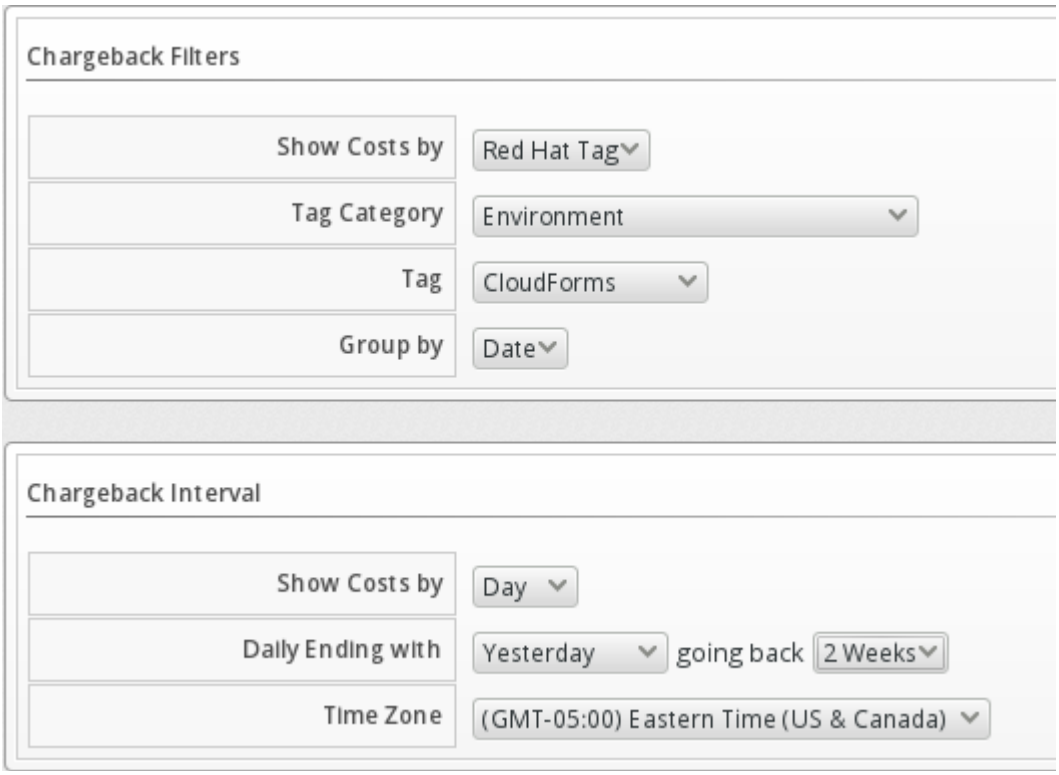
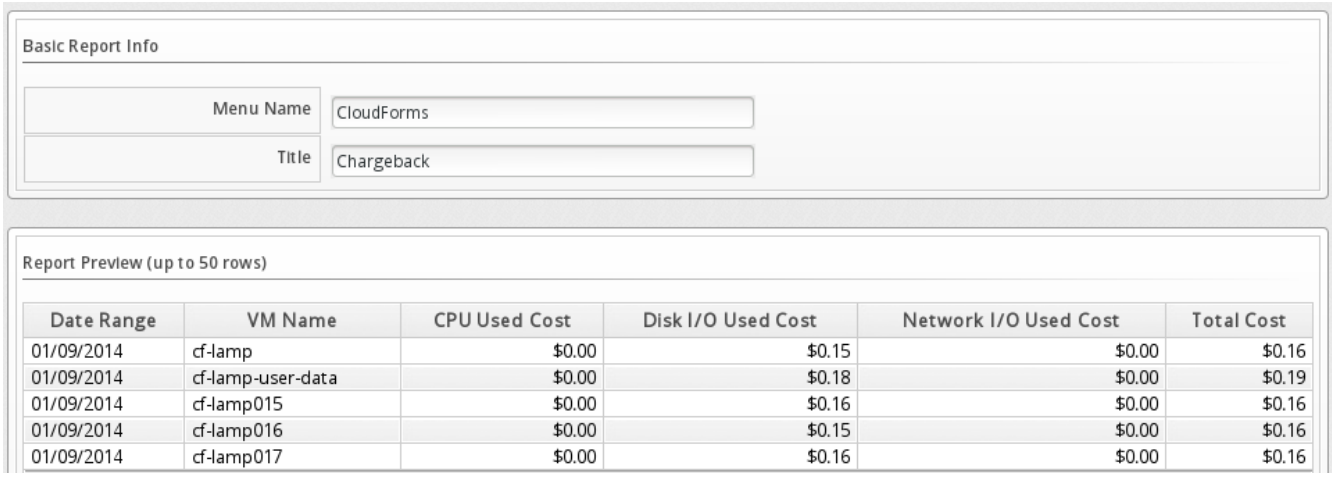


Figure 7.4.1-3: Chargeback Report Filter Settings

Preview

The **Preview** tab offers the ability to review the report as it would typically run before committing to adding the new report. Click the **Load** button to generate the preview. Click **Add** to complete the report creation.



Date Range	VM Name	CPU Used Cost	Disk I/O Used Cost	Network I/O Used Cost	Total Cost
01/09/2014	cf-lamp	$0.00	$0.15	$0.00	$0.16
01/09/2014	cf-lamp-user-data	$0.00	$0.18	$0.00	$0.19
01/09/2014	cf-lamp015	$0.00	$0.16	$0.00	$0.16
01/09/2014	cf-lamp016	$0.00	$0.15	$0.00	$0.16
01/09/2014	cf-lamp017	$0.00	$0.16	$0.00	$0.16

Figure 7.4.1-4: Chargeback Report Preview

The *CloudForms* chargeback report appears on the accordion menu in the left window pane under **Reports**, **All Reports**, **Red Hat (All EVM Groups)**, **Custom**.

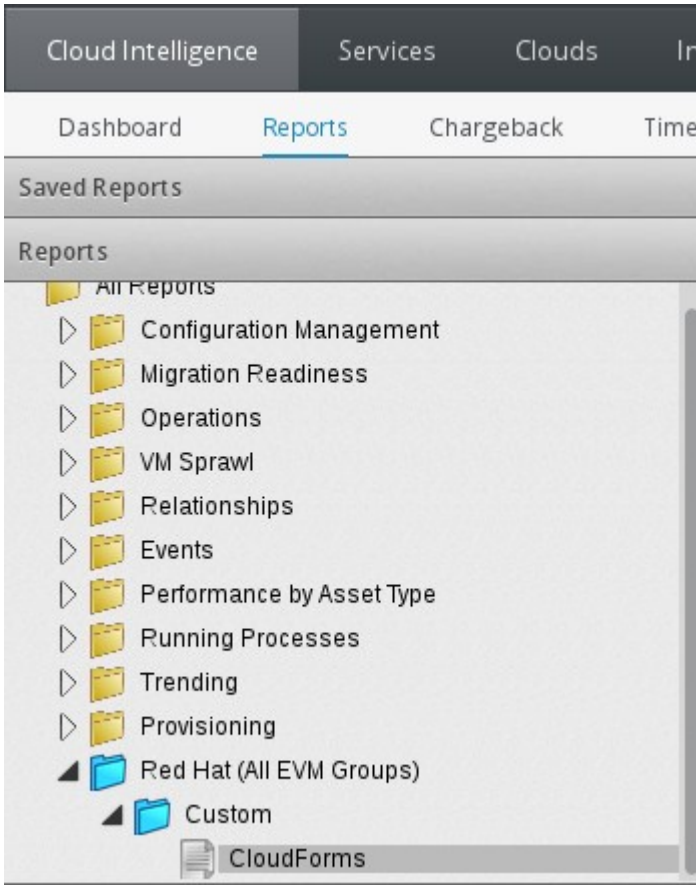


Figure 7.4.1-5: CloudForms Chargeback Report

7.4.2 Report Generation

Reports can be generated in two ways, by scheduling or by single instance manual invocation.

To schedule a report, on the accordion menu in the left window pane, select **Reports** and navigate to the *CloudForms* report under **All Reports**, **Red Hat (All EVM Groups)**, **Custom**.

Select *CloudForms* and on the right window pane and click **Configuration** and select **Add a new Schedule**.

In the **Basic Information** input box place a check mark next to **Active**.

In the **Timer** input box change the **Run** field to *Daily* every *Day*. Set the **Starting Date** and **Starting Time**.

In the **E-mail after Running** input box place a check mark next to **Send an E-mail**. Provide a **From** and **To** e-mail address.

In the **E-mail Options** input box place a check mark next to *Send if Report is Empty* and next to *PDF* for **Attachments**.

For the reference environment the scheduled report will run daily at 12PM EST. Click **Add** to complete.

