Red Hat Reference Architecture Series

Red Hat Cloud, HP Edition

Reference Architecture

Marc Nozell, Solution Architect, HP

Ian Pilcher, Principal Architect, Red Hat

Version 1.0

May 19, 2011

Red Hat Cloud, HP Edition
Reference Architecture

1801 Varsity Drive™
Raleigh NC 27606-2072 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

Table of Contents

1 Introduction

In traditional IT environments, deploying a new application involves a high degree of coordination among many people and technology components. An IT architect might draw up a design specifying the servers, virtual machines, storage, and networks needed to support the application — and the connections between all of these components — adhering to established policies and standards. Based on this design, the different IT staff members responsible for servers, storage, networking, and facilities assemble and/or activate the needed resources. This entire process often takes multiple weeks and involves repeated communication between the different IT groups. This model also tends to create over-provisioned silos that waste resources.

Cloud computing is a model for enabling convenient, on-demand network access to a more standardized, shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Information on cloud technology, such as terminology and service models can be found in the Red Hat white paper, Cloud 101[1].

Some of the most well known cloud implementations are external to the corporate firewall, and are accessible through the Internet from companies offering Infrastructure as a Service (IaaS), such as Amazon's Elastic Compute Cloud (EC2). While there are a number of cost benefits to using external cloud implementations, relying on an external infrastructure is impractical or undesirable for many organizations, especially for critical business functions.

As a way forward, organizations can take advantage of the technologies and methodologies that have been developed for external clouds to build their own internal private cloud infrastructure. Using private cloud resources for deploying internal applications provides the cost and time savings of cloud technology by allowing the IT organization to focus on managing a pool of resources that can host many applications, rather than building and supporting systems for each application.

The first step in moving to a private cloud is consolidating and virtualizing server resources with virtual machines and virtual machine managers known as hypervisors. The Red Hat Cloud portfolio (introduced at Red Hat Summit 2010 as “Red Hat Cloud Foundations” or “RHCF”) goes beyond server virtualization to also include cloud management functions such as operating system, middleware, application, and scheduling management.

However, servers are only one component of the infrastructure. A complete infrastructure involves compute, storage, and networking components. In order to efficiently manage a private cloud, a solution that manages the other aspects of the hardware infrastructure is also required. The HP CloudSystem Matrix and the Matrix Operating Environment complement the Red Hat Cloud reference architecture, coordinating and automating virtualization, provisioning, and management operations across all of the components in a complex private cloud infrastructure.

HP CloudSystem Matrix and the Matrix Operating Environment bring the elasticity and resource pooling of cloud computing to the physical layer, providing the ideal platform for

1 Red Hat Cloud: Cloud 101, www.redhat.com/f/pdf/cloud/101_whitepaper.pdf

cloud infrastructure with the ability to rapidly add capacity and reallocate resources among applications. Red Hat Cloud adds the unique ability to provision and manage all of the software components on the physical layer — from the bare metal operating system or hypervisor to virtualization management to scheduling and application management. Together, HP CloudSystem Matrix and Red Hat Cloud provide a hardware and software platform that delivers on the promise of cloud computing.

Red Hat Cloud provides the software infrastructure required to build public, private, and hybrid clouds by using open source technology. The key technologies in RHCF include Red Hat Enterprise Linux (RHEL), Red Hat Enterprise Virtualization (RHEV), and Red Hat Network (RHN) Satellite.

Red Hat CloudForms, announced as beta at the Red Hat Summit in the Spring of 2011, brings a wide range of additional capabilities to organizations that are building private IaaS clouds. Red Hat CloudForms provides advanced features such as self-service, resource abstraction and pooling, image creation and management, and complete application life cycle management. These capabilities build on top of the Red Hat Cloud, as shown in Figure 1, and can be added incrementally. Red Hat Cloud is an avenue to start deploying cloud computing today.

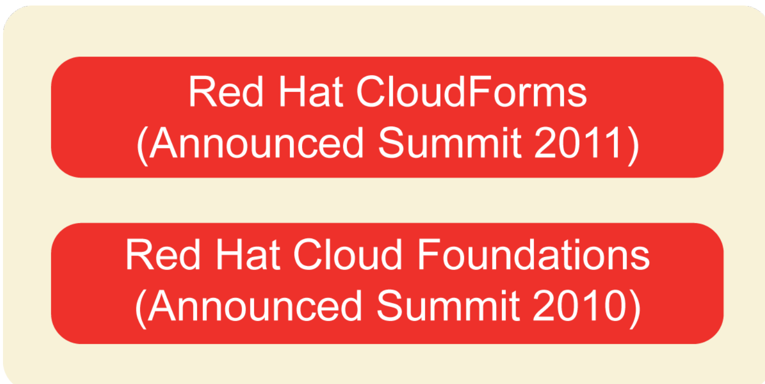

Figure 1: Red Hat CloudForms

This document is an implementation guide for the *Red Hat Cloud Foundations Reference Architecture, Private IaaS Clouds (RHCF-RA)*[2] on HP CloudSystem Matrix. It is a subset of of the original Red Hat Cloud Foundations reference architecture that shows how to take advantage of the HP converged infrastructure for servers, storage, and networking. In this document, a solution for private cloud infrastructure is described, followed by an implementation illustration of a working proof-of-concept. The RHCF-RA has extensive information on setting up all of the Red Hat components used in the solution. Refer to that document as needed for additional details.

2 Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds, inquiries.redhat.com/go/redhat/cloud-foundations

Part I – Solution Overview

2 Solution Overview

The goal of implementing a private cloud in an organization is to provide a flexible infrastructure that can quickly be provisioned or re-provisioned to meet rapidly evolving application and business requirements. Additionally, the solution should minimize the impact of normal IT infrastructure lifecycle management on applications, while giving IT more flexibility to perform critical tasks. These tasks include:

- Adding additional server or storage capacity as necessary to meet demand
- Migrating workloads to permit
  - Reallocation of resources to better meet requirements
  - Hardware and software maintenance or reconfiguration
  - Hardware upgrades and decommissioning of older hardware

The solution detailed in this reference architecture guide discusses how these goals can be met using the Red Hat Cloud on HP CloudSystem Matrix.

In the *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds*, a complete model of a cloud infrastructure solution is built. The consumers (users) of the infrastructure are given access to virtual machines (VM) that can be provisioned by the cloud management system on an on-demand basis. These virtual machines are referred to as *tenant VMs* because they are essentially renting space on the infrastructure as needed.

The cloud infrastructure management system is used by the cloud's owner (the service provider) to manage the physical and virtual systems in the cloud. In the original reference architecture, this is a pair of systems in a highly available cluster that are dedicated to managing the cloud infrastructure. The key components on this pair of servers are Red Hat Enterprise Linux and Red Hat Enterprise Virtualization.

The hypervisor hosts are the pool of physical machines that host the virtual machines that are built on-demand for users of the cloud infrastructure to run their applications. The tenants' virtual machines can be created and migrated as necessary across the physical machines to give the service provider the flexibility needed to effectively manage the cloud infrastructure.

The base reference architecture is shown in Figure 2.

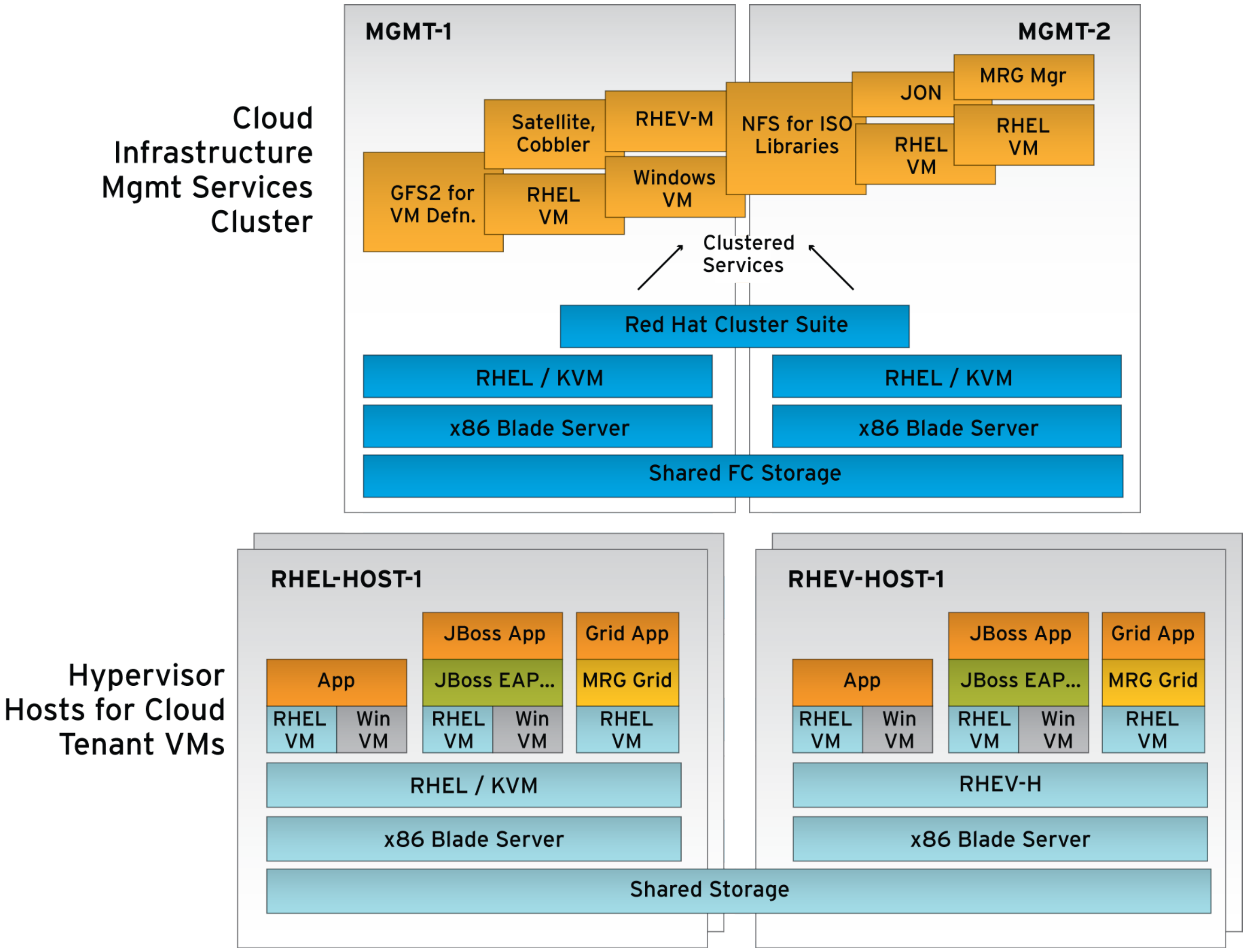

Figure 2: Red Hat Cloud Foundations Reference Architecture

The solution implemented in this guide differs slightly from the original reference architecture in order to take advantage of the capabilities of the HP CloudSystem Matrix. Some of the differences include:

- The HP Matrix Operating Environment is used to provision the entire physical system environment, including servers, storage, and networking. An additional blade is used to host the HP Matrix Operating Environment.
- The Matrix Operating Environment is used to streamline deployment of Red Hat Enterprise Linux to all of the bare machines in environment. In the original reference architecture, Red Hat Enterprise Linux is installed manually on the cloud infrastructure management nodes and then Red Hat Network Satellite is used to bootstrap the hypervisor host machines.
- Only Red Hat Enterprise Linux hosts using the KVM hypervisor are used. The Red Hat Enterprise Virtualization Hypervisor (RHEV-H) hosts on bare metal are omitted. The RHEL+KVM hypervisor hosts are registered with Red Hat Enterprise Virtualization Manager (RHEV-M) to provide the benefits of RHEV management. Using RHEL+KVM

for hypervisor deployment, allows those systems to be managed using the HP Matrix Operating Environment.

- The additional system and cluster configuration for the Cloud Infrastructure Manager are omitted for brevity.
- The use of Red Hat MRG Grid and JBoss Enterprise Application Platform (EAP) as sample applications running on cloud virtual machines are omitted as is the accompanying management software JBoss Operations Network (JON) and MRG Grid Manager.

The solution as deployed in this guide is shown in Figure 3.

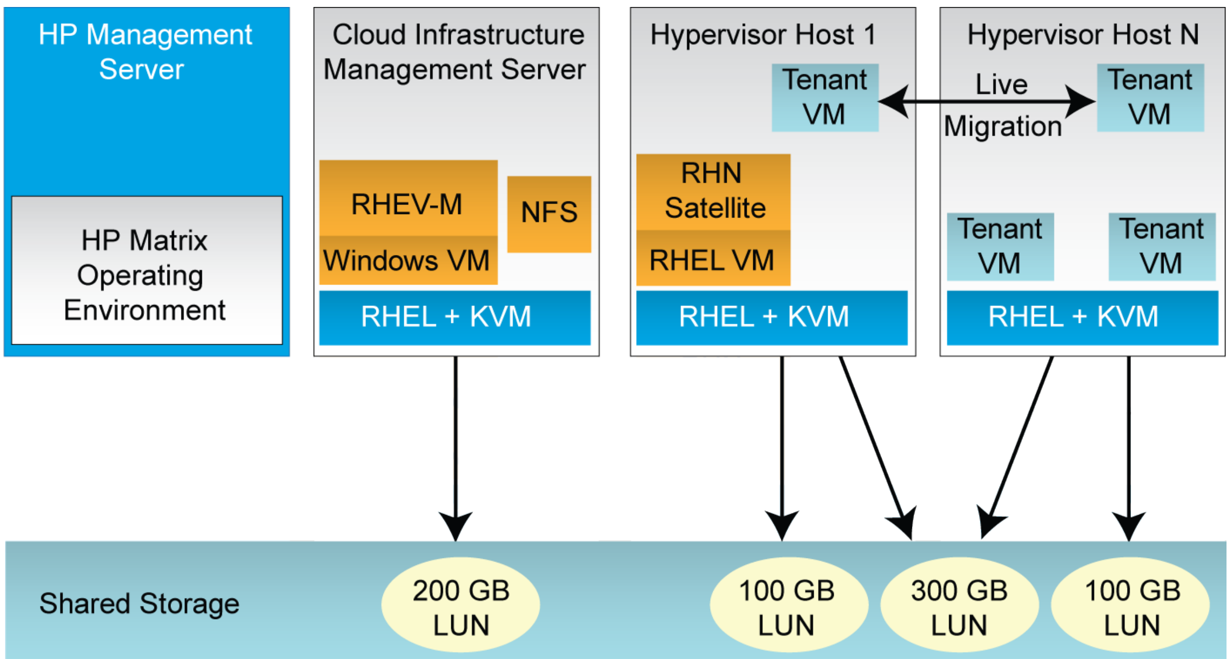

Figure 3: Red Hat Cloud, HP Edition

2.1 Solution Components

2.1.1 HP CloudSystem Matrix

HP CloudSystem Matrix is a converged infrastructure platform designed to simplify the deployment of applications and business services by delivering IT capacity through pools of readily deployed resources. The goal of CloudSystem Matrix is to accelerate provisioning, optimize IT capacity across physical and virtual environments, and to help ensure predictable delivery and service levels. CloudSystem Matrix is a pre-integrated hardware, software, and services solution. It includes the HP Matrix Operating Environment, FlexFabric, and Virtual Resource Pools.

An administrator can use the management console to define pools of server, network, and storage resources, register user accounts, and assign resource pools to users. Through the console, the administrator can also define virtual machine images and software deployment jobs.

The HP StorageWorks 4400 Enterprise Virtual Array (EVA4400) can be factory integrated to the CloudSystem Matrix, which is the recommended option for this reference architecture. Alternatively, Matrix can be connected to a separate HP StorageWorks or supported third-party Fibre Channel (FC) storage area networks (SANs). The physical machines in the HP CloudSystem Matrix can boot from the SAN in order to provide maximum flexibility.

2.1.1.1 HP FlexFabric

HP FlexFabric is a virtualized, high-performance, low-latency network that consolidates Ethernet and storage networks onto a single fabric. Physical connections are dynamically multiplexed between multiple Ethernet and storage networks. Identifying information (FC WWNs and Ethernet MAC addresses) is assigned by the Matrix Operating Environment, rather than locked to physical hardware.

Normally the fixed hardware identifiers for networking and storage results in a number of extra steps for system, storage, and network administrators when things need to be reconfigured in cases such as replacing failed hardware or performing upgrades. By virtualizing and managing these identifiers in the HP converged infrastructure, the extra steps and coordination with multiple operations people are eliminated.

2.1.1.2 Matrix Operating Environment

The Matrix Operating Environment includes an integrated service designer, self-service portal, and auto-provisioning capabilities; the tools to manage and optimize the resource pools; and a recovery management solution.

The infrastructure orchestration capabilities provided in Matrix enable provisioning and re-provisioning of the shared pools of servers, storage, and network connectivity as needed, based on predefined templates. An IT architect or designer uses a graphical designer tool to build and publish infrastructure service templates incorporating physical and virtual servers, storage, networks, and the connections between them. Typically, each template represents the complete infrastructure needed to host a specific application service. A catalog of published service templates is then available to users. Using a simple self-service portal, users can request an instance of a service template to be automatically provisioned.

2.1.2 Benefits of the HP Converged Infrastructure for the Private Cloud

The Matrix Operating Environment coordinates and automates operations for all physical components, including:

- Selecting appropriate servers, based on processor and memory requirements
- Allocating and initializing SAN storage
- Managing access to shared SAN storage

- Partitioning Flex10 networks and assigning MAC addresses
- Installing operating systems
- Defining workflows that can install software and apply the necessary updates

The Matrix Operating Environment's ability to easily provision both hardware and software allows a cloud administrator to rapidly add capacity to the cloud infrastructure, maintaining the illusion of infinite computing resources. At times when the cloud workload is lighter, physical resources can be reallocated to other applications.

2.2 Red Hat Cloud

2.2.1 Red Hat Enterprise Linux

Red Hat Enterprise Linux is the world's leading open source application platform. Red Hat Enterprise Linux is used as the base operating system on the physical servers in this solution

2.2.2 Red Hat Enterprise Linux KVM Virtualization

The Kernel-based Virtual Machine (KVM) is a full virtualization solution provided with Red Hat Enterprise Linux. The KVM hypervisor is a Linux kernel module built for the standard Red Hat Enterprise Linux kernel. KVM can run multiple, unmodified, fully virtualized guests including Microsoft Windows and Linux operating systems. The KVM hypervisor in Red Hat Enterprise Linux is managed with the libvirt API and tools built for libvirt, including virt-manager and virsh. Virtualized guests are run as Linux processes and threads that are controlled by these modules.

The KVM hypervisor supports full virtualization of guest systems. Full virtualization offers the most flexibility since all of the guest hardware is emulated, which allows unmodified guest operating systems to be used. However, there is a performance cost for that flexibility, as all guest I/O runs through the virtualization layer. In order to improve guest I/O performance, virtualization-aware storage and network drivers, known as para-virtualized drivers, can be installed on the guest operating system. These drivers provide a much more direct path to the I/O on the host system. The para-virtualized drivers are available for Linux and Microsoft Windows guests, and are in the VirtIO driver package.

Additional information can be found in the Red Hat Enterprise Linux Virtualization Guide[3].

2.2.3 Red Hat Enterprise Virtualization for Servers

Red Hat Enterprise Virtualization for Servers is an end-to-end virtualization solution that offers comprehensive management functionality, including live migration, high availability, and a system scheduler, to enable large-scale datacenter-level virtualization. RHEV consists of two parts, the management system for controlling the physical and virtual machines in the environment, and the hypervisor and platform management infrastructure that runs on the

3 Red Hat Enterprise Linux Virtualization Guide, docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/pdf/Virtualization/Red_Hat_Enterprise_Linux-5-Virtualization-en-US.pdf

physical hosts and supports guests.

2.2.3.1 Red Hat Enterprise Virtualization Manager

Red Hat Enterprise Virtualization Manager delivers a centralized management system to administer and control all aspects of a virtualized infrastructure, from host and guest management through storage management and high availability. RHEV-M provides a rich user interface that allows an administrator to manage their virtual infrastructure from a web browser.

Red Hat Enterprise Virtualization manager manages both Red Hat Enterprise Virtualization Hypervisors and Red Hat Enterprise Linux hosts with the KVM hypervisor, delivering leading performance and scalability for virtual machines on a stable and secure platform.

2.2.3.2 Red Hat Enterprise Virtualization Hypervisor

Red Hat Enterprise Virtualization Hypervisor is a modern hypervisor based on KVM that can be deployed as either RHEV-H, a standalone bare metal hypervisor that installs directly on the hardware instead of a host operating system, or as Red Hat Enterprise Linux as the host operating system with the KVM hypervisor.

RHEV builds on the virtualization capabilities provided by KVM to add centralized management and control for managing multiple hypervisor hosts and all of their guest VMs. RHEV-H includes the platform management infrastructure that allows either style of hypervisor deployment to be administered by RHEV-M.

The functionality of RHEV that is important for managing private cloud infrastructure includes:

- Live Migration — allows virtual machines to be moved between hosts with no service interruption.
- High Availability — Virtual machines can be configured to automatically restart on another host in case of failure. Priority levels can be specified to allow VMs to be restarted in order of importance.
- Centralized Host Management — Hypervisor host configuration, including storage and network configuration such as bonding and VLANs, can be administered through the RHEV-M console.
- Storage Pool Management — RHEV can automate the management of pools of storage used by VMs. Thin provisioning is available to allow more efficient management of storage by delaying the allocation of physical space until it is actually required by the VMs.
- VM Image Management — Templates can be defined to automate the creation of guest VMs. Point-in-time snapshots can be created to allow VMs to be rolled back to a previous state.
- System Scheduling — The System Scheduler allows policies to dynamically allocate VMs to physical hosts based on criteria such as load balancing of resource utilization. An advanced feature of the System Scheduler is Power Saving mode, which can reduce power consumption during off-peak periods by consolidating running VMs onto

a smaller number of physical hosts.

Additional information on RHEV, including an overview of the architecture and terminology, can be found the in the *Red Hat Enterprise Virtualization for Servers Administration Guide*[4].

2.2.4 Red Hat Network

Red Hat Network (RHN) is a web-based system management platform that provides software updates, configuration management, provisioning, and monitoring across both physical and virtual Red Hat Enterprise Linux servers. Red Hat Network is hosted by Red Hat and access is via the Internet. Red Hat Network is included with a Red Hat Enterprise Linux subscription. An administrator can use RHN from a browser to help ensure that all physical and virtual systems are up-to-date with the latest bug and security fixes and are consistent with each other. This is especially important in heavily-virtualized environments like private cloud implementations, where there are significantly more operating system installs to manage on all of the tenant virtual machines.

2.2.5 Red Hat Network Satellite

Red Hat Network Satellite provides the functionality of RHN but is hosted within the enterprise. Because the Satellite server is located on premises, it offers much greater functionality and customization. The Satellite server communicates with Red Hat's servers over the Internet to download new content and updates. For high security environments, Satellite can also operate in disconnected mode, with Red Hat content manually transferred across a physical air gap.

Other RHN Satellite features that benefit private cloud administrators include:

- An embedded database to store packages, profiles, and system information
- The ability to distribute internal and/or third-party applications to all of the systems and keep them up-to-date using the same tool set
- Access to advanced features in the Provisioning Module, such as bare-metal PXE boot provisioning and integrated network install trees
- Access to Red Hat Network Monitoring Module for tracking system and application performance
- Customization and integration: an API layer enables scripts to be created to automate functions or integrate with existing management applications

4 Red Hat Enterprise Virtualization for Servers Administration Guide, docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Virtualization_for_Servers/

2.2.6 Other Components

This proof-of-concept is focused on integrating the Red Hat Cloud reference architecture with HP's Matrix Operating Environment. Thus, there are some components of the reference architecture that are not installed. The purpose of these components in the original reference architecture is to provide the capabilities to run two additional types of workloads on the infrastructure: grid computing jobs and Java enterprise applications.

- **JBoss Enterprise Application Platform (EAP)** is the market leading platform for innovative and scalable Java applications. Integrated, simplified, and delivered by the leader in enterprise open source software, it includes leading open source technologies for building, deploying, and hosting enterprise Java applications and services.
- **JBoss Operations Network (JON)** is an integrated management platform that simplifies the development, testing, deployment, and monitoring of JBoss middleware.
- **Red Hat Enterprise MRG Grid** provides high throughput and high performance computing. Additionally, it enables enterprises to move to a utility model of computing to help achieve both higher peak computing capacity and higher IT usage by leveraging existing infrastructures to build high performance grids.

Because this reference architecture anticipates that all of these components will be installed in virtual machines, shielding them from differences in servers, storage, networking, etc., the detailed installation instructions in *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds* can be used with very few changes.

2.3 Infrastructure Build Overview

Here is a high level overview of how the selected products are used to build this private cloud solution. The full details are included in Part II, which details the steps to build a full proof-of-concept.

1. The HP Insight Orchestration Designer graphical tool is used to define the template for the reference architecture. This includes the two server groups, the cloud infrastructure server and the hypervisor hosts, and the storage requirements and network connectivity. The template includes details such as the minimum processor and memory requirements for each server that the Matrix Operating Environment will use when determining what physical hardware to use in deploying the solution.
2. The Matrix Operating Environment builds the Cloud Infrastructure Server and installs the operating system using a Matrix provided Red Hat Enterprise Linux 5 template.
3. The hosted version of the Red Hat Network system management tool is used to install the KVM virtualization software and any necessary updates on the cloud infrastructure server.
4. A Network File System (NFS) shared volume Is created to host the templates and ISO files for creating virtual machines.
5. The virtual machine to run the RHEV-M management system is created. The guest operating system, Microsoft Windows Server 2008, is installed on the VM. The RHEV-

M software is then installed on top of Windows.