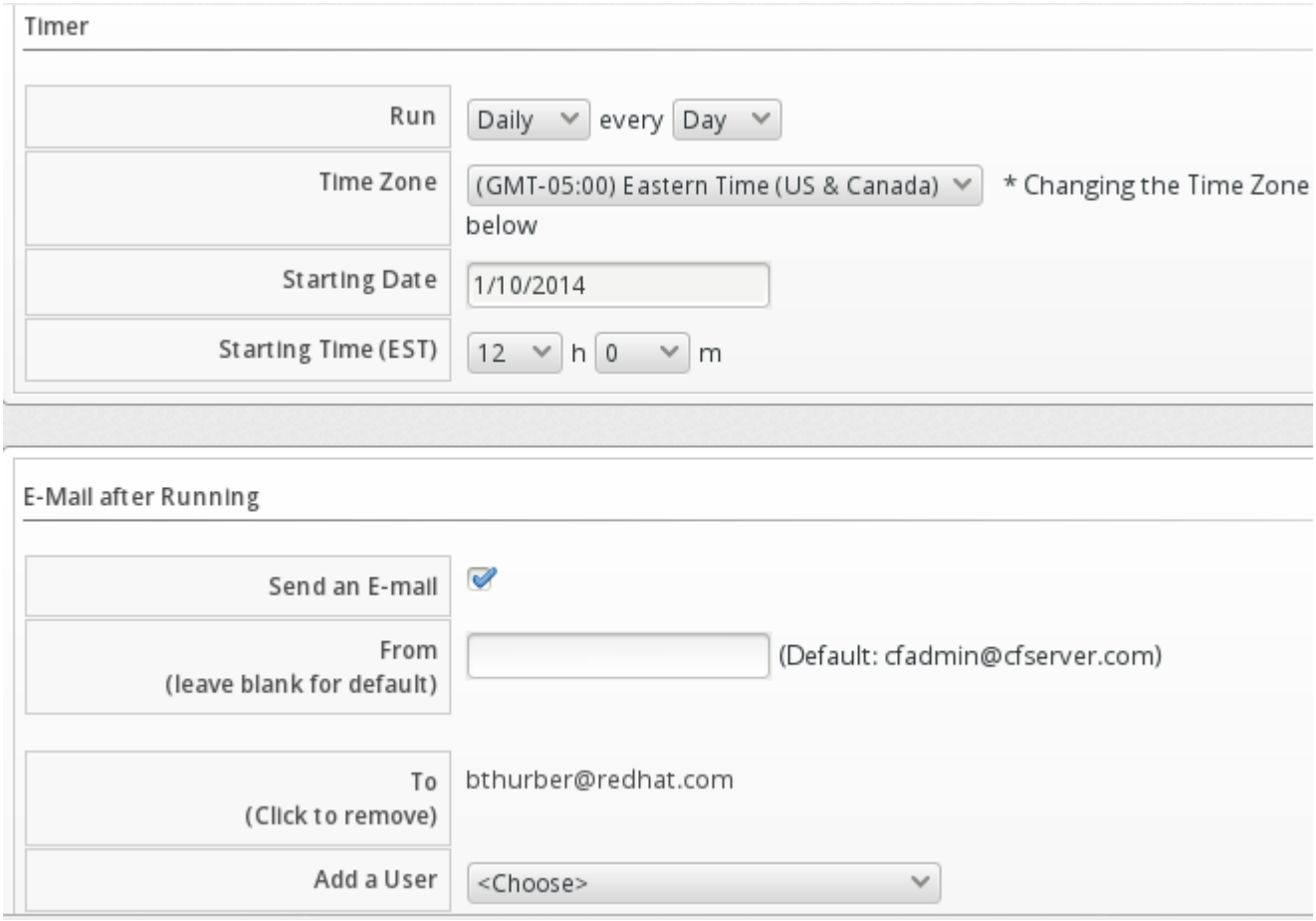


Figure 7.4.2-1: CloudForms Report Schedule

The schedule for the report is located on the accordion menu in the left window pane under **Schedules**, **All Schedules**.

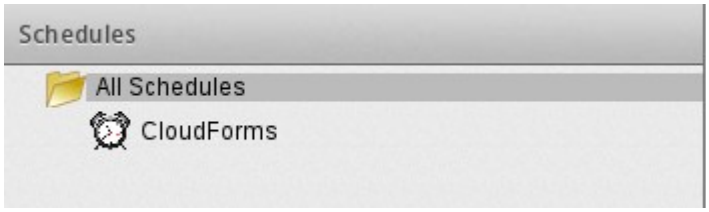


Figure 7.4.2-2: Scheduled Report

Example e-mail report generation.

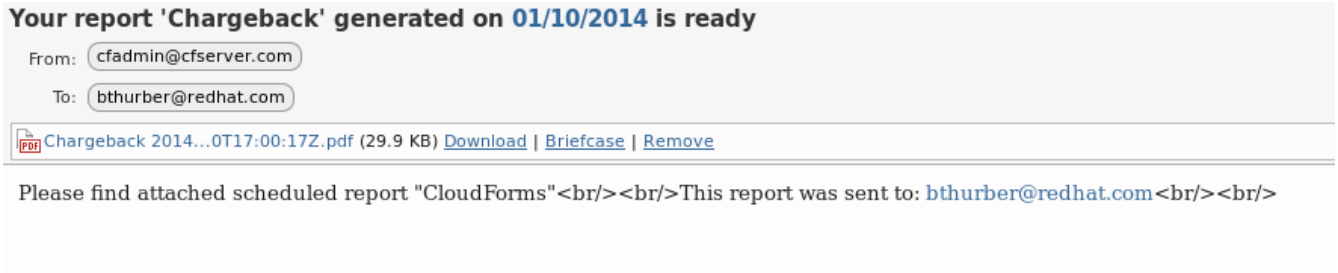


Figure 7.4.2-3: E-mail Report Generation

To manually invoke the *CloudForms* report at any time, on the accordion menu in the left window pane select **Reports** and navigate to **All Reports**, **Red Hat (All EVM Groups)**, **Custom**.

Highlight the *CloudForms* report and on the right window pane click **Queue**. Upon selection a status window is displayed on the right window pane.

Figure 7.4.2-4: CloudForms Manual Report Invocation Status

Click the circular arrow icon to refresh the generation status.

To access the report click the document with the green check mark or on the accordion menu in the left window pane, select **Saved Reports**, expand **CloudForms** under **All Saved Report** and select the report.

The completed report appears on the right window pane and can be saved to local disk as a *.txt*, *.csv*, or *.pdf* format. Optionally the report can be displayed in a full screen or deleted.

Saved Report "Chargeback - Fri Jan 10 11:35:42 EST 2014"

Date Range	VM Name	CPU Used Cost	Disk I/O Used Cost	Network I/O Used Cost	Total Cost
01/09/2014	cf-lamp	$0.00	$0.15	$0.00	$0.16
01/09/2014	cf-lamp-user-data	$0.00	$0.18	$0.00	$0.19
01/09/2014	cf-lamp015	$0.00	$0.16	$0.00	$0.16
01/09/2014	cf-lamp016	$0.00	$0.15	$0.00	$0.16
01/09/2014	cf-lamp017	$0.00	$0.16	$0.00	$0.16
01/09/2014					
Totals:		**$0.00**	**$0.80**	**$0.02**	**$0.82**
All Rows					
Totals:		**$0.00**	**$0.80**	**$0.02**	**$0.82**

Figure 7.4.2-5: CloudForms Manual Report Output

Note: At the time of this writing, CPU utilization and memory monitoring are not supported for Red Hat Enterprise Linux OpenStack Platform however, is planed in future Red Hat CloudForms releases.

8 Conclusion

Red Hat CloudForms 3.0 provides a feature rich, Infrastructure-as-a-Service (IaaS) platform giving customers a single interface needed to optimize and manage their environments. Whether needs revolve around discovery, resource costing and compliance, self-service to automation and scaling, Red Hat CloudForms 3.0 provides the tools needed for managing public and private clouds.

The goal of this reference architecture focuses on managing and integrating with Red Hat Enterprise Linux OpenStack Platform 4.0 in an on-premise IaaS cloud environment. The following use cases are demonstrated successfully:

- Importing CloudForms Management Engine appliance into RHEL OSP for discovery and management
- Creating and deploying a Linux, Apache, MySQL, and PHP (LAMP) stack from the Service Catalog as a self-service user
- Demonstrating the use of Control Policies for instance management and automation
- Demonstrating RHEL OSP metering integration with Capacity and Utilization to include chargeback for managed resources

Each use case demonstrated builds upon the previous reaching a pinnacle of functionality containing the details necessary to provide Red Hat customers with the ability to reproduce in their own environments.

Appendix A: Revision History

Revision 1.0	Friday, March 28, 2014	Brett Thurber
Initial Release		

Appendix B: Contributors

Contributor	Title	Contribution
Aaron Weitekamp	Sr. Software Engineer	Review
Dave Johnson	Supervisor, Quality Engineering	Review
Pete Savage	Senior Quality Engineer	Review
Xavier Lecauchois	Principal Product Manager - Technical	Review

Appendix C: iptables

cf-cfme-rhelosp *and* ***cf-cfme-rhev (CloudForms Management Engine)***

/etc/sysconfig/iptables

```
# 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 5432 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

cf-rhos-glance (RHEL OSP Controller)

/etc/sysconfig/iptables

```
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*mangle
:PREROUTING ACCEPT [41932:75903055]
:INPUT ACCEPT [39239:75691822]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [38652:15680606]
:POSTROUTING ACCEPT [38652:15680606]
:nova-api-POSTROUTING - [0:0]
-A POSTROUTING -j nova-api-POSTROUTING
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*nat
:PREROUTING ACCEPT [2726:213062]
:POSTROUTING ACCEPT [3736:284253]
:OUTPUT ACCEPT [3736:284253]
: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 POSTROUTING -j nova-api-POSTROUTING
-A POSTROUTING -j nova-postrouting-bottom
-A OUTPUT -j nova-api-OUTPUT
-A nova-api-snat -j nova-api-float-snat
-A nova-postrouting-bottom -j nova-api-snat
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [7479:3820195]
: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 -p tcp -m multiport --dports 8777 -m comment --comment "001
ceilometer-api incoming ALL" -j ACCEPT
-A INPUT -j nova-api-INPUT
-A INPUT -s 10.16.11.25/32 -p tcp -m multiport --dports 3260,8776 -m comment
--comment "001 cinder incoming 10.16.11.25" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 80 -m comment --comment "001 horizon
incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 9292 -m comment --comment "001 glance
incoming ALL" -j ACCEPT
-A INPUT -i lo -p tcp -m multiport --dports 27017 -m comment --comment "001
mongodb incoming localhost" -j ACCEPT
```