6. To automate deployment of all of the necessary software on the hypervisor hosts, a workflow is created in the Matrix Operating Environment that registers the system with RHN and identifies the system so that the specific software for hypervisor hosts is installed. An RHN activation key is created for the system to use when registering with RHN. On RHN, the systems associated with the activation keys are pre-subscribed to configuration channels that cause the desired software to be installed.
7. RHEV-M is used to define one or more virtual datacenters and storage pools for the hypervisor hosts to use when running tenant VMs. When installed, RHEV-M creates a default virtual datacenter that can be used If there is no need to differentiate between different groups of tenant VMs.
8. In order to install RHN Satellite, a VM is created using RHEV-M. Red Hat Enterprise Linux 5 and RHN Satellite are then installed after downloading the software from hosted RHN.

3 Managing the Private Cloud Infrastructure

The scenarios below illustrate how the Matrix Operating Environment and the Red Hat Enterprise Virtualization Manager can be used together to manage a private cloud infrastructure.

3.1 Creating and Modifying Tenant VMs

To add tenant VMs to the cloud infrastructure, create VMs in the appropriate virtual datacenter using the RHEV-M console. All management operations on tenant VMs are performed using RHEV-M. See section 7, *Deploying VMs in Hypervisor Hosts* of the original reference architecture.

3.2 Adding Hypervisor Hosts for Tenant VMs

As demand for cloud services grows, cloud administrators can seamlessly scale the cloud infrastructure by adding more hypervisor hosts.

1. Use the Matrix Operating Environment to provision additional hypervisor server blades. The Matrix Operating Environment allocates blade server(s), creates LUN(s) for boot/root, grants access to shared LUN(s), and partitions the Flex10 network.
2. Use RHEV-M to add new hypervisor(s) hosts.

3.3 Adding Storage

Storage can be added to the cloud infrastructure non-disruptively.

1. Use the Matrix Operating Environment to create additional shared LUN(s) and make them visible to existing hypervisor blades. The infrastructure template is modified, so that new shared LUN(s) is visible to newly deployed hypervisors.
2. Use RHEV-M to add new shared LUN(s) to the existing storage pool or create additional pools.

3.4 Maintaining Hypervisor Hosts

Cloud administrators can perform maintenance on cloud infrastructure components without impacting the consumers of cloud services.

1. Use RHEV M to place the hypervisor host in maintenance mode. All virtual machines running on the hypervisor host can be live migrated to other hypervisors without impacting their users.
2. Use the Matrix Operating Environment to perform required maintenance, such as host operating system upgrade, firmware upgrade, network, or storage reconfiguration, etc.

3. Use RHEV-M to place the hypervisor host back in service by activating it. RHEV-M distributes workload to the hypervisor host as required.

3.5 Replacing Hypervisor Hardware

The cloud infrastructure shields users from hardware failures and simplifies the amount of work that needs to be performed when hardware is replaced by managing the identifiers that are normally tied to the hardware.

1. The Matrix Operating Environment alerts administrator to impending hardware failure.
2. Use RHEV-M to place the hypervisor host in maintenance mode. All virtual machines running on the hypervisor are live migrated to other hypervisors without impacting their users. Alternatively if RHEV-M detects that the hypervisor host has failed, RHEV-M can restart the VMs on other hypervisor hosts. If the VMs have been assigned a priority level, they are restarted in priority order.
3. When a blade is replaced, the Matrix Operating Environment preserves the configuration, including storage mapping, FC WWNs, Flex10 network partitioning, MAC addresses, etc.
4. Use RHEV-M to place the hypervisor host back in service. The workload is distributed to the newly reactivated server by RHEV-M as required.

Part II – Proof of Concept Implementation

4 Proof of Concept Environment

A proof-of-concept showing the Red Hat Cloud Foundations Reference Architecture for Private IaaS clouds implemented using the HP CloudSystem Matrix converged infrastructure was conducted by Red Hat and HP personnel at HP's Partner Technology Access Center (PTAC) in Houston, Texas, during January, 2011. Three blades and 700 GB of storage were used to implement the cloud infrastructure management system and the hypervisor hosts. The HP Matrix Central Management Server ran on its own dedicated blade.

The cloud infrastructure server ran on one blade, with a 200 GB boot/root LUN allocated by the Matrix environment. Red Hat Enterprise Linux 5 was installed by the Matrix Operating environment. KVM virtualization was used to run Red Hat Enterprise Virtualization Manager (RHEV-M) under the Microsoft Windows Server 2008 operating system. The cloud infrastructure manager server also provides an NFS share for use as the RHEV ISO library.

Two blades were configured as hypervisors. Each was allocated its own dedicated 100 GB boot/root LUN. To provide storage for tenant VMs, a 300 GB shared LUN was created and made accessible to both hypervisors. HP Matrix Operations Orchestration workflows, in conjunction with the Red Hat Network (RHN) system management platform, were used to automate the deployment of Red Hat Enterprise Linux 5, and the Red Hat Enterprise Virtualization Hypervisor (RHEV-H) management agents on the hypervisor hosts. The automation makes it easy to deploy additional hypervisor hosts as needed to scale the cloud infrastructure.

The solution is shown in Figure 4.

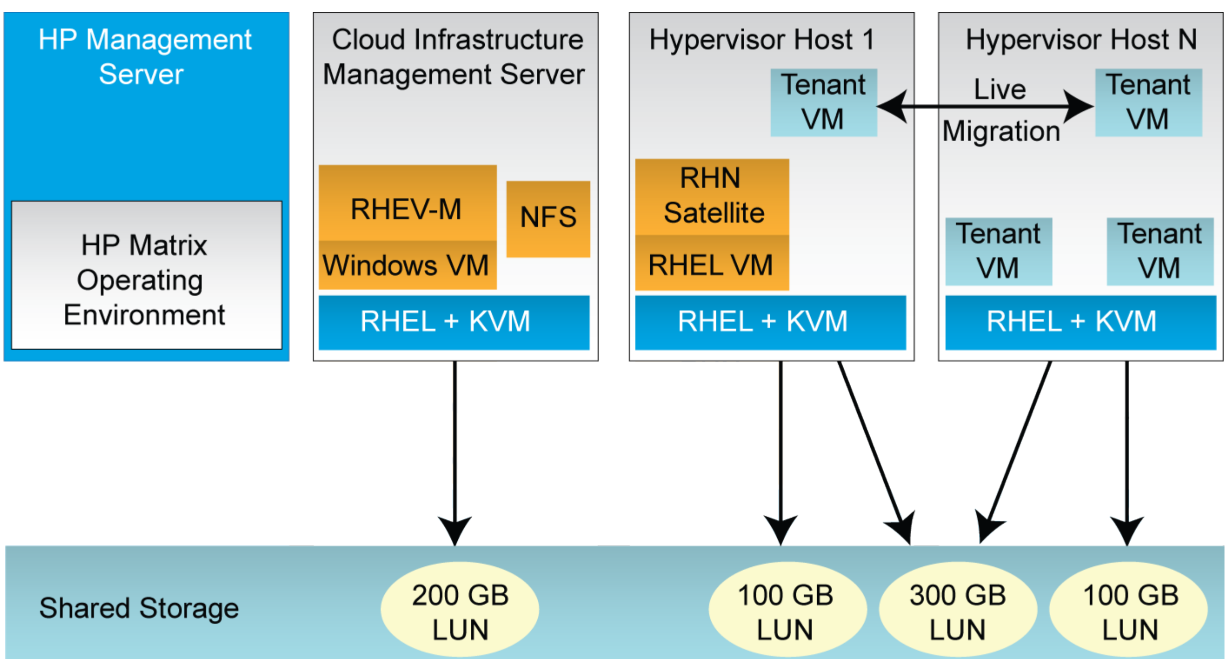

Figure 4: Red Hat Cloud on HP CloudSystem Matrix

The result of the proof-of-concept is:

- A hardware environment consisting of servers, storage, and networking that is managed by the HP CloudSystem Matrix
- A cloud infrastructure management system, with software managed by the hosted version of the Red Hat Network
- A private cloud infrastructure that is managed by Red Hat Enterprise Virtualization for Servers
- System software management services for tenant VMs provided by RHN Satellite

4.1 CloudSystem Matrix Hardware

The configuration used was standard CloudSystem Matrix starter kit with four ProLiant server blades.

The following are the key hardware components used in this reference architecture:

- HP CloudSystem c7000 enclosure
- HP ProLiant BL460c G6 (half-height) server blade for the Matrix central management server
- HP StorageWorks 4400 Enterprise Virtual Array Starter Kit (StorageWorks EVA4400 controller pair enclosure with embedded switches, drive enclosure, and eight 450 GB 10K-rpm 4 GB dual-port FC EVA M6412 hard disk drives, CommandView and SmartStart media, and 4x 1 TB licenses to use)
- HP Virtual Connect Flex-10 10 Gb Ethernet modules (redundant pairs)
- HP Virtual Connect 8 Gb 24-port FC modules (redundant pair)

Three HP ProLiant BL460c G6 server blades were used for the cloud infrastructure server and the Red Hat Enterprise Virtualization hypervisor hosts. The server configurations included:

- Dual processors — Intel® Xeon® Processor E5540 (2.53 GHz, 8 MB L3 Cache, DDR3-1066, HT, Turbo 1/1/2/2)
- 32 GB (4 x 8 GB Dual Ranked PC3-10600 Registered DIMMs)
- QLogic QMH2562 8 Gb FC HBA for HP CloudSystem c-Class

4.2 Storage Configuration

In the standard CloudSystem Matrix configuration, there is no local storage. All storage is located on an HP StorageWorks EVA4400 SAN. This includes the boot devices for all of the servers, as well as the shared disk between the KVM hypervisor servers.

For this proof-of-concept, 200 GB is allocated to the cloud infrastructure server. Each of the hypervisor servers is allocated 100 GB for its root volume and given access to a shared

300GB LUN that is used to store the data for the tenant VMs. Table 1 lists the storage breakout.

Table 1. Storage breakout.

Size	Description
200 GB	Cloud Infrastructure Management Server
100 GB	Hypervisor Host 1
100 GB	Hypervisor Host N
300 GB	Cloud storage for tenant VMs, shared across all hypervisor hosts.

4.3 Network Configuration

The HP PTAC lab is configured with the networks listed in Table 2 within the FLEX-10 environment: DNS and DHCP services are provided by the PTAC environment.

Table 2. Networks in the HP PTAC lab.

Network	IP Address Range	Bandwidth
production	10.95.36.191-10.95.36.209	8 Gb
deployment	192.168.100.0/22	1 Gb
management	192.168.0.0/22	1 Gb

4.4 Software Versions

4.4.1 HP Matrix Operating Environment

This proof-of-concept was built using the HP Matrix Operating Environment v6.1 update 1, which was run on a dedicated blade.

4.4.2 Red Hat Enterprise Virtualization for Servers

Red Hat Enterprise Virtualization for Servers version 2.2 was used for both the manager component on the Cloud Infrastructure Management Server and the platform management agents on the hypervisor hosts.

The current releases of RHEV-M will only run under Microsoft Windows. A virtual machine running Microsoft Windows Server 2008, SP2 was installed on the cloud infrastructure server to host RHEV-M.

4.4.3 Red Hat Enterprise Linux

The systems were initially installed using Red Hat Enterprise Linux 5 update 5. After installation, the systems were updated to Red Hat Enterprise Linux 5.6 by using yum. Note: the current release of RHEV at the time this guide was written was 2.2, which only supported Red Hat Enterprise Linux 5. In the future, a newer release of RHEV will be available with support for Red Hat Enterprise Linux 6.

4.4.4 Red Hat Network Satellite

Red Hat Network Satellite Server version 5.4 was installed to provide system management capabilities for the guest VMs on the hypervisor hosts.

5 Implementation

The implementation of the private cloud infrastructure in this guide is a subset of the Red Hat Cloud Foundations Reference Architecture for Private IaaS clouds. The information provided here details the specific steps for building the system while taking advantage of the capabilities of the HP CloudSystem Matrix. This guide only covers the pieces that are specific to the implementation on HP. Refer to the original document for the complete implementation of the reference architecture.

5.1 *Changes from Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds*

There are a number of differences between the configuration used for this proof-of-concept and the configuration described in *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds* which is shown in Figure 5, below.