```
-A INPUT -p tcp -m multiport --dports 5000,35357 -m comment --comment "001
keystone incoming ALL" -j ACCEPT
-A INPUT -s 10.16.11.24/32 -p tcp -m multiport --dports 3306 -m comment
--comment "001 mysql incoming 10.16.11.24" -j ACCEPT
-A INPUT -s 10.16.11.25/32 -p tcp -m multiport --dports 3306 -m comment
--comment "001 mysql incoming 10.16.11.25" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 80 -m comment --comment "001 nagios
incoming" -j ACCEPT
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 5666 -m comment
--comment "001 nagios-nrpe incoming 10.16.11.27" -j ACCEPT
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 3306 -m comment
--comment "001 mysql incoming 10.16.11.27" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 6080 -m comment --comment "001
novncproxy incoming" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 8773,8774,8775 -m comment --comment
"001 novaapi incoming" -j ACCEPT
-A INPUT -s 10.16.11.24/32 -p tcp -m multiport --dports 5671,5672 -m comment
--comment "001 qpid incoming 10.16.11.24" -j ACCEPT
-A INPUT -s 10.16.11.25/32 -p tcp -m multiport --dports 5671,5672 -m comment
--comment "001 qpid incoming 10.16.11.25" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 8080 -m comment --comment "001 swift
proxy incoming" -j ACCEPT
-A INPUT -s 10.16.11.25/32 -p tcp -m multiport --dports 6000,6001,6002,873
-m comment --comment "001 swift storage and rsync incoming 10.16.11.25" -j
ACCEPT
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 6000,6001,6002,873
-m comment --comment "001 swift storage and rsync incoming 10.16.11.27" -j
ACCEPT
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 5671,5672 -m comment
--comment "001 qpid incoming 10.16.11.27" -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 -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j nova-filter-top
-A FORWARD -j nova-api-FORWARD
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -j nova-filter-top
-A OUTPUT -j nova-api-OUTPUT
-A nova-api-INPUT -d 10.16.11.27/32 -p tcp -m tcp --dport 8775 -j ACCEPT
-A nova-filter-top -j nova-api-local
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
```

cf-rhos-neutron (RHEL OSP Network)

/etc/sysconfig/iptables

```
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2609:482241]
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 5666 -m comment
```

```
--comment "001 nagios-nrpe incoming 10.16.11.27" -j ACCEPT
-A INPUT -s 10.16.11.24/32 -p tcp -m multiport --dports 9696,67,68 -m
comment --comment "001 neutron incoming 10.16.11.24" -j ACCEPT
-A INPUT -s 10.16.11.25/32 -p tcp -m multiport --dports 9696,67,68 -m
comment --comment "001 neutron incoming 10.16.11.25" -j ACCEPT
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 9696,67,68 -m
comment --comment "001 neutron incoming 10.16.11.27" -j ACCEPT
-A INPUT -p tcp -m multiport --dports 9696,67,68 -m comment --comment "001
neutron incoming everyone" -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 -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
```

cf-rhos-nova-1 (RHEL OSP Compute)

/etc/sysconfig/iptables

```
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*nat
:PREROUTING ACCEPT [2113:165398]
:POSTROUTING ACCEPT [211:16571]
:OUTPUT ACCEPT [211:16571]
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p tcp -j
MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p udp -j
MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j MASQUERADE
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*mangle
:PREROUTING ACCEPT [4461:5670984]
:INPUT ACCEPT [2368:5506755]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2678:1405541]
:POSTROUTING ACCEPT [2678:1405541]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM
--checksum-fill
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
# Generated by iptables-save v1.4.7 on Wed Feb 26 10:40:35 2014
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [816:193263]
-A INPUT -s 10.16.11.27/32 -p tcp -m multiport --dports 5666 -m comment
--comment "001 nagios-nrpe incoming 10.16.11.27" -j ACCEPT
-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 -s 10.16.11.27/32 -p tcp -m multiport --dports 5900:5999 -m comment
--comment "001 nova compute incoming 10.16.11.27" -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 -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
COMMIT
# Completed on Wed Feb 26 10:40:35 2014
```

cf-rhevm-32 (RHEV Manager)

/etc/sysconfig/iptables

```
# Generated by iptables-save v1.4.7 on Mon Oct 21 21:28:03 2013
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [377:134420]
-A INPUT -s 10.16.11.153/32 -p tcp -m tcp --dport 5432 -j ACCEPT
-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 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Oct 21 21:28:03 2013
```

cf-rhevh and *cf-rhevh-2 (RHEV Hypervisors)*

/etc/sysconfig/iptables

```
# oVirt automatically generated firewall configuration
*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
# libvirt tls
-A INPUT -p tcp --dport 16514 -j ACCEPT
# SSH
```

```
-A INPUT -p tcp --dport 22 -j ACCEPT
# guest consoles
-A INPUT -p tcp -m multiport --dports 5634:6166 -j ACCEPT
# migration
-A INPUT -p tcp -m multiport --dports 49152:49216 -j ACCEPT
# snmp
-A INPUT -p udp --dport 161 -j ACCEPT
#
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -m physdev ! --physdev-is-bridged -j REJECT --reject-with
icmp-host-prohibited
COMMIT
```

sysman-rhel6 (Satellite Server)

/etc/sysconfig/iptables

```
# Generated by iptables-save v1.3.5 on Tue Jan 11 09:19:06 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [4418854:1249223840]
-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 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 67 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 68 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 68 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 69 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 69 -j ACCEPT
-A INPUT -p udp -m udp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 137 -j ACCEPT
-A INPUT -p udp -m udp --dport 137 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 138 -j ACCEPT
-A INPUT -p udp -m udp --dport 138 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 139 -j ACCEPT
-A INPUT -p udp -m udp --dport 139 -j ACCEPT
-A INPUT -p udp -m udp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 4545 -j ACCEPT
-A INPUT -p udp -m udp --dport 4545 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 5222 -j ACCEPT
-A INPUT -p udp -m udp --dport 5222 -j ACCEPT
-A INPUT -p udp -m udp --dport 25150 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 25151 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 111 -j ACCEPT
-A INPUT -p udp -m udp --dport 111 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 662 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 875 -j ACCEPT
-A INPUT -p udp -m udp --dport 875 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 892 -j ACCEPT
```

```
-A INPUT -p udp -m udp --dport 892 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 2049 -j ACCEPT
-A INPUT -p udp -m udp --dport 2049 -j ACCEPT
-A INPUT -p udp -m udp --dport 32769 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 32803 -j ACCEPT
#-A INPUT -j LOG --log-prefix "---FIREWALL REJECTS----"
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Jan 11 09:19:06 2011
```

Appendix D: kickstart

lampsrv (RHEL OSP Image)

```
# Kickstart config file generated by Red Hat Satellite Config Management
# Profile Label : lampsrv
# Date Created  : 2013-09-12 13:46:16.0

install
text
network --bootproto dhcp
url --url
http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/ks-rhel-x86_64-server-6-6
.4
lang en_US
keyboard us
zerombr
clearpart --all
bootloader --location mbr
timezone America/New_York
auth --enablemd5 --enableshadow
rootpw --iscrypted [REDACTED]
selinux --enforcing
reboot
firewall --enabled
skipx
repo --name=rhel-x86_64-server-supplementary-6
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhel-x86_
64-server-supplementary-6/ks-rhel-x86_64-server-6-6.4
repo --name=rhn-tools-rhel-x86_64-server-6
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhn-tools
-rhel-x86_64-server-6/ks-rhel-x86_64-server-6-6.4
repo --name=rhel-x86_64-rhev-agent-6-server
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhel-x86_
64-rhev-agent-6-server/ks-rhel-x86_64-server-6-6.4
part /boot --fstype=ext3 --size=200
part pv.01 --size=1000 --grow
part swap --size=1000   --maxsize=2000
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --size=1000 --grow

%packages
@ Base
@ Web Server
```

```
%end

%pre

wget
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/trig/mode/pre/profil
e/lampsrv:1:RedHat" -O /dev/null



echo "Saving RHN keys..." > /dev/ttyS0
SYSTEM_ID=/etc/sysconfig/rhn/systemid
rhn_keys_found=no

mkdir -p /tmp/rhn

drives=$(list-harddrives | awk '{print $1}')
for disk in $drives; do
    DISKS="$DISKS $(fdisk -l /dev/$disk | grep -v "swap\|LVM\|Extended" |
awk '/^\/dev/{print $1}')"
done

# Try to find the keys on ordinary partitions
for disk in $DISKS; do
    name=test-$(basename $disk)
    mkdir -p /tmp/$name
    mount $disk /tmp/$name
    [ $? -eq 0 ] || continue # Skip to the next partition if the mount fails

    # Copy current RHN host keys out to be reused
    if [ -f /tmp/${name}$SYSTEM_ID ]; then
        cp -a /tmp/${name}$SYSTEM_ID /tmp/rhn
        rhn_keys_found="yes"
        umount /tmp/$name
        break
    fi
    umount /tmp/$name
    rm -r /tmp/$name
done

# Try LVM if that didn't work
if [ "$rhn_keys_found" = "no" ]; then
    lvm lvmdiskscan
    vgs=$(lvm vgs | tail -n +2 | awk '{ print $1 }')
    for vg in $vgs; do
        # Activate any VG we found
        lvm vgchange -ay $vg
    done

    lvs=$(lvm lvs | tail -n +2 | awk '{ print "/dev/" $2 "/" $1 }')
    for lv in $lvs; do
        tmpdir=$(mktemp -d findkeys.XXXXXX)
        mkdir -p /tmp/${tmpdir}
        mount $lv /tmp/${tmpdir} || continue # Skip to next volume if this
fails
```

```
        # Let's see if the keys are in there
        if [ -f /tmp/${tmpdir}$SYSTEM_ID ]; then
            cp -a /tmp/${tmpdir}$SYSTEM_ID /tmp/rhn/
            rhn_keys_found="yes"
            umount /tmp/${tmpdir}
            break # We're done!
        fi
        umount /tmp/${tmpdir}
        rm -r /tmp/${tmpdir}
    done

    # And clean up..
    for vg in $vgs; do
        lvm vgchange -an $vg
    done
fi


%end

%post --nochroot
mkdir /mnt/sysimage/tmp/ks-tree-copy
if [ -d /oldtmp/ks-tree-shadow ]; then
cp -fa /oldtmp/ks-tree-shadow/* /mnt/sysimage/tmp/ks-tree-copy
elif [ -d /tmp/ks-tree-shadow ]; then
cp -fa /tmp/ks-tree-shadow/* /mnt/sysimage/tmp/ks-tree-copy
fi
cp /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
cp -f /tmp/ks-pre.log* /mnt/sysimage/root/ || :