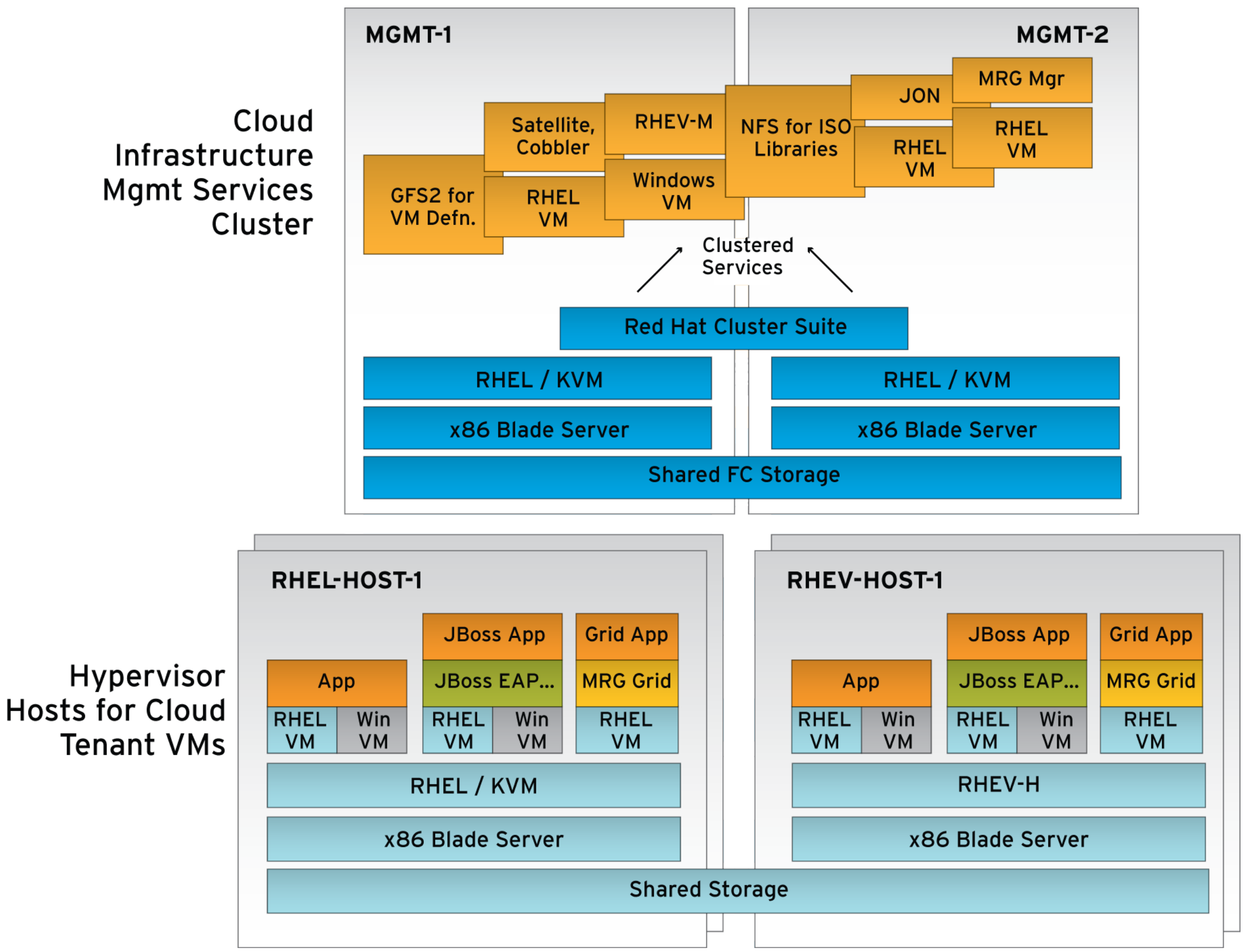

Figure 5: Red Hat Cloud Reference Architecture: Private IaaS Clouds

- For the sake of simplicity, this proof-of-concept uses a single cloud infrastructure server, rather than the management services cluster used in the original reference architecture. In a production environment, use of a high-availability cluster for RHEV-M is highly recommended, and *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds* provides detailed instructions for setting up such a cluster.

  It is worth noting, however, that the services hosted on the infrastructure server in this configuration are not critical for the continued operation of the guests in the RHEV environment. If a RHEV-M outage does occur, no changes to the environment will be possible (starting, stopping, or migrating virtual machines), but all virtual machines will remain running on their current host until the RHEV-M is back online. The absence of the NFS ISO library will generally only affect operating system installation in new guests.

- Two styles of hypervisor deployment are described in the original reference architecture, Red Hat Enterprise Linux with KVM virtualization and RHEV-H. In this guide only RHEL + KVM style hypervisor hosts are used. Centralized RHEV Management of the RHEL+KVM hypervisor hosts is enabled by using the RHEV-H management agents. The Matrix is used for installing Red Hat Enterprise Linux on the hypervisor hosts at deployment time. Integration of Matrix Operations Orchestration workflows and RHN are used to automate the installation of the KVM virtualization software and RHEV management agents.

- This proof-of-concept places the RHN Satellite virtual machine within the RHEV environment, rather than on the infrastructure server/cluster. RHN Satellite is not needed for installation of the hypervisor hosts because the Matrix Operating Environment provides bare-metal provisioning functionality. Thus, RHN Satellite is installed at a later stage and used for provisioning and management of RHEV guests only.

  Since Satellite is not available during installation of the infrastructure server or the hypervisors, those systems are registered with *hosted* Red Hat Network (rhn.redhat.com).

- As noted in section 4.3.4, the JBoss Enterprise Application Platform (EAP), JBoss Operations Network (JON), and MRG Grid components are not installed as part of this proof-of-concept. Thus the JON and MRG Manager virtual machines are not present in this configuration.

5.2 Build Preparation

This section describes the steps to prepare for the infrastructure build using the HP Matrix Operating Environment.

5.2.1 Infrastructure Template Design

The Matrix Operating Environment includes an infrastructure orchestration designer with a web-based interface that allows a system architect to create a template to define the minimum required resources for a particular purpose. See Figure 6. A template can include multiple server groups with different server sizes, the disk requirements, and network configuration information. At deployment time, the Matrix Operating Environment is responsible for allocating the requested server, storage, and network resources for the service from the available hardware resource pools. For additional information see the *HP Insight Orchestration User Guide*[5].

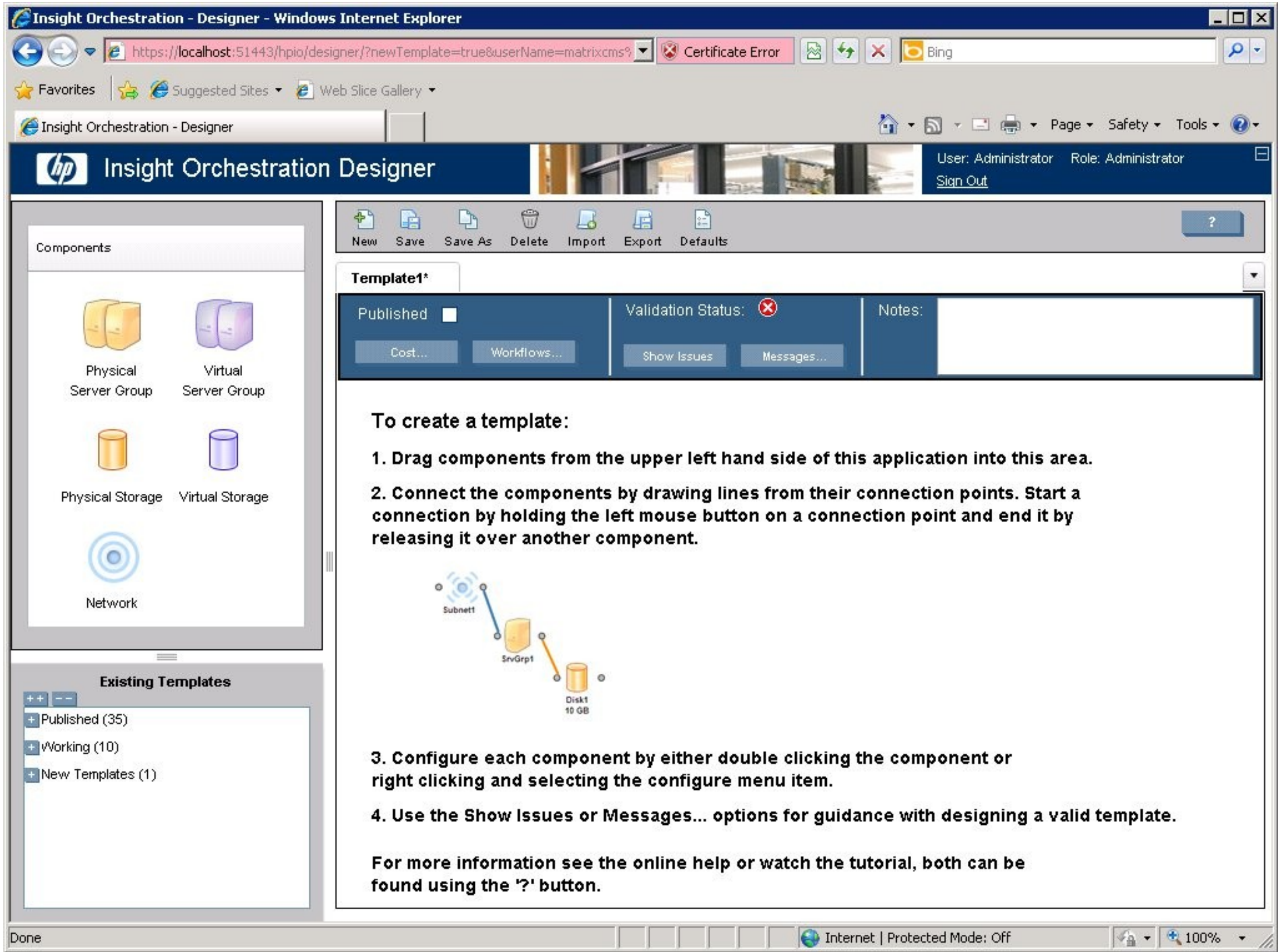

Figure 6: HP Insight Orchestration Designer.

A Matrix infrastructure orchestration template is designed that describes the two servers groups used to host the cloud management infrastructure and the hypervisor hosts. Figure 7 shows the *RHCF-RefArch* template in the designer interface. There are two server groups defined, *RHEVMkvm* and *KVMhosts*. The RHEVMkvm contains just one server with Red Hat Enterprise Linux 5.5 attached to a 200 GB SAN disk for the root filesystem. This server will be used to host the RHEV-M management station. The KVMhosts server group contains two servers running Red Hat Enterprise Linux 5.5, each with a 100 GB SAN disk for the root

5 HP Insight Orchestration User Guide, bizsupport2.austin.hp.com/bc/docs/support/SupportManual/c01658528/c01658528.pdf

filesystem and a shared 300 GB SAN disk. The three networks defined, *prod*, *mgmt*, and *deploy,* are mapped to three NICs defined in the Flex-10 network adapter.

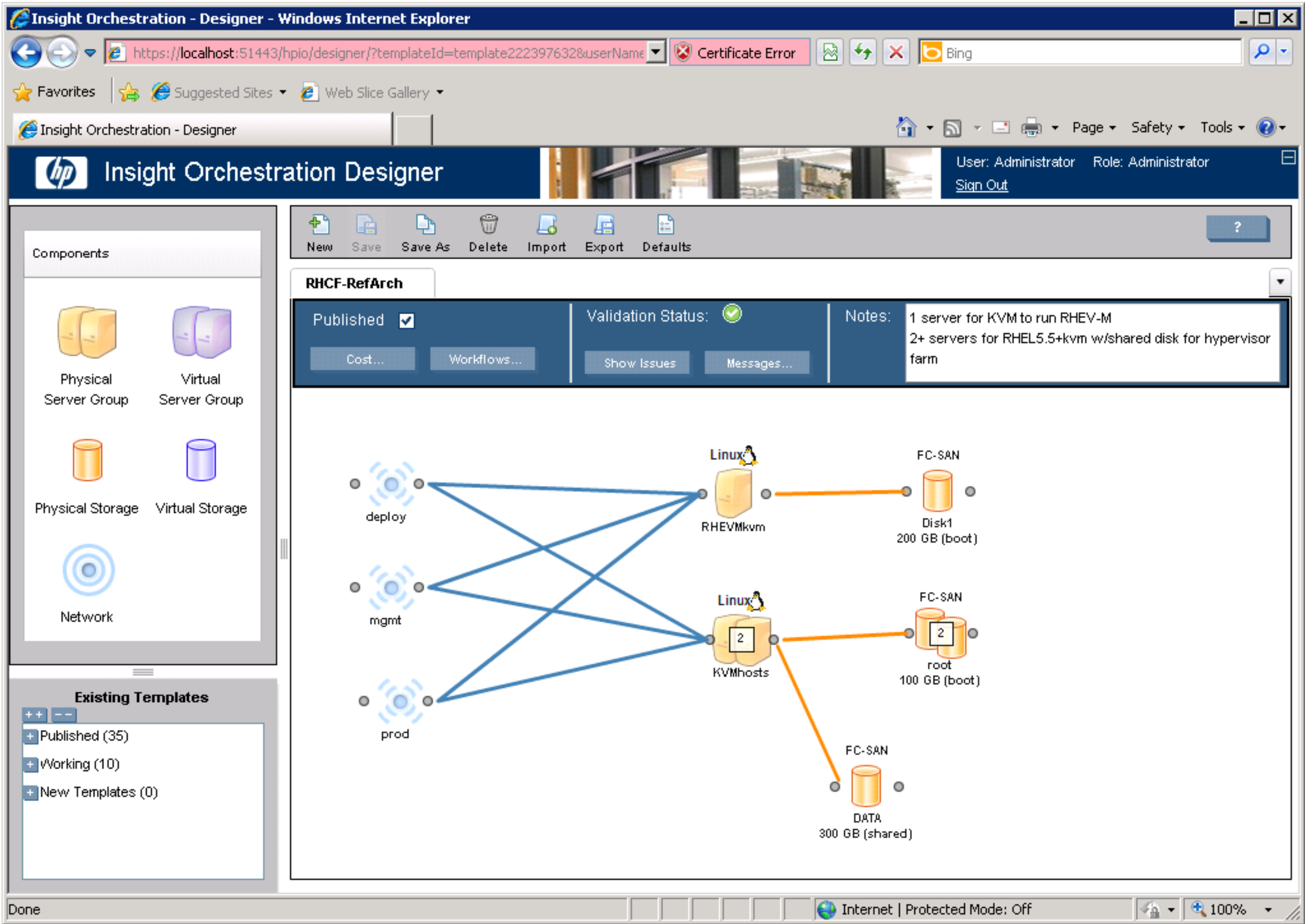

Figure 7: HP Insight Orchestration Designer.