%end

%post --nochroot --interpreter /usr/bin/python
try:
    import xmlrpclib
    import shutil
    import sys
    import os.path
    old_system_id = "/tmp/rhn/systemid"
    new_system_id = "/mnt/sysimage/root/systemid.old"
    tmp_key = "/mnt/sysimage/tmp/key"

    new_keys = "1-efd85a2a7349e4009a20f301b180bbf5"
    for key in new_keys.split(','):
        if key.startswith('re-'):
            sys.exit(0)
    if os.path.exists(old_system_id):
        client =
xmlrpclib.Server("http://sysman-rhel6.refarch.bos.redhat.com/rpc/api")
        key =
client.system.obtain_reactivation_key(open(old_system_id).read())
        if os.path.exists(tmp_key):
            f = open(tmp_key, "r+")
            contents = f.read()
```

```
            if contents and not contents[-1] == ',':
                f.write(',')
        else:
            f = open(tmp_key, "w")
        f.write(key)
        f.close()
        shutil.copy(old_system_id, new_system_id)
except:
    # xml rpc due to  a old/bad system id
    # we don't care about those
    # we'll register those as new.
    pass


%end

%post --log /root/ks-rhn-post.log
# --Begin Red Hat Satellite command section--
cat > /tmp/ssl-key-1 <<'EOF'
                        <content removed for brevity>

EOF
# ssl-key1
cat /tmp/ssl-key-* > /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -pe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i
/etc/sysconfig/rhn/up2date

mkdir -p /tmp/rhn_rpms/optional
cd /tmp/rhn_rpms/optional
wget -P /tmp/rhn_rpms/optional
http://sysman-rhel6.refarch.bos.redhat.com/download/package/a9c316de818bd6bb
55193fcd5749a17885d72cf2/0/1/15503/rhnlib-2.5.22-15.el6.noarch.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/81968b347f818c8f
f66768c920add0debe30241a/0/1/2079/pyOpenSSL-0.10-2.el6.x86_64.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/e1c171eef6593349
3837c19406bf053979ce5e79/0/1/16893/libxml2-2.7.6-14.el6.i686.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/2c0ae71f69ef7764
1a18fdb50698e669055f0966/0/1/16895/libxml2-python-2.7.6-14.el6.x86_64.rpm
rpm -Uvh --replacepkgs --replacefiles /tmp/rhn_rpms/optional/pyOpenSSL*
/tmp/rhn_rpms/optional/rhnlib* /tmp/rhn_rpms/optional/libxml2-python*
/tmp/rhn_rpms/optional/libxml2*
perl -npe 's|^(\s*(noSSLS\|s)erverURL\s*=\s*[^:]+://)
[^/]*/|${1}sysman-rhel6.refarch.bos.redhat.com/|' -i
/etc/sysconfig/rhn/up2date

# now copy from the ks-tree we saved in the non-chroot checkout
cp -fav /tmp/ks-tree-copy/* /
rm -Rf /tmp/ks-tree-copy
# --End Red Hat Satellite command section--

# begin cobbler snippet
# set default MOTD
echo "Kickstarted on $(date +'%Y-%m-%d')" >> /etc/motd

# begin Red Hat management server registration
```

```
mkdir -p /usr/share/rhn/
wget http://sysman-rhel6.refarch.bos.redhat.com/pub/RHN-ORG-TRUSTED-SSL-CERT
-O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -npe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i
/etc/sysconfig/rhn/*
if [ -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release ]; then
    rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
fi
key=""
if [ -f /tmp/key ]; then
    key=`cat /tmp/key`
fi

if [ $key ]; then
    rhnreg_ks --serverUrl=https://sysman-rhel6.refarch.bos.redhat.com/XMLRPC
--sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
--activationkey=$key,1-efd85a2a7349e4009a20f301b180bbf5
else
     rhnreg_ks
--serverUrl=https://sysman-rhel6.refarch.bos.redhat.com/XMLRPC
--sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
--activationkey=1-efd85a2a7349e4009a20f301b180bbf5
fi
# end Red Hat management server registration

# end cobbler snippet

rhn_check

# Start post_install_network_config generated code
# End post_install_network_config generated code

%end


%post
#Please edit this script on sysman-rhel6 under /scripts
wget -O -
http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/refarch-lamp.sh |
/bin/bash
%end

%post


# Start koan environment setup
echo "export COBBLER_SERVER=sysman-rhel6.refarch.bos.redhat.com" >
/etc/profile.d/cobbler.sh
echo "setenv COBBLER_SERVER sysman-rhel6.refarch.bos.redhat.com" >
/etc/profile.d/cobbler.csh
# End koan environment setup



wget
```

```
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/ks/profile/lampsrv:1
:RedHat" -O /root/cobbler.ks
wget
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/trig/mode/post/profi
le/lampsrv:1:RedHat" -O /dev/null
%end
```

openstack (RHEL OSP Nodes)

```
# Kickstart config file generated by Red Hat Satellite Config Management
# Profile Label : openstack
# Date Created  : 2013-07-09 14:40:19.0

install
text
network --bootproto dhcp
url --url
http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/ks-rhel-x86_64-server-6-6
.5
lang en_US
keyboard us
zerombr
clearpart --all
bootloader --location mbr
timezone America/New_York
auth --enablemd5 --enableshadow
rootpw --iscrypted [REDACTED]
selinux --enforcing
reboot
firewall --enabled
skipx
repo --name=rhn-tools-rhel-x86_64-server-6
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhn-tools
-rhel-x86_64-server-6/ks-rhel-x86_64-server-6-6.5
repo --name=rhel-x86_64-server-rh-common-6
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhel-x86_
64-server-rh-common-6/ks-rhel-x86_64-server-6-6.5
repo --name=rhel-x86_64-server-6-ost-4
--baseurl=http://sysman-rhel6.refarch.bos.redhat.com/ks/dist/child/rhel-x86_
64-server-6-ost-4/ks-rhel-x86_64-server-6-6.5
part /boot --fstype=ext3 --size=200
part pv.01 --size=1000 --grow
part swap --size=1000   --maxsize=2000
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --size=1000 --grow

%packages
@ Base
%end

%pre

wget
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/trig/mode/pre/profil
```

```
e/openstack:1:RedHat" -O /dev/null



echo "Saving RHN keys..." > /dev/ttyS0
SYSTEM_ID=/etc/sysconfig/rhn/systemid
rhn_keys_found=no

mkdir -p /tmp/rhn

drives=$(list-harddrives | awk '{print $1}')
for disk in $drives; do
    DISKS="$DISKS $(fdisk -l /dev/$disk | grep -v "swap\|LVM\|Extended" |
awk '/^\/dev/{print $1}')"
done

# Try to find the keys on ordinary partitions
for disk in $DISKS; do
    name=test-$(basename $disk)
    mkdir -p /tmp/$name
    mount $disk /tmp/$name
    [ $? -eq 0 ] || continue # Skip to the next partition if the mount fails

    # Copy current RHN host keys out to be reused
    if [ -f /tmp/${name}$SYSTEM_ID ]; then
        cp -a /tmp/${name}$SYSTEM_ID /tmp/rhn
        rhn_keys_found="yes"
        umount /tmp/$name
        break
    fi
    umount /tmp/$name
    rm -r /tmp/$name
done

# Try LVM if that didn't work
if [ "$rhn_keys_found" = "no" ]; then
    lvm lvmdiskscan
    vgs=$(lvm vgs | tail -n +2 | awk '{ print $1 }')
    for vg in $vgs; do
        # Activate any VG we found
        lvm vgchange -ay $vg
    done

    lvs=$(lvm lvs | tail -n +2 | awk '{ print "/dev/" $2 "/" $1 }')
    for lv in $lvs; do
        tmpdir=$(mktemp -d findkeys.XXXXXX)
        mkdir -p /tmp/${tmpdir}
        mount $lv /tmp/${tmpdir} || continue # Skip to next volume if this
fails

        # Let's see if the keys are in there
        if [ -f /tmp/${tmpdir}$SYSTEM_ID ]; then
            cp -a /tmp/${tmpdir}$SYSTEM_ID /tmp/rhn/
            rhn_keys_found="yes"
            umount /tmp/${tmpdir}
```

```
            break # We're done!
        fi
        umount /tmp/${tmpdir}
        rm -r /tmp/${tmpdir}
    done

    # And clean up..
    for vg in $vgs; do
        lvm vgchange -an $vg
    done
fi


%end

%post --nochroot
mkdir /mnt/sysimage/tmp/ks-tree-copy
if [ -d /oldtmp/ks-tree-shadow ]; then
cp -fa /oldtmp/ks-tree-shadow/* /mnt/sysimage/tmp/ks-tree-copy
elif [ -d /tmp/ks-tree-shadow ]; then
cp -fa /tmp/ks-tree-shadow/* /mnt/sysimage/tmp/ks-tree-copy
fi
cp /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
cp -f /tmp/ks-pre.log* /mnt/sysimage/root/ || :

cp `awk '{ if ($1 ~ /%include/) {print $2}}' /tmp/ks.cfg` /tmp/ks.cfg
/mnt/sysimage/root
%end

%post --nochroot --interpreter /usr/bin/python
try:
    import xmlrpclib
    import shutil
    import sys
    import os.path
    old_system_id = "/tmp/rhn/systemid"
    new_system_id = "/mnt/sysimage/root/systemid.old"
    tmp_key = "/mnt/sysimage/tmp/key"

    new_keys = "1-eabcc0138ac099eb31aa503d575a7092"
    for key in new_keys.split(','):
        if key.startswith('re-'):
            sys.exit(0)
    if os.path.exists(old_system_id):
        client =
xmlrpclib.Server("http://sysman-rhel6.refarch.bos.redhat.com/rpc/api")
        key =
client.system.obtain_reactivation_key(open(old_system_id).read())
        if os.path.exists(tmp_key):
            f = open(tmp_key, "r+")
            contents = f.read()
            if contents and not contents[-1] == ',':
                f.write(',')
        else:
            f = open(tmp_key, "w")
```

```
        f.write(key)
        f.close()
        shutil.copy(old_system_id, new_system_id)
except:
    # xml rpc due to  a old/bad system id
    # we don't care about those
    # we'll register those as new.
    pass


%end

%post --log /root/ks-rhn-post.log
# --Begin Red Hat Satellite command section--
cat > /tmp/ssl-key-1 <<'EOF'
                        <content removed for brevity>

EOF
# ssl-key1
cat /tmp/ssl-key-* > /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -pe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i
/etc/sysconfig/rhn/up2date

mkdir -p /tmp/rhn_rpms/optional
cd /tmp/rhn_rpms/optional
wget -P /tmp/rhn_rpms/optional
http://sysman-rhel6.refarch.bos.redhat.com/download/package/a9c316de818bd6bb
55193fcd5749a17885d72cf2/0/1/15503/rhnlib-2.5.22-15.el6.noarch.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/2c0ae71f69ef7764
1a18fdb50698e669055f0966/0/1/16895/libxml2-python-2.7.6-14.el6.x86_64.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/9b0362f4ec08abc1
fbe985ccfd551d6cd47744bf/0/1/18861/pyOpenSSL-0.13.1-1.el6ost.x86_64.rpm
http://sysman-rhel6.refarch.bos.redhat.com/download/package/972717cba3b8713b
6af0cc874d789ece6a2bbee8/0/1/16901/libxml2-2.7.6-14.el6.x86_64.rpm
rpm -Uvh --replacepkgs --replacefiles /tmp/rhn_rpms/optional/pyOpenSSL*
/tmp/rhn_rpms/optional/rhnlib* /tmp/rhn_rpms/optional/libxml2-python*
/tmp/rhn_rpms/optional/libxml2*
perl -npe 's|^(\s*(noSSLS\|s)erverURL\s*=\s*[^:]+://)
[^/]*/|${1}sysman-rhel6.refarch.bos.redhat.com/|' -i
/etc/sysconfig/rhn/up2date

# now copy from the ks-tree we saved in the non-chroot checkout
cp -fav /tmp/ks-tree-copy/* /
rm -Rf /tmp/ks-tree-copy
# --End Red Hat Satellite command section--

# begin cobbler snippet
# set default MOTD
echo "Kickstarted on $(date +'%Y-%m-%d')" >> /etc/motd

# begin Red Hat management server registration
mkdir -p /usr/share/rhn/
wget http://sysman-rhel6.refarch.bos.redhat.com/pub/RHN-ORG-TRUSTED-SSL-CERT
-O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -npe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i
```

```
/etc/sysconfig/rhn/*
if [ -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release ]; then
    rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
fi
key=""
if [ -f /tmp/key ]; then
    key=`cat /tmp/key`
fi

if [ $key ]; then
    rhnreg_ks --serverUrl=https://sysman-rhel6.refarch.bos.redhat.com/XMLRPC
--sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
--activationkey=$key,1-eabcc0138ac099eb31aa503d575a7092
else
     rhnreg_ks
--serverUrl=https://sysman-rhel6.refarch.bos.redhat.com/XMLRPC
--sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
--activationkey=1-eabcc0138ac099eb31aa503d575a7092
fi
# end Red Hat management server registration

# end cobbler snippet

rhn_check

# Start post_install_network_config generated code
# End post_install_network_config generated code

%end


%post --log /root/ks-post.log.1
#Please edit this script on sysman-rhel6 under /scripts
wget -O -
http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/refarch-rhos-post
.sh | /bin/bash
%end

%post


# Start koan environment setup
echo "export COBBLER_SERVER=sysman-rhel6.refarch.bos.redhat.com" >
/etc/profile.d/cobbler.sh
echo "setenv COBBLER_SERVER sysman-rhel6.refarch.bos.redhat.com" >
/etc/profile.d/cobbler.csh
# End koan environment setup



wget
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/ks/profile/openstack
:1:RedHat" -O /root/cobbler.ks
wget
"http://sysman-rhel6.refarch.bos.redhat.com/cblr/svc/op/trig/mode/post/profi
```

```
le/openstack:1:RedHat" -O /dev/null
%end
```

D.1 Post Install Scripts

refarch-common-post.sh

```
#!/bin/bash
#
# Call this script from a kickstart post, for example in Satellite
# Just add this one line as your script:
# wget -O -
http://sysman-thel6.refarch.bos.redhat.com/cobbler/scripts/refarch-common-po
st.sh | /bin/bash

LOGFILE=/root/refarch-common-post-`hostname -s`-`date +%F_%T`.log
(

# get redeploy for future use
if [ ! -d /usr/local/bin ]
then
  echo making /usr/local/bin
  mkdir -p /usr/local/bin
fi
wget http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/redeploy
-O /usr/local/bin/redeploy
chmod +x /usr/local/bin/redeploy

# Adding group bashrc
wget http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/bashrc -O
/etc/profile.d/refarch-bashrc.sh

# Fix SELinux issues with the /root/.ssh directory and files.
restorecon -r /root/.ssh

# Importing gpg key for custom RHN channel/packages
wget
http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/public_key.txt -O
/root/public_key.txt
rpm --import /root/public_key.txt

# Adding packages that should always be installed
# echo -Adding vital packages
yum -y install screen firefox nfs-utils xorg-x11-xauth autofs sg3_utils
policycoreutils-python rhevm-guest-agent web-1-1
yum -y groupinstall @network-file-system-client
chkconfig httpd on
# full system update
echo -Running full update ...
yum -y update

) 2>&1 | tee ${LOGFILE}
echo Done with [refarch-common-post]. Wrote logfile: ${LOGFILE}
```

refarch-rhos-post.sh

```
#!/bin/bash
#
# Call this script from a kickstart post, for example in Satellite
# Just add this one line as your script:
# wget -O -
http://sysman-thel6.refarch.bos.redhat.com/cobbler/scripts/refarch-common-po
st.sh | /bin/bash

LOGFILE=/root/refarch-rhos-post-`hostname -s`-`date +%F_%T`.log
(

# get redeploy for future use
if [ ! -d /usr/local/bin ]
then
  echo making /usr/local/bin
  mkdir -p /usr/local/bin
fi
wget http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/redeploy
-O /usr/local/bin/redeploy
chmod +x /usr/local/bin/redeploy

# Adding group bashrc
wget http://sysman-rhel6.refarch.bos.redhat.com/cobbler/scripts/bashrc -O
/etc/profile.d/refarch-bashrc.sh

# Fix SELinux issues with the /root/.ssh directory and files.
restorecon -r /root/.ssh

# Adding packages that should always be installed
# echo -Adding vital packages
yum -y groupinstall @network-file-system-client nfs-utils
iscsi-initiator-utils

rm -rf /etc/modprobe.d/anaconda.conf

# full system update
echo -Running full update ...
yum -y update


) 2>&1 | tee ${LOGFILE}
echo Done with [refarch-rhos-post]. Wrote logfile: ${LOGFILE}
```

Appendix E: CloudForms Custom Methods and Scripts

lampsrv.rb

```
#
#            Automate Method
#
$evm.log("info", "lampsrv Automate Method Started")
@method = 'buildrequest'
@log_prefix = "[#{@method}]"
@debug = true
#
#            Method Code Goes here
#

def build_request(solution_hash)

output = ''

prov = $evm.root["service_template_provision_task"]

#Set some things
  # Get the current logged in user
  user = $evm.root['user']
  #$evm.log("info","#{@method} - Inspecting User object:<#{user.inspect}>")
if @debug

  if user.nil?
    userid = 'admin'
    user_mail = 'bthurber@redhat.com'
    # Should get rid of these 2 below in the dialog
    #user_first = 'Admin'
    #user_last = 'Administrator'
  else
    userid = user.userid
    user_mail = user.email

    # If currently logged in user email is nil assign a default email
address
    user_mail ||= 'bthurber@redhat.com'
    user_first = "Brett"
    user_last = "Thurber"
  end

$evm.log("info","Output of Root Object")
$evm.root.attributes.sort.each { |k, v| $evm.log("info", "\t#{k}: #{v}")}

lampsrv = true

if lampsrv == true
    $evm.log("info","lampsrv is a go")
```

```
    # arg1 = version
    args = ['1.1']
    # arg2 = templateFields
    args << "name=#{solution_hash[:template_name]}|request_type=template"
    # arg3 = vmFields
    args <<
"vm_name=#{solution_hash[:vm_name]}|instance_type=#{solution_hash[:instance_
type]}|customization_template_id=#{solution_hash[:customization_template_id]
}|guest_access_key_pair=#{solution_hash[:guest_access_key_pair]}"
    # arg4 = requester
    args <<
"owner_email=#{solution_hash[:email]}|owner_last_name=#{solution_hash[:last_
name]}|owner_first_name=#{solution_hash[:first_name]}"
    # arg5 = tags
    args << "environment=cloudforms"
    # arg6 = additionalValues
    args << "cloud_network=#{solution_hash[:cloud_network]}"
    # arg7 = emsCustomAttributes
    args << solution_hash.collect { |k, v| "#{k}=#{v}" }.join('|')
    # arg8 = miqCustomAttributes
    args <<
"security_groups=#{solution_hash[:security_groups]}|availability_zones=#{sol
ution_hash[:availability_zones]}"
    $evm.log("info", "Inline Method: <#{@log_prefix}> - Building
provisioning request with the following arguments: <#{args.inspect}>")
    # exit MIQ_ABORT
    $evm.execute('create_provision_request', *args)
else
     $evm.log("info","lampsrv is foo bar, not provisioning")
end

end

def parse_piped_string(text_input, options={})
  return {} unless text_input.kind_of?(String)
  result = {}
  text_input.split('|').each do |value|

    next if value.blank?
    idx = value.index('=')
    next if idx.nil?
    key = !options[:modify_key_name] ? value[0, idx].strip : value[0,
idx].strip.to_sym
    result[key] = value[idx+1..-1].strip
  end
  return result
end

## Get variables
solution_id = $evm.root['dialog_solution_id'].to_i
$evm.root['solution_id'] = solution_id
$evm.log("info", "Inline Method: <#{@log_prefix}> - Solution_ID :
#{solution_id}") if @debug
```