The template that was designed for this proof-of-concept, and additional files that automate some of the workflows, are available for download. See Section 8: *Downloads*, for the details. The template, *RHCF-RefArch.xml* is in the *Teplate.ZIP* archive. To use the downloaded template, it needs to be imported into the Insight Orchestration Designer. Note that since the template defines storage, networking, and boot images, there will be errors when importing it, since it won't match the physical environment it is being imported into. This is to be expected, and those parameters need to be changed to match the target environment. Alternatively, the designer can be used to create an entirely new template. The workflow files from the downloaded template can be imported and attached to the newly created template.

5.2.1.1 Physical Server Specification

To specify the minimum server requirements for the cloud infrastructure server, right click on the RHEVMkvm server group to bring up the server configuration dialog box as shown in Figure 8. The requested server must be a physical blade that has a minimum of 4 GB of memory and two x86 64-bit CPUs running at least 2 GHz. The Matrix infrastructure will allocate at least this size of server when the service is deployed.

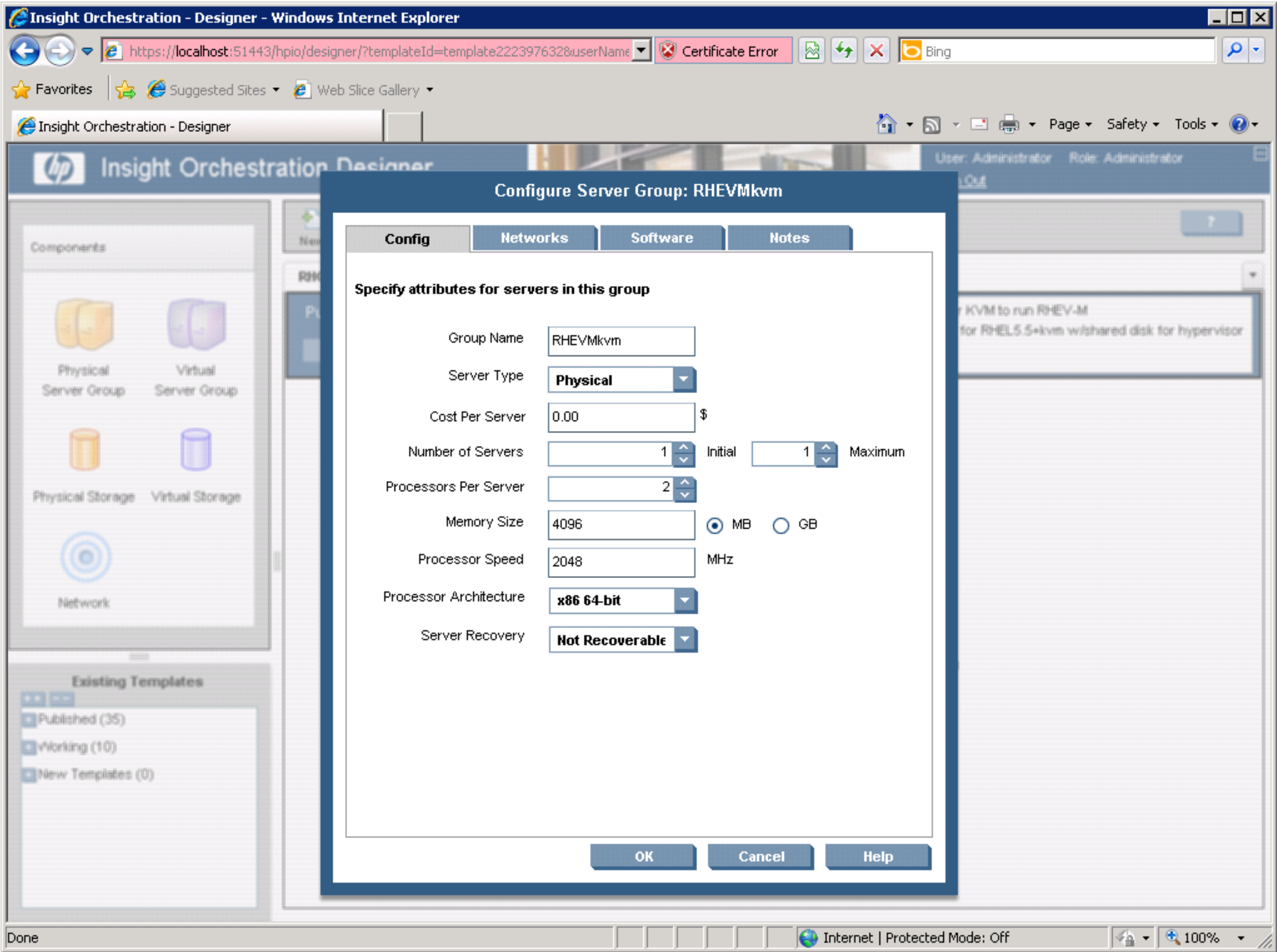

Figure 8: RHEVMkvm server group configuration.

The process is repeated to specify the sizing for the hypervisor hosts by right clicking on the KVMhosts server group as shown in Figure 9. Two physical servers are requested with at least 4 GB of memory and two x86 64-bit CPUs running at at 2 GHz or higher. The Matrix infrastructure will allocate at least this size of server when the service is deployed. Initially two servers will be deployed, but later up to a total of 99 servers can be added by using the Matrix portal.

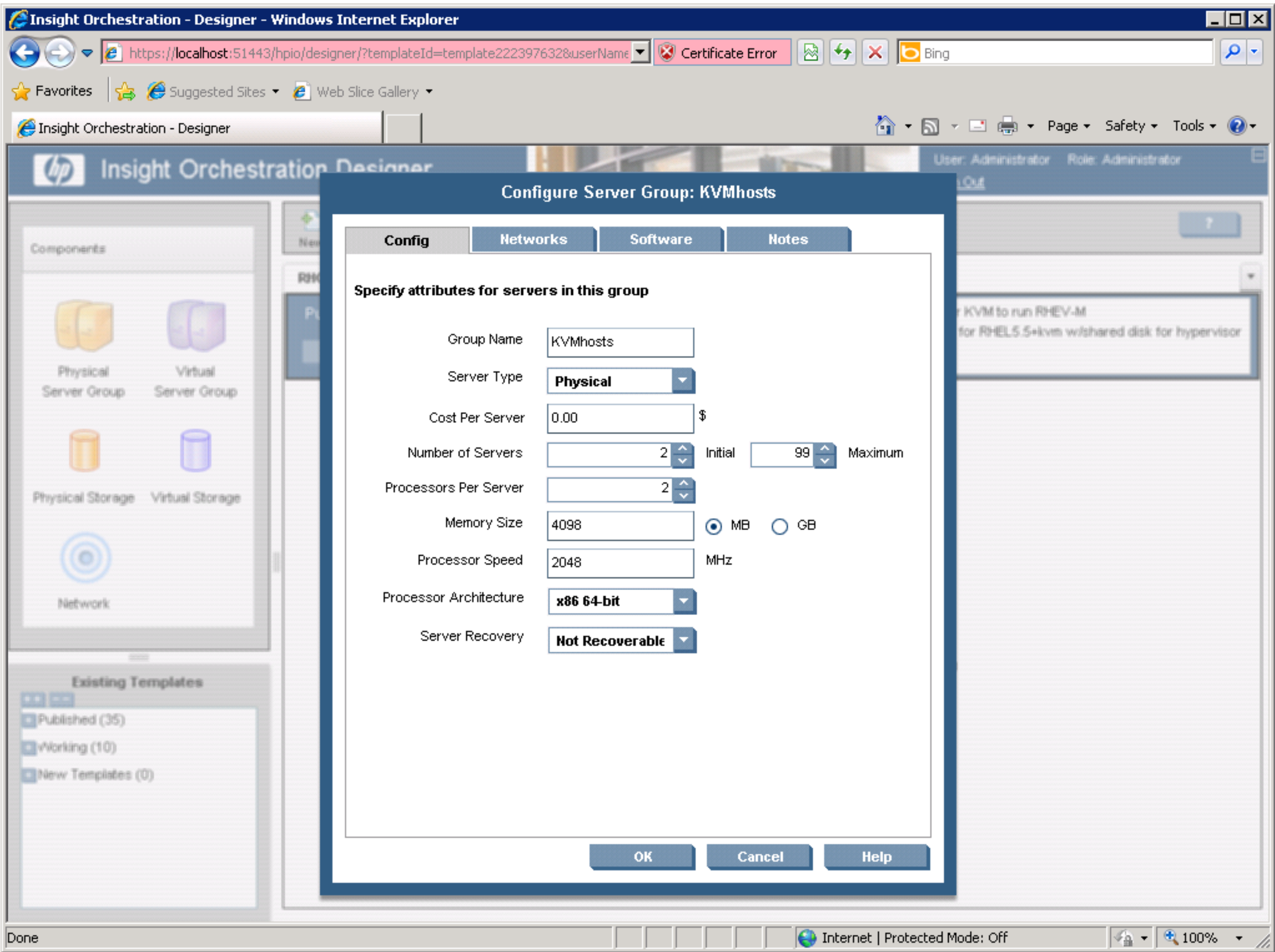

Figure 9: KVMhosts server group configuration.

5.3 Automating the Hypervisor Host Deployment

The process to take a bare-metal blade and built it to be a hypervisor host for cloud tenants is performed at least twice for the two blades in the proof-of-concept. In actuality, the process could potentially be repeated many times as the pool of hypervisor hosts is scaled out to meet the demands of the cloud's users. To increase efficiency, Matrix Operations Orchestration workflows are used in conjunction with the Red Hat Network system management platform to automate the software installation and configuration.

The process that should occur when a hypervisor host is deployed is as follows:

1. The Matrix environment selects an available blade that matches the criteria that was defined in the KVMhosts server group.
2. Red Hat Enterprise Linux 5 is installed on the blade by the Matrix.
3. The workflow registers the server with RHN using the previously-generated activation key, which identifies the server as a hypervisor host. The definition of the activation key tells RHN which additional software and configuration files are to be installed.
4. The KVM virtualization software, the RHEV-H management agents, and all necessary Red Hat Enterprise Linux updates are installed automatically from RHN, when the workflow instructs the server to do a software update through yum.
5. The workflow reboots the server when the update is complete.

5.3.1 RHN Preparation Steps

This section describes the steps that are performed using the hosted Red Hat Network to enable the hypervisor host build automation.

5.3.1.1 RHN System Group

Although not strictly required, a Red Hat Network system group was created for this proof-of-concept. All physical servers deployed in this proof-of-concept are members of the group. A system group provides a single, graphical status indicator in the RHN interface that alerts administrators when errata updates are available for any system in the group. It also provides a simple method of selecting all of the systems in the group for the RHN System Set Manager. Through the System Set Manager, an administrator can easily schedule actions for all of the systems in the group. These actions include:

- Installing additional packages or updating installed packages
- Managing software channel subscriptions
- Deploying configuration files and changing configuration channel subscriptions
- Running remote commands on all hosts in the system group

The steps to create the system group are:

- Log in to Red Hat Network, https://rhn.redhat.com.
- Click on “Systems” in the red bar near the top of the browser window.
- Click on “System Groups” in the gray box in the upper left of the window.
- Click on “create new group” near the upper right of the window.
- Fill in the Name and Description of the group. For example “hp-cloud-found” and “Red Hat Cloud, HP Edition”.

Appropriate users are assigned administrative rights to the new group:

- Click on the name of the new group.

- Click on “Admins” in the “tab bar” near the top of the browser window.
- Select any users that should have administrative rights to the new group.
- Click on the “Update” button in the bottom right of the window.

This enables those users to push updates and configuration files to the servers in the new group.

5.3.1.2 Hypervisor Host Configuration Files

A configuration channel is used to automatically deploy the necessary configuration files to the hypervisor servers. Red Hat Enterprise Virtualization requires that the RHEV Manager be added to the /etc/hosts file on each hypervisor host. The configuration channel that accomplishes this is created with the following steps:

- Click on “Configuration” in the red bar near the top of the browser window.
- Click on “Create a New Configuration Channel” in the “Configuration Actions” box on the right side of the window.
- Fill in the Name, Label, and Description fields. For example, “hp-cloud-found” for for all three fields.
- Click on “Add Files” in the “tab bar” near the top of the browser window.
- Click on “Create File” in the “sub-tab bar.”
- Enter “/etc/hosts” (without the quotation marks) in the Filename/Path field.
- Enter the desired contents of the file in the “File Contents” box.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain   localhost
::1                     localhost6.localdomain6 localhost6

192.168.100.200         rhevm.deploy.hp.local   rhevm
```

- Click on the “Create Configuration File” button at the bottom right of the window.

Note: For this proof-of-concept, the Matrix template used to install Red Hat Enterprise Linux systems disabled the *iptables* firewall. This is not recommended in a production environment. If iptables had been active, this configuration channel would have been used to deploy the necessary rules to the iptables configuration file, /etc/sysconfig/iptables.

5.3.1.3 Hypervisor Host RHN Activation Key

An activation key is created and used when the hypervisor hosts are registered with Red Hat Network to accomplish the following:

- Install the virtualization software, which includes Red Hat Enterprise KVM virtualization and RHEV virtualization management agent.

- Install RHN tools for Red Hat Enterprise Linux servers and some needed additional Red Hat Enterprise Linux packages.
- Subscribe the system to the configuration channel created in the step above.

The steps to create the activation key are:

- Click on “Systems” in the red bar near the top of the browser window.
- Click on “Activation Keys” in the gray box in the upper left of the window.
- Click on “create new key” near the upper right of the window.
- Fill in the Description field. For example, “hp-cloud-found”.
- Select “Provisioning” and “Virtualization Platform” add-on entitlements.
- Click on the “Create Key” button in the lower right of the window.
- Click on the “hp-cloud-found” key.
- Click on “Child Channels” in the “sub-tab bar.”
- Use control-click to select the following child channels:
    - Red Hat Network Tools for RHEL Server (v. 5 for 64-bit x86_64)
    - Red Hat Enterprise Virtualization Management Agent (v.5 for x86_64)
    - RHEL Virtualization (v. 5 for 64-bit x86_64)
- Click on the “Update Key” button in the lower right of the window.
- Click on “Configuration” in the “sub-tab bar.”
- Enter a 1 beside “hp-cloud-found.”
- Select “Schedule a deploy action for all of the configuration files in the channels listed above upon system registration.”
- Click on the “Update” button in the lower right of the window.
- Click on “Packages” in the “sub-tab bar.”
- Note that several packages have already been added. These are the packages required for configuration file management and deployment.
- Add the following packages to the text entry box, one per line:
    - bridge-utils
    - fence-agents
    - kvm-qemu-img
- Click on the “Update Key” button in the lower right of the window.
- Click on “Groups” in the “sub-tab bar.”
- Select “hp-cloud-found”.

- Click on the “Update Key” button in the lower right of the window.
- Click on “Activation Keys” in the gray box in the upper left of the window.
- Copy the generated 32-character key. You will need to paste it into the Matrix Operations Environment later.

5.3.2 Registration Workflow

The Matrix operating environment can easily be extended by developing Operations Orchestration (OO) workflows that are run when specific operations are performed. To automate the hypervisor host deployment, the workflow will be executed when a new service is deployed or more servers added to the server group KVMhosts.

The workflow is run after the Matrix Operating Environment has installed Red Hat Enterprise Linux on the hypervisor systems. It used *ssh* to perform the following steps:

- Configure the system to allow RHN configuration management. This must be performed before the system is registered with RHN, to enable configuration files to be deployed automatically:
  - `mkdir -p /etc/sysconfig/rhn/allowed-actions/script`
  - `mkdir -p /etc/sysconfig/rhn/allowed-actions/configfiles`
  - `touch /etc/sysconfig/rhn/allowed-actions/script/all`