```
  solution_hash= {
      :template_name => "cf-lamp",
      :first_name => "Brett",
      :last_name => "Thurber",
      :email => "bthurber@redhat.com",
      :vm_name => "cf-lamp$n{3}",
      :instance_type => "2",
      :security_groups => "1",
      :cloud_network => "2",
      :availability_zones => "1",
      :customization_template_id => "7",
      :guest_access_key_pair => "2"
  }
  $evm.log("info", "Building request") if @debug

  build_request(solution_hash)

$evm.log("info", "<Inline Method: <#{@log_prefix}> - EVM Service Task
Finished")  if @debug

#
#
#
$evm.log("info", "lampsrv Automate Method Ended")
exit MIQ_OK
```

PreProvision.rb

```
###################################
#
# EVM Automate Method: PreProvision
#
# Notes: This default method is used to apply PreProvision customizations
for VMware, RHEV and Amazon provisioning
#
###################################
# Method for logging
def log(level, message)
  @method = 'PreProvision'
  $evm.log(level, "#{@method} - #{message}")
end

begin
  log(:info, "EVM Automate Method Started")

  ###################################
  #
  # Method: process_vmware
  # Notes: Process vmware specific provisioning options
  #
  ###################################
  def process_vmware(prov)
    # Choose the sections to process
    set_vlan = false
    set_folder = false
```

```
    set_resource_pool = false
    set_notes = true

    # Get information from the template platform
    template = prov.vm_template
    product  = template.operating_system['product_name'].downcase
    bitness = template.operating_system['bitness']
    log(:info,"Template:<#{template.name}> Vendor:<#{template.vendor}>
Product:<#{product}> Bitness:<#{bitness}>")

    if set_vlan
      ###################################
      # Was a VLAN selected in dialog?
      # If not you can set one here.
      ###################################
      default_vlan = "vlan1"
      default_dvs = "portgroup1"

      if prov.get_option(:vlan).nil?
        log(:info, "Provisioning object <:vlan> updated with
<#{default_vlan}>")
        prov.set_vlan(default_vlan)
        #prov.set_dvs(default_dvs)
      end
    end

    if set_folder
      ###################################
      # Drop the VM in the targeted folder if no folder was chosen in the
dialog
      # The vCenter folder must exist for the VM to be placed correctly else
the
      # VM will placed along with the template
      # Folder starts at the Data Center level
      ###################################
      default_folder = 'DC1/Infrastructure/ManageIQ/SelfService'

      if prov.get_option(:placement_folder_name).nil?
        prov.get_folder_paths.each { |key, path| $evm.log("info",
"#{@method} - Eligible folders:<#{key}> - <#{path}>") }
        prov.set_folder(default_folder)
        log(:info, "Provisioning object <:placement_folder_name> updated
with <#{default_folder}>")
      else
        log(:info, "Placing VM in folder:
<#{prov.get_option(:placement_folder_name)}>")
      end
    end

    if set_resource_pool
      if prov.get_option(:placement_rp_name).nil?
        ###########################################
        # Find and set the Resource Pool for a VM:
        ###########################################
        default_resource_pool = 'MyResPool'
```

```
        respool = prov.eligible_resource_pools.detect {|c|
c.name.casecmp(default_resource_pool)==0}
        log(:info, "Provisioning object <:placement_rp_name> updated with
<#{respool.name}>")
        prov.set_resource_pool(respool)
      end
    end

    if set_notes
      ###################################
      # Set the VM Description and VM Annotations  as follows:
      # The example would allow user input in provisioning dialog
"vm_description"
      # to be added to the VM notes
      ###################################
      # Stamp VM with custom description
      unless prov.get_option(:vm_description).nil?
        vmdescription = prov.get_option(:vm_description)
        prov.set_option(:vm_description,vmdescription)
        log(:info,"Provisioning object <:vmdescription> updated with
<#{vmdescription}>")
      end

      # Setup VM Annotations
      vm_notes =  "Owner: #{prov.get_option(:owner_first_name)}
#{prov.get_option(:owner_last_name)}"
      vm_notes += "\nEmail: #{prov.get_option(:owner_email)}"
      vm_notes += "\nSource Template: #{template.name}"
      vm_notes += "\nCustom Description: #{vmdescription}" unless
vmdescription.nil?
      prov.set_vm_notes(vm_notes)
      log(:info,"Provisioning object <:vm_notes> updated with
<#{vm_notes}>")
    end

      # Log all of the provisioning options to the automation.log
      prov.options.each { |k,v| log(:info,"Provisioning Option
Key:<#{k.inspect}> Value:<#{v.inspect}>") }
  end


  ###################################
  #
  # Method: process_redhat
  # Notes: Process redhat specific provisioning options
  #
  ###################################
  def process_redhat(prov )
    # Choose the sections to process
    set_vlan = true
    set_notes = false

    # Get information from the template platform
    template = prov.vm_template
    product  = template.operating_system['product_name'].downcase
```

```
    log(:info,"Template:<#{template.name}> Vendor:<#{template.vendor}>
Product:<#{product}>")


    if set_vlan
      # Set default VLAN here if one was not chosen in the dialog?
      default_vlan = "rhevm"

      if prov.get_option(:vlan).nil?
        prov.set_vlan(default_vlan)
        log(:info, "Provisioning object <:vlan> updated with
<#{default_vlan}>")
      end
    end

    if set_notes
      ###################################
      # Set the VM Description and VM Annotations  as follows:
      # The example would allow user input in provisioning dialog
"vm_description"
      # to be added to the VM notes
      ###################################
      # Stamp VM with custom description
      unless prov.get_option(:vm_description).nil?
        vmdescription = prov.get_option(:vm_description)
        prov.set_option(:vm_description,vmdescription)
        log(:info,"Provisioning object <:vmdescription> updated with
<#{vmdescription}>")
      end

      # Setup VM Annotations
      vm_notes =  "Owner: #{prov.get_option(:owner_first_name)}
#{prov.get_option(:owner_last_name)}"
      vm_notes += "\nEmail: #{prov.get_option(:owner_email)}"
      vm_notes += "\nSource Template: #{template.name}"
      vm_notes += "\nCustom Description: #{vmdescription}" unless
vmdescription.nil?
      prov.set_vm_notes(vm_notes)
      log(:info,"Provisioning object <:vm_notes> updated with
<#{vm_notes}>")
    end

      # Log all of the provisioning options to the automation.log
      prov.options.each { |k,v| log(:info,"Provisioning Option
Key:<#{k.inspect}> Value:<#{v.inspect}>") }
  end


  ###################################
  #
  # Method: process_amazon
  # Notes: Process Amazon specific provisioning options
  #
  ###################################
  def process_amazon(prov )
```

```
  end # end process_amazon


  ###################################
  #
  # Method: process_openstack
  # Notes: Process OpenStack specific provisioning options
  #
  ###################################
  def process_openstack(prov )
    cloud_network = prov.get_option(:ws_values)[:cloud_network]
    cloud_network = cloud_network.to_i
    log(:info, "Cloud network [#{cloud_network.inspect}]")
    prov.set_network_adapter(0, {:network_id => cloud_network})
  end # end process_openstack


  # Get provisioning object
  prov = $evm.root["miq_provision"]
  log(:info, "Provision:<#{prov.id}>
Request:<#{prov.miq_provision_request.id}> Type:<#{prov.type}>")

  # Build case statement to determine which type of processing is required
  case prov.type
  when 'MiqProvisionRedhatViaIso', 'MiqProvisionRedhatViaPxe';
process_redhat(prov)
  when 'MiqProvisionVmware';
process_vmware(prov)
  when 'MiqProvisionAmazon';
process_amazon(prov)
  when 'MiqProvisionOpenstack';
process_openstack(prov)
  else                                                          log(:info,
"Provision Type:<#{prov.type}> does not match, skipping processing")
  end

  #
  # Exit method
  #
  log(:info, "EVM Automate Method Ended")
  exit MIQ_OK

  #
  # Set Ruby rescue behavior
  #
rescue => err
  log(:error, "[#{err}]\n#{err.backtrace.join("\n")}")
  exit MIQ_ABORT
end
```

cloudinit.sh

```
#!/bin/bash
#Configure LAMP
```

```
HOSTNAME=`hostname -I | sed 's/ *$//g'`
#hostname
echo $HOSTNAME $HOSTNAME >> /etc/hosts
sed -i "s/HOSTNAME=cf-lamp.refarch.bos.redhat.com/HOSTNAME=$HOSTNAME/"
/etc/sysconfig/network
#Apache
sed -i "s/ServerName cf-lamp.refarch.bos.redhat.com:80/ServerName
$HOSTNAME:80/" /etc/httpd/conf/httpd.conf
#MYSQL
mysql --user=root --password=[REDACTED] -e "grant all privileges on *.* to
web@$HOSTNAME identified by 'web';"
#PHP
sed -i "s/mysql.default_host =
cf-lamp.refarch.bos.redhat.com/mysql.default_host =
$HOSTNAME/;s/mysqli.default_host =/mysqli.default_host = $HOSTNAME/"
/etc/php.ini
```

Appendix F: RHEL OSP Files

packstack answer file:

```
[general]

# Path to a Public key to install on servers. If a usable key has not
# been installed on the remote servers the user will be prompted for a
# password and this key will be installed so the password will not be
# required again
CONFIG_SSH_KEY=/root/.ssh/id_rsa.pub

# Set to 'y' if you would like Packstack to install MySQL
CONFIG_MYSQL_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack Image
# Service (Glance)
CONFIG_GLANCE_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack Block
# Storage (Cinder)
CONFIG_CINDER_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack Compute
# (Nova)
CONFIG_NOVA_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack
# Networking (Neutron)
CONFIG_NEUTRON_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack
# Dashboard (Horizon)
CONFIG_HORIZON_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack Object
# Storage (Swift)
CONFIG_SWIFT_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack
# Metering (Ceilometer)
CONFIG_CEILOMETER_INSTALL=y

# Set to 'y' if you would like Packstack to install OpenStack
# Orchestration (Heat)
CONFIG_HEAT_INSTALL=y

# Set to 'y' if you would like Packstack to install the OpenStack
# Client packages. An admin "rc" file will also be installed
CONFIG_CLIENT_INSTALL=y

# Comma separated list of NTP servers. Leave plain if Packstack
# should not install ntpd on instances.
CONFIG_NTP_SERVERS=10.16.255.1,10.16.255.2
```

```
# Set to 'y' if you would like Packstack to install Nagios to monitor
# OpenStack hosts
CONFIG_NAGIOS_INSTALL=y

# Comma separated list of servers to be excluded from installation in
# case you are running Packstack the second time with the same answer
# file and don't want Packstack to touch these servers. Leave plain if
# you don't need to exclude any server.
EXCLUDE_SERVERS=

# The IP address of the server on which to install MySQL
CONFIG_MYSQL_HOST=10.16.11.27

# Username for the MySQL admin user
CONFIG_MYSQL_USER=root

# Password for the MySQL admin user
CONFIG_MYSQL_PW=[REDACTED]

# The IP address of the server on which to install the QPID service
CONFIG_QPID_HOST=10.16.11.27

# Enable SSL for the QPID service
CONFIG_QPID_ENABLE_SSL=n

# Enable Authentication for the QPID service
CONFIG_QPID_ENABLE_AUTH=n

# The password for the NSS certificate database of the QPID service
CONFIG_QPID_NSS_CERTDB_PW=[REDACTED]

# The port in which the QPID service listens to SSL connections
CONFIG_QPID_SSL_PORT=5671

# The filename of the certificate that the QPID service is going to
# use
CONFIG_QPID_SSL_CERT_FILE=/etc/pki/tls/certs/qpid_selfcert.pem

# The filename of the private key that the QPID service is going to
# use
CONFIG_QPID_SSL_KEY_FILE=/etc/pki/tls/private/qpid_selfkey.pem

# Auto Generates self signed SSL certificate and key
CONFIG_QPID_SSL_SELF_SIGNED=y

# User for qpid authentication
CONFIG_QPID_AUTH_USER=qpid_user

# Password for user authentication
CONFIG_QPID_AUTH_PASSWORD=[REDACTED]

# The IP address of the server on which to install Keystone
CONFIG_KEYSTONE_HOST=10.16.11.27
```

```
# The password to use for the Keystone to access DB
CONFIG_KEYSTONE_DB_PW=[REDACTED]

# The token to use for the Keystone service api
CONFIG_KEYSTONE_ADMIN_TOKEN=3f10af933ecd4987bfe4623dc87083be

# The password to use for the Keystone admin user
CONFIG_KEYSTONE_ADMIN_PW=[REDACTED]

# The password to use for the Keystone demo user
CONFIG_KEYSTONE_DEMO_PW=[REDACTED]

# Kestone token format. Use either UUID or PKI
CONFIG_KEYSTONE_TOKEN_FORMAT=PKI

# The IP address of the server on which to install Glance
CONFIG_GLANCE_HOST=10.16.11.27

# The password to use for the Glance to access DB
CONFIG_GLANCE_DB_PW=[REDACTED]

# The password to use for the Glance to authenticate with Keystone
CONFIG_GLANCE_KS_PW=[REDACTED]

# The IP address of the server on which to install Cinder
CONFIG_CINDER_HOST=10.16.11.27

# The password to use for the Cinder to access DB
CONFIG_CINDER_DB_PW=[REDACTED]

# The password to use for the Cinder to authenticate with Keystone
CONFIG_CINDER_KS_PW=[REDACTED]

# The Cinder backend to use, valid options are: lvm, gluster, nfs
CONFIG_CINDER_BACKEND=lvm

# Create Cinder's volumes group. This should only be done for testing
# on a proof-of-concept installation of Cinder.  This will create a
# file-backed volume group and is not suitable for production usage.
CONFIG_CINDER_VOLUMES_CREATE=n

# Cinder's volumes group size. Note that actual volume size will be
# extended with 3% more space for VG metadata.
CONFIG_CINDER_VOLUMES_SIZE=450G

# A single or comma separated list of gluster volume shares to mount,
# eg: ip-address:/vol-name, domain:/vol-name
CONFIG_CINDER_GLUSTER_MOUNTS=

# A single or comma seprated list of NFS exports to mount, eg: ip-
# address:/export-name
CONFIG_CINDER_NFS_MOUNTS=

# The IP address of the server on which to install the Nova API
# service
```

```
CONFIG_NOVA_API_HOST=10.16.11.27

# The IP address of the server on which to install the Nova Cert
# service
CONFIG_NOVA_CERT_HOST=10.16.11.27

# The IP address of the server on which to install the Nova VNC proxy
CONFIG_NOVA_VNCPROXY_HOST=10.16.11.27

# A comma separated list of IP addresses on which to install the Nova
# Compute services
CONFIG_NOVA_COMPUTE_HOSTS=10.16.11.25

# The IP address of the server on which to install the Nova Conductor
# service
CONFIG_NOVA_CONDUCTOR_HOST=10.16.11.27

# The password to use for the Nova to access DB
CONFIG_NOVA_DB_PW=[REDACTED]

# The password to use for the Nova to authenticate with Keystone
CONFIG_NOVA_KS_PW=[REDACTED]

# The IP address of the server on which to install the Nova Scheduler
# service
CONFIG_NOVA_SCHED_HOST=10.16.11.27

# The overcommitment ratio for virtual to physical CPUs. Set to 1.0
# to disable CPU overcommitment
CONFIG_NOVA_SCHED_CPU_ALLOC_RATIO=16.0

# The overcommitment ratio for virtual to physical RAM. Set to 1.0 to
# disable RAM overcommitment
CONFIG_NOVA_SCHED_RAM_ALLOC_RATIO=1.5

# Private interface for Flat DHCP on the Nova compute servers
CONFIG_NOVA_COMPUTE_PRIVIF=eth3

# The list of IP addresses of the server on which to install the Nova
# Network service
CONFIG_NOVA_NETWORK_HOSTS=10.16.11.27

# Nova network manager
CONFIG_NOVA_NETWORK_MANAGER=nova.network.manager.FlatDHCPManager

# Public interface on the Nova network server
CONFIG_NOVA_NETWORK_PUBIF=eth1

# Private interface for network manager on the Nova network server
CONFIG_NOVA_NETWORK_PRIVIF=eth3

# IP Range for network manager
CONFIG_NOVA_NETWORK_FIXEDRANGE=192.168.12.0/24

# IP Range for Floating IP's
```

```
CONFIG_NOVA_NETWORK_FLOATRANGE=10.16.12.0/23

# Name of the default floating pool to which the specified floating
# ranges are added to
CONFIG_NOVA_NETWORK_DEFAULTFLOATINGPOOL=nova

# Automatically assign a floating IP to new instances
#CONFIG_NOVA_NETWORK_AUTOASSIGNFLOATINGIP=n

# First VLAN for private networks
CONFIG_NOVA_NETWORK_VLAN_START=100

# Number of networks to support
CONFIG_NOVA_NETWORK_NUMBER=1

# Number of addresses in each private subnet
CONFIG_NOVA_NETWORK_SIZE=255

# The IP addresses of the server on which to install the Neutron
# server
CONFIG_NEUTRON_SERVER_HOST=10.16.11.24

# The password to use for Neutron to authenticate with Keystone
CONFIG_NEUTRON_KS_PW=[REDACTED]

# The password to use for Neutron to access DB
CONFIG_NEUTRON_DB_PW=[REDACTED]

# A comma separated list of IP addresses on which to install Neutron
# L3 agent
CONFIG_NEUTRON_L3_HOSTS=10.16.11.24

# The name of the bridge that the Neutron L3 agent will use for
# external traffic, or 'provider' if using provider networks
CONFIG_NEUTRON_L3_EXT_BRIDGE=provider

# A comma separated list of IP addresses on which to install Neutron
# DHCP agent
CONFIG_NEUTRON_DHCP_HOSTS=10.16.11.24

# A comma separated list of IP addresses on which to install Neutron
# LBaaS agent
CONFIG_NEUTRON_LBAAS_HOSTS=

# The name of the L2 plugin to be used with Neutron
CONFIG_NEUTRON_L2_PLUGIN=openvswitch

# A comma separated list of IP addresses on which to install Neutron
# metadata agent
CONFIG_NEUTRON_METADATA_HOSTS=10.16.11.24

# A comma separated list of IP addresses on which to install Neutron
# metadata agent
CONFIG_NEUTRON_METADATA_PW=[REDACTED]
```

```
# The type of network to allocate for tenant networks (eg. vlan,
# local, gre)
CONFIG_NEUTRON_LB_TENANT_NETWORK_TYPE=local

# A comma separated list of VLAN ranges for the Neutron linuxbridge
# plugin (eg. physnet1:1:4094,physnet2,physnet3:3000:3999)
CONFIG_NEUTRON_LB_VLAN_RANGES=

# A comma separated list of interface mappings for the Neutron
# linuxbridge plugin (eg. physnet1:br-eth1,physnet2:br-eth2,physnet3
# :br-eth3)
CONFIG_NEUTRON_LB_INTERFACE_MAPPINGS=

# Type of network to allocate for tenant networks (eg. vlan, local,
# gre)
CONFIG_NEUTRON_OVS_TENANT_NETWORK_TYPE=vlan

# A comma separated list of VLAN ranges for the Neutron openvswitch
# plugin (eg. physnet1:1:4094,physnet2,physnet3:3000:3999)
CONFIG_NEUTRON_OVS_VLAN_RANGES=physint:1000:1010,physext

# A comma separated list of bridge mappings for the Neutron
# openvswitch plugin (eg. physnet1:br-eth1,physnet2:br-eth2,physnet3
# :br-eth3)
CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=physint:br-eth3,physext:br-eth1

# A comma separated list of colon-separated OVS bridge:interface
# pairs. The interface will be added to the associated bridge.
CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-eth3:eth3,br-eth1:eth1

# A comma separated list of tunnel ranges for the Neutron openvswitch
# plugin (eg. 1:1000)
CONFIG_NEUTRON_OVS_TUNNEL_RANGES=