  - `touch /etc/sysconfig/rhn/allowed-actions/configfiles/all`

- Register with the RHN using the previously-created RHN activation key:
  ```
  rhnreg_ks --activationkey=<32 character key noted above>
  ```

- Run yum update to install the latest packages and reboot:
  ```
  yum -y update
  reboot
  ```

In figure 10, the *RHNRegisterAndUpdate* workflow is shown in the OO Studio tool that is used to create and test workflows. Execution starts at the *RHCFFindIPAddress* step with the green outline. That step is a sub-workflow that returns a list of the IP addresses assigned to each of the newly deployed servers. It then goes to the *List Iterator* step, and loops for each IP address and calls the three ssh commands

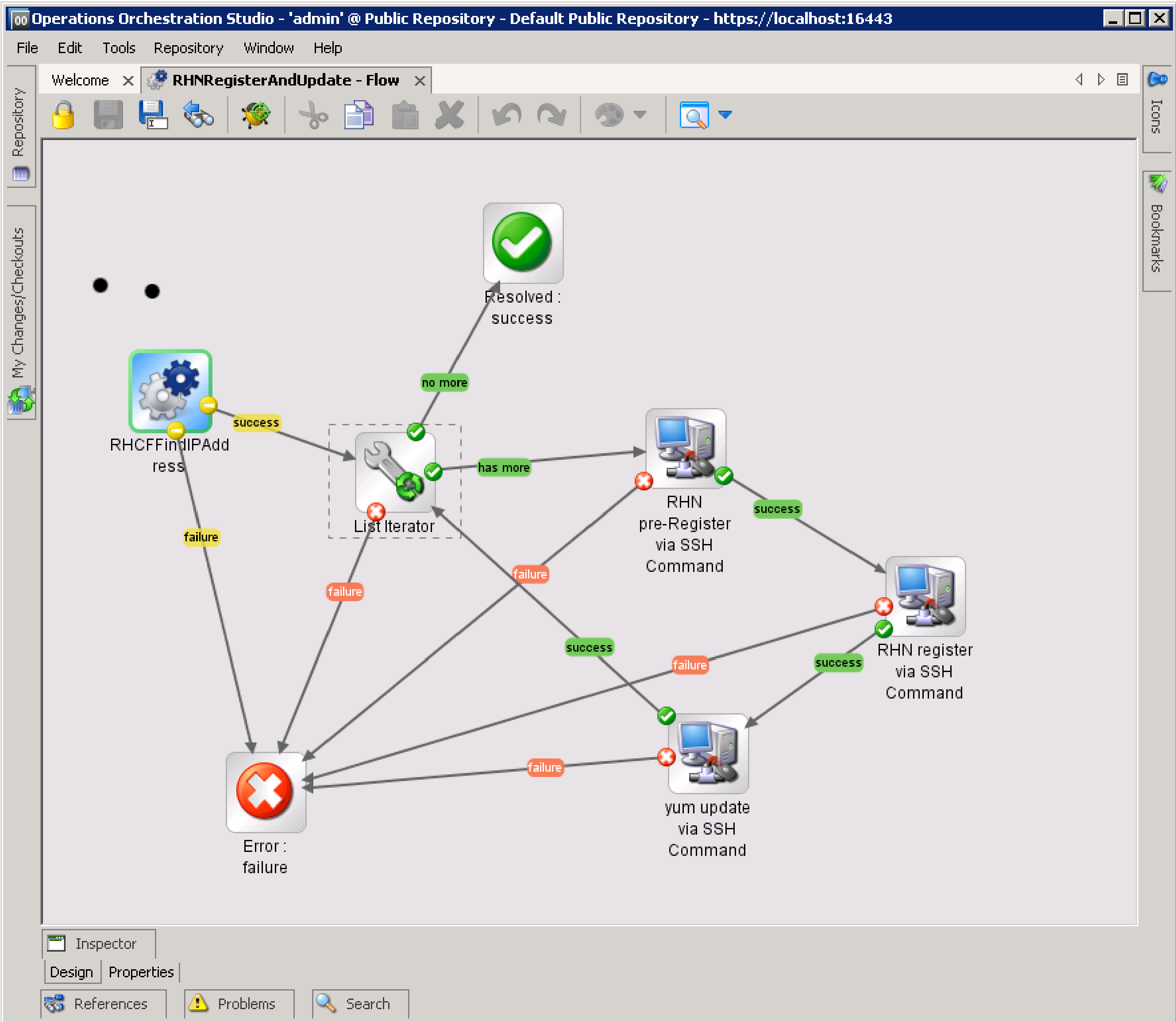

Figure 10: RHN Register and Update workflow.

Figure 11 shows the RHCFFindIPAddress sub-workflow, which is used to find the IP address assigned to the newly created server. The workflow engine provides all of the configuration details in an XML document. A *transform on* is performed on the input XML to obtain the list of IP addresses assigned, which are on the prod network. The XSL transform file *RHCFFindPrimaryIPAddressByLogicalSubnet.xsl* can be found in the *XSL.ZIP* file contained in the *RHCF-HP.ZIP* download. It needs to be placed in the *{$HpioConfDir}* folder on the Matrix management server. The default location is C:\Program Files\HP\Insight Orchestration\conf\OO\

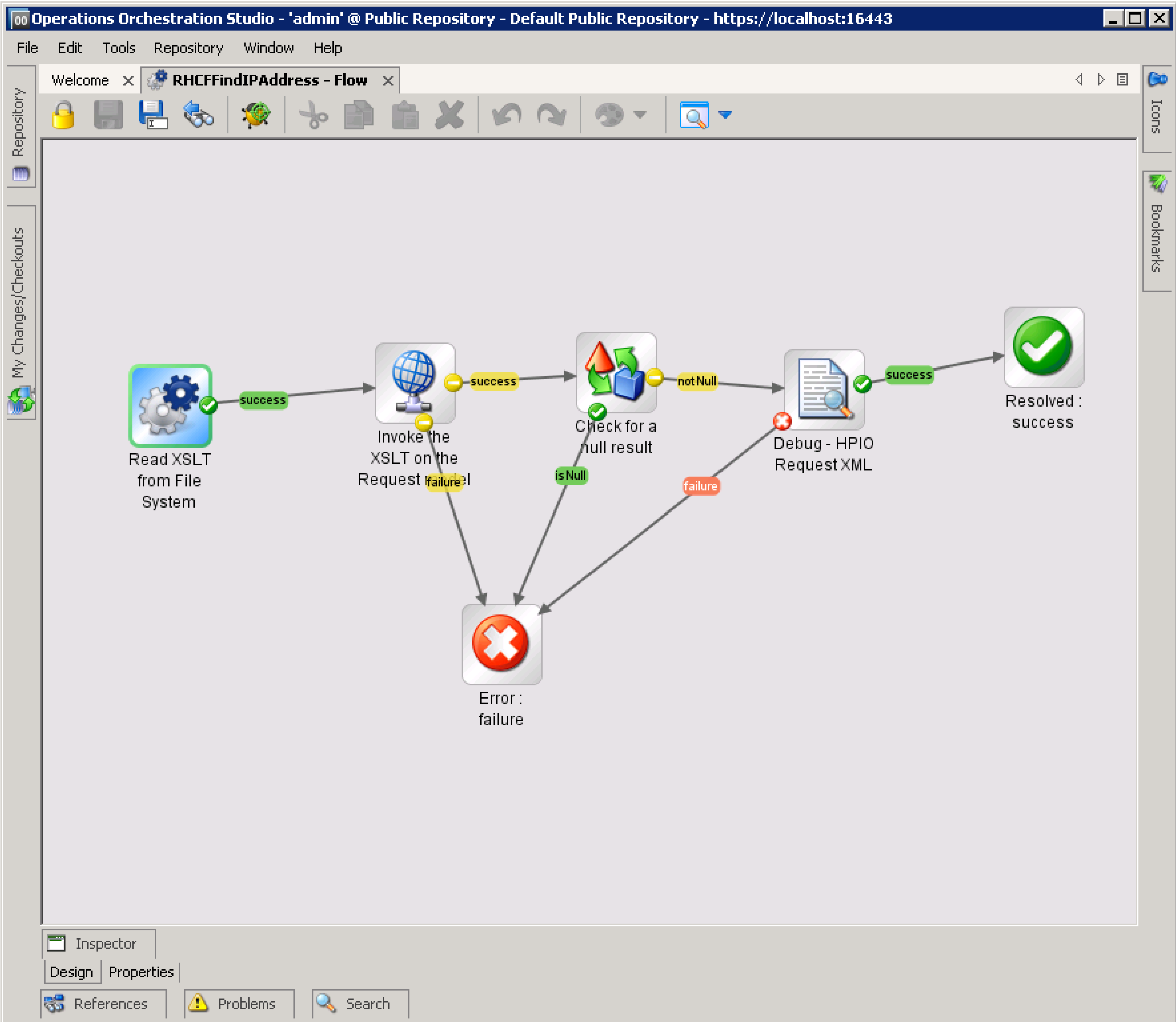

Figure 11: RHCF Find Primary IP Address by Subnet sub-workflow.

5.3.2.1 Import Workflow

The workflows are provided in the *WorkFlow.ZIP* archive contained in the *RHCF-HP.ZIP* download. To import the workflows into OO Studio:

1. Select “Repository” and then “Add Repository” as show in Figure 11.
2. Type in a “Repository Name” and select the location where you extracted the *WorkFlow.ZIP* file.
3. Click “OK” to open the repository in OO Studio.
4. Click on “Repository” and then “Set Target Repository” and select your “Default Public

Repository”.

5. Click on “Repository” again and then select “Publish source to Target – Preview”.
6. Click on the “Apply” icon. A confirmation message is displayed and workflows have been added to the default repository.

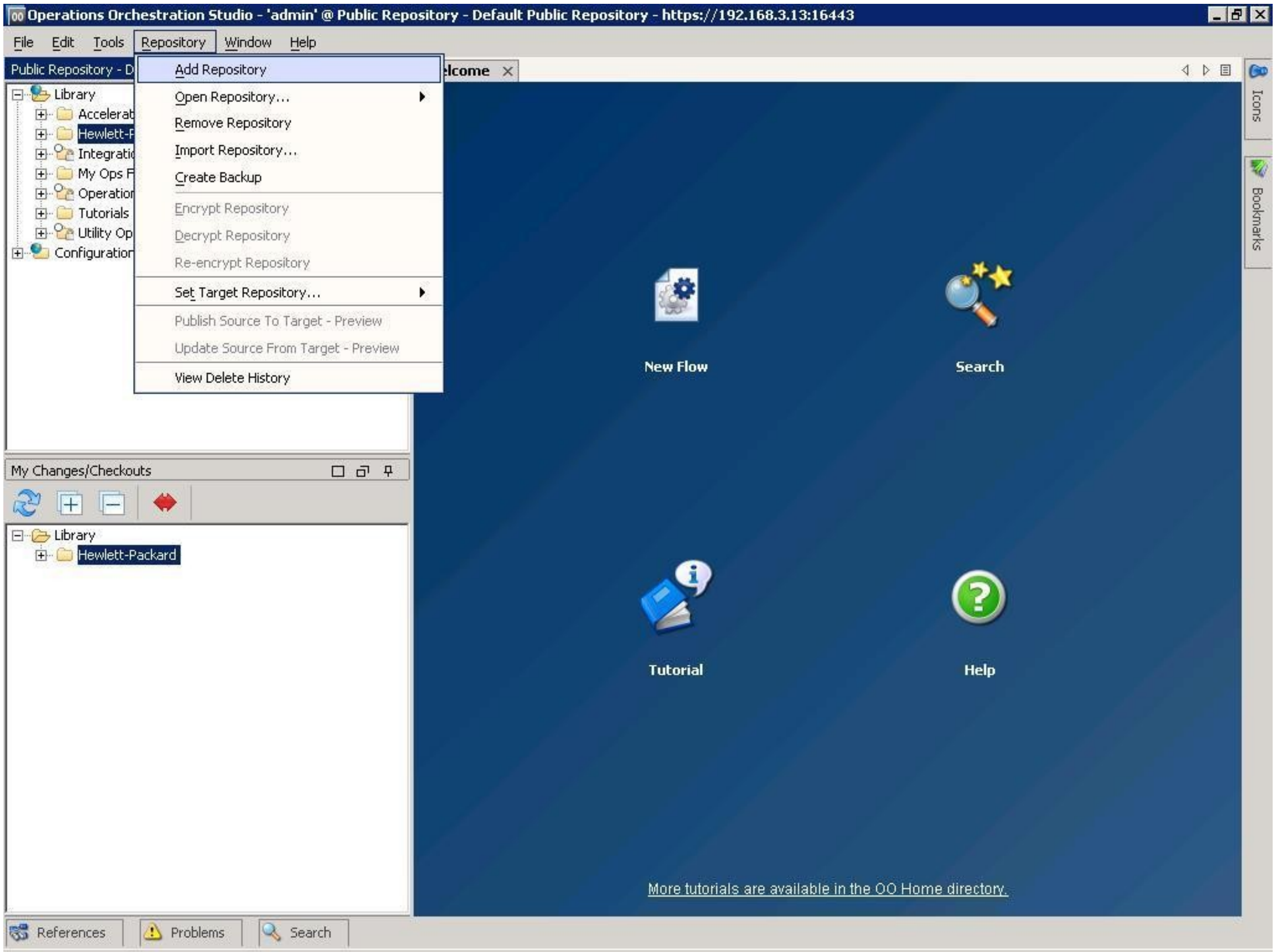

Figure 12 – Importing RH Workflows

There are three required global configuration settings that need to be manually created after importing the provided workflows. Open OO Studio, then open the repository to Configuration/ System *RHCFFindPrimaryIPAddressByLogicalSubnet.xsl* Properties and create the entries in Table 3, which need to be customized to your environment.

Table 3. Global configuration settings.

Configuration Name	Value	Function
RHCFkey	*<32 byte key>*	Red Hat Network activation defined earlier
RHCFuser	root	default root account of deployed servers

Configuration Name	Value	Function
RHCFpassword	*<password>*	default root password of deployed servers

5.3.3 Configure the Workflow to Run Automatically

The *RHNRegisterAndUpdate* workflow, which should now be located in /Library/Hewlett-Packard/Insight Orchestration/Service Actions/Red Hat Cloud Foundation/ needs to be configured to be called after the initial create service operation for the KVMhosts group and after new servers are added to the group. See figure 13.

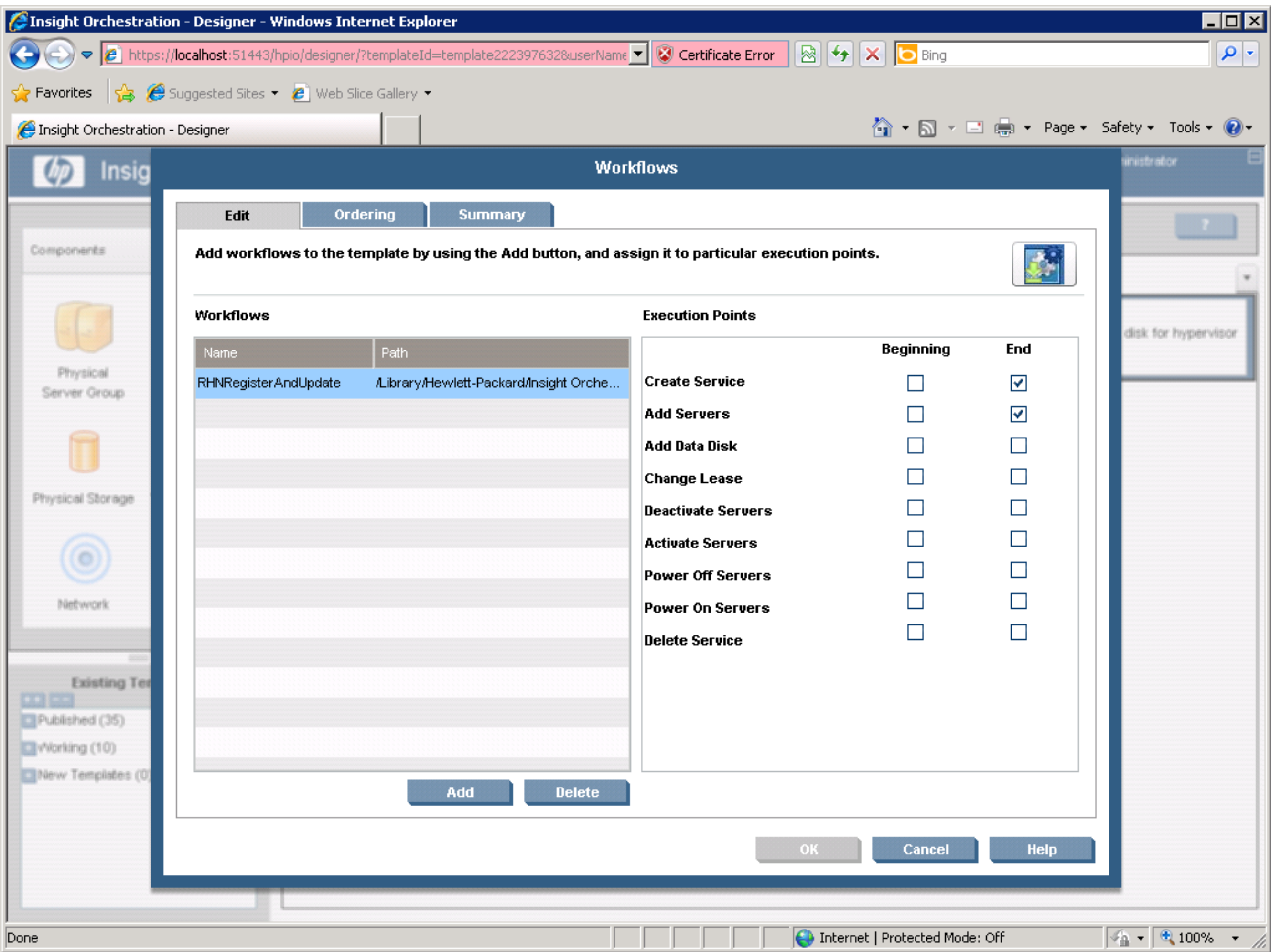

Figure 13: Defining when to execute RHN Register workflow.

5.4 Cloud Infrastructure Server Build

The cloud infrastructure server was initially deployed by the Matrix Operating Environment using a base Red Hat Enterprise Linux 5.5 template. Afterward, the following steps were performed to complete the configuration.

5.4.1 Red Hat Network Registration

Register the system with Red Hat Network. Using the activation key from section 6.2 ensured that the necessary add-on entitlements were applied to the system and it was subscribed to the required child channels. (It is also subscribed to the "Virtualization Management Agent" channel, which is not required but does not do any harm.)

1. Register the system:
   ```
   rhn_register
   ```

2. Update the system.
   ```
   yum -y update
   ```

3. Use the hosted RHN web interface to perform the following actions:
   1. Subscribe the system to required child channels
      1. RHEL Supplementary (v. 5 for 64-bit x86_64)
      2. RHEL Virtualization (v. 5 for 64-bit x86_64
   2. Add the system to the hp-cloud-found system group
   3. Add Management, Provisioning, and Virtualization Platform add-on entitlements

4. Install the high-performance "para-virtualized" disk and network drivers for KVM Windows guests.
   ```
   yum -y install virtio-win
   ```

5.4.2 Preparing for KVM Virtualization

1. Install the additional packages required for KVM virtualization.

   ```
   yum  -y groupinstall KVM
   ```

2. Prevent the libvirt default network from starting.

   ```
   rm -f /etc/libvirt/qemu/networks/autostart/*
   ```

3. Disable the netfilter firewall on virtual networks (bridges) by adding the following lines to /etc/sysctl.conf.

   ```
   net.bridge.bridge-nf-call-arptables = 0
   net.bridge.bridge-nf-call-ip6tables = 0
   net.bridge.bridge-nf-call-iptables = 0
   net.bridge.bridge-nf-filter-vlan-tagged = 0
   ```

4. Reboot the system.