# The interface for the OVS tunnel. Packstack will override the IP
# address used for GRE tunnels on this hypervisor to the IP found on
# the specified interface. (eg. eth1)
CONFIG_NEUTRON_OVS_TUNNEL_IF=

# The IP address of the server on which to install the OpenStack
# client packages. An admin "rc" file will also be installed
CONFIG_OSCLIENT_HOST=10.16.11.27

# The IP address of the server on which to install Horizon
CONFIG_HORIZON_HOST=10.16.11.27

# To set up Horizon communication over https set this to "y"
CONFIG_HORIZON_SSL=n

# PEM encoded certificate to be used for ssl on the https server,
# leave blank if one should be generated, this certificate should not
# require a passphrase
CONFIG_SSL_CERT=

# Keyfile corresponding to the certificate if one was entered
```

```
CONFIG_SSL_KEY=

# The IP address on which to install the Swift proxy service
# (currently only single proxy is supported)
CONFIG_SWIFT_PROXY_HOSTS=10.16.11.27

# The password to use for the Swift to authenticate with Keystone
CONFIG_SWIFT_KS_PW=[REDACTED]

# A comma separated list of IP addresses on which to install the
# Swift Storage services, each entry should take the format
# <ipaddress>[/dev], for example 127.0.0.1/vdb will install /dev/vdb
# on 127.0.0.1 as a swift storage device(packstack does not create the
# filesystem, you must do this first), if /dev is omitted Packstack
# will create a loopback device for a test setup
CONFIG_SWIFT_STORAGE_HOSTS=10.16.11.27

# Number of swift storage zones, this number MUST be no bigger than
# the number of storage devices configured
CONFIG_SWIFT_STORAGE_ZONES=1

# Number of swift storage replicas, this number MUST be no bigger
# than the number of storage zones configured
CONFIG_SWIFT_STORAGE_REPLICAS=1

# FileSystem type for storage nodes
CONFIG_SWIFT_STORAGE_FSTYPE=ext4

# Shared secret for Swift
CONFIG_SWIFT_HASH=0210c4c2d2b54f65

# Size of the swift loopback file storage device
CONFIG_SWIFT_STORAGE_SIZE=50G

# Whether to provision for demo usage and testing
CONFIG_PROVISION_DEMO=n

# The CIDR network address for the floating IP subnet
CONFIG_PROVISION_DEMO_FLOATRANGE=172.24.4.224/28

# Whether to configure tempest for testing
CONFIG_PROVISION_TEMPEST=n

# The uri of the tempest git repository to use
CONFIG_PROVISION_TEMPEST_REPO_URI=https://github.com/openstack/tempest.git

# The revision of the tempest git repository to use
CONFIG_PROVISION_TEMPEST_REPO_REVISION=master

# Whether to configure the ovs external bridge in an all-in-one
# deployment
CONFIG_PROVISION_ALL_IN_ONE_OVS_BRIDGE=n

# The IP address of the server on which to install Heat service
CONFIG_HEAT_HOST=10.16.11.27
```

```
# The password used by Heat user to authenticate against MySQL
CONFIG_HEAT_DB_PW=[REDACTED]

# The password to use for the Heat to authenticate with Keystone
CONFIG_HEAT_KS_PW=[REDACTED]

# Set to 'y' if you would like Packstack to install Heat CloudWatch
# API
CONFIG_HEAT_CLOUDWATCH_INSTALL=n

# Set to 'y' if you would like Packstack to install Heat
# CloudFormation API
CONFIG_HEAT_CFN_INSTALL=n

# The IP address of the server on which to install Heat CloudWatch
# API service
CONFIG_HEAT_CLOUDWATCH_HOST=10.16.11.27

# The IP address of the server on which to install Heat
# CloudFormation API service
CONFIG_HEAT_CFN_HOST=10.16.11.27

# The IP address of the server on which to install Ceilometer
CONFIG_CEILOMETER_HOST=10.16.11.27

# Secret key for signing metering messages.
CONFIG_CEILOMETER_SECRET=9c9f2887ffad4f6f

# The password to use for Ceilometer to authenticate with Keystone
CONFIG_CEILOMETER_KS_PW=[REDACTED]

# The IP address of the server on which to install the Nagios server
CONFIG_NAGIOS_HOST=10.16.11.27

# The password of the nagiosadmin user on the Nagios server
CONFIG_NAGIOS_PW=[REDACTED]

# To subscribe each server to EPEL enter "y"
CONFIG_USE_EPEL=n

# A comma separated list of URLs to any additional yum repositories
# to install
CONFIG_REPO=

# To subscribe each server with Red Hat subscription manager, include
# this with CONFIG_RH_PW
CONFIG_RH_USER=

# To subscribe each server with Red Hat subscription manager, include
# this with CONFIG_RH_USER
CONFIG_RH_PW=

# To subscribe each server to Red Hat Enterprise Linux 6 Server Beta
# channel (only needed for Preview versions of RHOS) enter "y"
```

```
CONFIG_RH_BETA_REPO=n

# To subscribe each server with RHN Satellite,fill Satellite's URL
# here. Note that either satellite's username/password or activation
# key has to be provided
CONFIG_SATELLITE_URL=

# Username to access RHN Satellite
CONFIG_SATELLITE_USER=

# Password to access RHN Satellite
CONFIG_SATELLITE_PW=

# Activation key for subscription to RHN Satellite
CONFIG_SATELLITE_AKEY=

# Specify a path or URL to a SSL CA certificate to use
CONFIG_SATELLITE_CACERT=

# If required specify the profile name that should be used as an
# identifier for the system in RHN Satellite
CONFIG_SATELLITE_PROFILE=

# Comma separated list of flags passed to rhnreg_ks. Valid flags are:
# novirtinfo, norhnsd, nopackages
CONFIG_SATELLITE_FLAGS=

# Specify a HTTP proxy to use with RHN Satellite
CONFIG_SATELLITE_PROXY=

# Specify a username to use with an authenticated HTTP proxy
CONFIG_SATELLITE_PROXY_USER=

# Specify a password to use with an authenticated HTTP proxy.
CONFIG_SATELLITE_PROXY_PW=
```

Appendix G: Troubleshooting

Log files for a CloudForms Management Engine appliance are located under:

*/var/www/miq/vmdb/log/**

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-1: CFME Log Files

Alternately, log file output can be accessed via the CloudForms Management Engine Console. If more than one CFME appliance exists, a collection of log files across multiple CFME appliances within the same Zone can be collected to a centralized location.

To configure a central location for log file collection, login to the CFME appliance with the *admin* account. Navigate to **Configure**, **Configuration**. On the left window pane select **Diagnostics**. Verify the *Server* and *Zone* for which log collection is to take place.

On the right window pane select the **Collect Logs** tab. At the top click the **Edit** button. In the **Edit Log Depot Settings for Server: <server_name>** input box choose the **Type**. Options are *Samba*, *Network File System*, and *File Transfer Protocol*. Provide the proper inputs and click the **Save** button to complete.

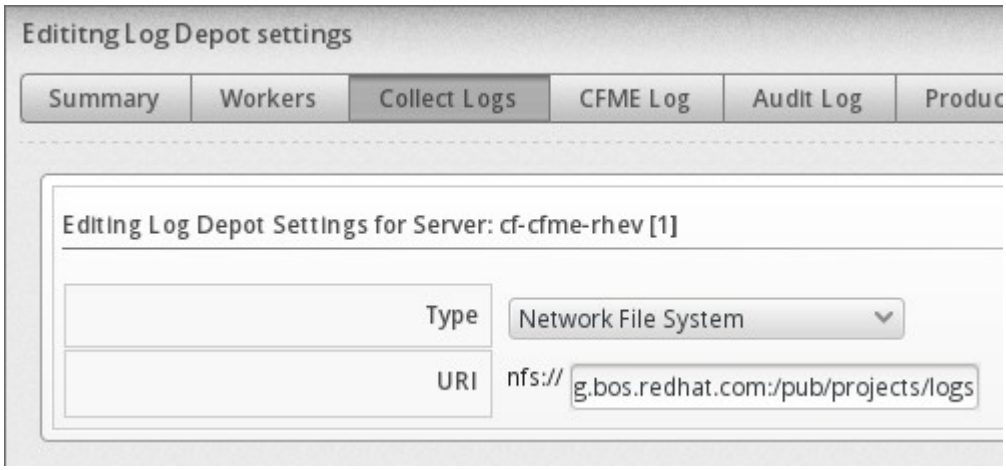


Figure G-1: Management Engine Log Collection Location

With the central log location configuration complete, click the **Collect** button. Chose **Collect all logs** or **Collect current logs**.

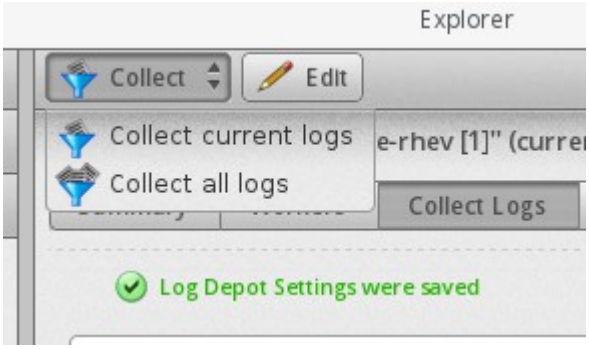


Figure G-2: Management Engine Log Collection

Information regarding log files and their location with Red Hat Enterprise Linux OpenStack Platform can be found in: *Red Hat Enterprise Linux OpenStack Platform 4: Getting Started Gude*[31].

31 https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Getting_Started_Guide/ch10.html

Appendix H: Configuration Files

All configuration files can be downloaded from the Red Hat customer portal[32]. A listing of the files and a brief description are provided below.

Files	Description
cloudinit.sh	cloud-init script used for post instance creation customization.
lamprsrv.rb	Customized in-line automate method for instance creation.
PreProvision.rb	Modified pre-provision method.
iptables	iptables configuration information for systems used.
cfrhos4.txt	RHEL OSP packstack answer file used for deployment.

Table H-1: Configuration Files

32 https://access.redhat.com/site/node/769223/40/1