```
reboot
```

When the system finishes rebooting, create a virtual network (bridge) for KVM guests. The virtual network is named br0, and it is connected to the eth0 physical interface. This process requires moving the host operating system's IP address from the physical interface (eth0) to the bridge (br0). Any error can cause a loss of network connectivity to the host, so it is useful to have console access (or ssh access through a different physical interface).

1. Create /etc/sysconfig/network-scripts/ifcfg-br0:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
```

2. Edit /etc/sysconfig/network-scripts/ifcfg-eth0. Remove the IP address information ("BOOTPROTO=dhcp" in this case) and add its association with the bridge:

```
# Broadcom Corporation NetXtreme II BCM57711e 10-Gigabit PCIe
DEVICE=eth0
HWADDR=00:21:51:9B:00:24
TYPE=ethernet
ONBOOT=yes
BRIDGE=br0
```

3. Restart the network.

```
service network restart
```

4. Verify the state of the network.

```
[root@pbNFS01 ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:21:5A:9B:00:24
          inet6 addr: fe80::221:5aff:fe9b:24/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25076400 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56681776 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12202384982 (11.3 GiB)  TX bytes:68779475862 (64.0 GiB)
          Interrupt:98 Memory:fb000000-fb7fffff

[root@pbNFS01 ~]# ifconfig br0
br0       Link encap:Ethernet  HWaddr 00:21:5A:9B:00:24
          inet addr:192.168.100.113  Bcast:192.168.103.255  Mask:255.255.252.0
          inet6 addr: fe80::221:5aff:fe9b:24/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17385150 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36279916 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9135602040 (8.5 GiB)  TX bytes:66947931424 (62.3 GiB)
```

```
[root@pbNFS01 ~]# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.00215a9b0024       no              eth0
```

This shows that eth0 does not have an IPv4 address assigned; it is now assigned to br0. eth0 is attached to br0.

5.4.3 NFS Share

An NFS share was created for use as the RHEV ISO library.

1. Create a directory to be exported.
   ```
   mkdir /NFS
   ```

2. SELinux was enabled in enforcing mode on the system, so the SELinux context of the shared directory had to be set appropriately. Rather than simply changing the context of the directory, it was added to the local SELinux policy, so that a future relabel keeps the correct context.
   ```
   semanage fcontext -a -t public_content_rw_t /NFS
   semanage fcontext -a -t public_content_rw_t '/NFS/.*'
   restorecon /NFS
   ```

3. Verify the context of the directory.

   ```
   [root@pbNFS01 ~]# ls -Zd /NFS
   drwxr-xr-x  root root system_u:object_r:public_content_rw_t /NFS
   ```

4. Create a subdirectory for the ISO library and set its ownership.

   ```
   mkdir /NFS/iso
   chown 36:36 /NFS/iso
   ```

5. Create /etc/exports:

   ```
   /NFS    192.168.100.0/22(rw,no_subtree_check)
   ```

6. Start NFS services and ensure that they are started when the system boots.

   ```
   service nfs start
   chkconfig nfs on
   ```

7. Verify the share.

   ```
   [root@pbNFS01 ~]# showmount -e localhost
   Export list for localhost:
   /NFS 192.168.100.0/22
   ```

5.5 RHEV Manager Virtual Machine Build

These steps create the KVM virtual machine for the Red Hat Enterprise Virtualization Manager.

5.5.1 Virtual Disk and Installation Media

Create a 40 GB pre-allocated disk image file for the RHEV-M guest.

```
dd if=/dev/zero of=/var/lib/libvirt/images/rhev-m.img \
bs=$((1024*1024)) count=$((1024*40))
```

Copy the Windows Server 2008 R2 installation ISO to the /var/lib/libvirt/images directory using scp.

5.5.2 Define the Guest

Use the *virt-manage*r tool to define the guest.

1. Start virt-manager:
   ```
   virt-manager
   ```
2. Double click on “localhost” to connect to the local KVM hypervisor, and click on the “New” button to create a new virtual machine.
3. Click on the “Forward” button.
4. Enter the name of the virtual machine, “rhev-m,” and click on the “Forward” button.
5. Leave all settings in the next step unchanged, and click on the “Forward” button.
6. Leave the installation media type set to “Local install media (ISO image or CDROM).
7. Set the “OS Type” to “Windows” and set the “OS Variant” to “Microsoft Windows 2008.”
8. Click on the “Forward” button.
9. Browse to or enter the ISO image location: /var/lib/libvirt/images/en_windows_server_2008_r2_standard_enterprise_datacenter_web_x64_dvd_x15-50365.iso
10. Click on the “Forward” button.
11. Set the “File (disk image)” field to the location of the virtual disk image file: /var/lib/libvirt/images/rhev-m.img.
12. Click on the “Forward” button.
13. Select “Shared physical device” as the guest network type. The “Device” should automatically be set to “eth0 (Bridge br0).”
14. Click on the “Forward” button.
15. Set the “Max memory (MB)” and “Startup memory (MB)” to 8192.
16. Set the “Virtual CPUs” to 4.
17. Click on the “Forward” button.
18. Review the virtual machine parameters.
19. Click on the “Finish” button. The guest will start.
20. As soon as the guest console appears, select Virtual Machine -> Shut Down -> Force Off from the console menu.
21. Click on the “Yes” button in the confirmation dialog.

5.5.3 Edit the Guest

A number of virtual machine parameters are changed before the actual guest operating system installation using these steps

1. Change the virtual disk type from IDE to VirtIO.
   1. Select the “Hardware” tab.
   2. Select “Disk hda” on the left wide of the window.
   3. Click on the “Remove” button.
   4. Click on the “Add Hardware” button.
   5. Select “Storage” as the hardware type and click the “Forward” button.
   6. Select “File (disk image)” and enter the path to the virtual disk image file: /var/lib/libvirt/images/rhev-m.img.
   7. Change the device type to “Virtio Disk.”
   8. Click on the “Forward” button and then click on the “Finish” button.

2. Select “Disk hdc” and verify that the Windows Server 2008 R2 ISO file is still “attached” to the virtual machine's optical drive. If it is not, click on the “Connect” button and browse to the ISO file to reconnect it.
3. Change the virtual NIC type to VirtIO.
   1. Select “NIC :XX:YY:ZZ.” (“:XX:YY:ZZ” represents the last three octets of the virtual NIC'S randomly generated MAC address.)
   2. Click on the “Remove” button.
   3. Click on the “Add Hardware” button.
   4. Select “Network” as the hardware type and click the “Forward” button.
   5. Select “Shared physical device” as the guest network type. The “Device” should automatically be set to “eth0 (Bridge br0).”
   6. Change the “Device Model” to “virtio.”
   7. Click on the “Forward” button and then click on the “Finish” button.

4. Attach the virtual floppy disk image containing the VirtIO storage and network drivers to the virtual machine.
   1. Click on the “Add Hardware” button.
   2. Select “Storage” as the hardware type and click the “Forward” button.
   3. Select “File (disk image)” and browse to or enter the location of the virtual floppy disk image: /usr/share/virtio-win/virtio-drivers.vfd.
   4. Change the device type to “Floppy disk.”
   5. Click on the “Forward” button and then click on the “Finish” button.

5. Set the boot options.
   1. Select “Boot Options” in the left side of the window.
   2. Select “Start virtual machine on host boot up.”
   3. Ensure that “Boot Device” is set to “Hard Disk.”
   4. Click on the “Apply” button.

The virtual machine must be set to boot from the hard disk, so that its BIOS will mark the disk as bootable. Even after the proper device driver is loaded, Windows will refuse to install to a disk that is not marked as bootable.

5.5.4 Install the Windows Operating System

By default, the virtual machine is configured to boot from its hard disk, so it is necessary to use the BIOS boot menu to boot from the installation media.

1. Select the “Console” tab.
2. Click on the start button (which looks like the play button from a tape recorder).
3. Ensure that the mouse pointer is within the console area and press F12 until “Select boot device:” appears in the console.
4. Press “3” to boot from the guest's CD-ROM and begin the Windows Server 2008 R2 installation.
5. Leave the language, time and currency, and keyboard settings unchanged and click on the “Next” button.
6. Click on “Install now.”
7. Leave the operating system selection unchanged — “Windows Server 2008 R2 Standard (Full Installation)” — and click on the “Next” button.
8. Select “I accept the license terms” and click on the “Next” button.
9. Click on “Custom (advanced)” to create a fresh Windows installation.
10. The “Where do you want to install Windows?” screen will appear, along with a warning that no drives were found. Click on “Load Driver.”
11. Click on the “OK” button.
12. Select “Red VirtIO SCSI controller (A:\amd64\Win2008\viostor.inf).”
13. Click on the “Next” button.
14. The “Where do you want to install Windows?” screen reappears, with the newly detected disk selected. Click on the “Next” button.

The remainder of the Windows installation is performed in a normal manner. After installation, the VirtIO network driver is installed.

1. Click on the “Start” button.
2. Right-click on “Computer” and select “Properties” from the pop-up menu.
3. Click on “Device Manager” in the upper left of the window.
4. The Device Manager window opens. The Ethernet adapter is shown under “Other devices,” along with a symbol indicating that it is not functional.
5. Right-click on “Ethernet Controller” and select “Update Driver Software...” from the pop-up menu.
6. Click on “Browse my computer for driver software.”
7. Enter or browse to “A:\amd64\Win2008” and click on the “Next” button.
8. Windows will install the driver. Click on the “Close” button.
9. The Ethernet adapter now appears as a “Red Hat VirtIO Ethernet Adapter,” and it can be configured through the Windows Control Panel.

Windows is then configured in a normal manner:

1. Activate Windows.
2. Set the time zone.
3. Set the computer name and DNS suffix.
4. Enable remote desktop connections.
5. Run Windows Update repeatedly until no additional updates are available. (“Microsoft .NET Framework 4” is **not** installed, as it can interfere with proper functioning of the RHEV Manager.)

Prepared the system for RHEV Manager installation, as documented in section 3.1 of the *Red Hat Enterprise Virtualization for Servers 2.2 Installation Guide*.

- Because Active Directory is not used, a local Windows user called “rhevadmin” is created; this user does not require any special operating system privileges.
- For purposes of this proof-of-concept, the Windows firewall is disabled. In a production environment, only required ports should be left open. (See https://access.redhat.com/kb/docs/DOC-45684.)

5.5.5 Red Hat Enterprise Virtualization Manager

Download the Red Hat Enterprise Virtualization Manager from Red Hat Network and install as described in section 6.9.3 of *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds*.

- Select Local authentication. The rhevadmin ID created in the previous section is used.

5.6 Completing the Build

Install the remaining components of the reference architecture as detailed in *Red Hat Cloud Foundations Reference Architecture: Private IaaS Clouds*. Only significant differences (those other than IP addresses, hostnames, and user IDs are noted here).

5.6.1 Hypervisor Servers

The hypervisor servers are deployed, updated, and configured automatically by the Matrix Operating Environment. It was only necessary to add them to the RHEV environment. See step 5 in section 7.4 of *RHCF-RA* (or the *Red Hat Enterprise Virtualization for Servers 2.2 Installation Guide*).

5.6.2 RHEV Storage and Datacenter

Configure the RHEV storage domains, datacenter, and cluster as described in section 6.9.4 of *RHCF-RA*. In the test proof-of-concept, the path used for the ISO library was 192.168.100.113:/NFS/iso. The 300 GB shared LUN allocated by the Matrix Operating Environment was used for the FCP storage pool.

5.6.3 Red Hat Network Satellite

Red Hat Network Satellite was installed as a RHEV guest, rather than the KVM guest shown in *RHCF-RA*.

1. Download Red Hat Enterprise Linux 5.5 and Red Hat Network Satellite 5.4 (embedded database) installation media from RHN to the RHEV Manager virtual machine.
2. Follow the steps in section 7.5.2 of RHCF-RA to create a virtual machine with 2 virtual CPUs, 4 GB of memory, and a 150 GB preallocated virtual disk.
3. Start the virtual machine and install Red Hat Enterprise Linux 5.5. (VNC is the recommended installation method when installing Red Hat Enterprise Linux in virtual machines.)
4. Choose “Customize now”,so that unneeded software packages groups can be “unselected” during installation.
5. Unselect all package groups other than Base System -> Base on the next screen.
6. Immediately after installation, remove all 32-bit packages:

```
yum remove '*.i?86'
```

7. Open the ports necessary for Satellite operation in the iptables firewall.

```
iptables -I RH-Firewall-1-INPUT 10 -p tcp -m state --state NEW --dport 5222 -j ACCEPT
iptables -I RH-Firewall-1-INPUT 10 -p tcp -m state --state NEW --dport 443 -j ACCEPT
iptables -I RH-Firewall-1-INPUT 10 -p tcp -m state --state NEW --dport 80 -j ACCEPT
service iptables save
```

8. Perform the remainder of the Satellite installation in accordance with section 6.3.4 of RHCF-RA.

6 Opportunities for Further Integration

The proof-of-concept described in this paper focuses on deploying the core components of the Red Hat Cloud Foundations reference architecture in the HP CloudSystem Matrix environment. This section briefly discusses some potential integration that can performed to further streamline management of the cloud infrastructure.

6.1 Automatic Hypervisor Registration

Currently, a Red Hat Enterprise Virtualization administrator must manually register all Red Hat Enterprise Linux hypervisors with the RHEV Manager. Section 5.3 of this document describes the Matrix workflow that is created to prepare newly deployed Red Hat Enterprise Linux systems for use as hypervisors. It would be convenient if this workflow could be extended to automatically inform the RHEV Manager of the new hypervisor and initiate the registration process.

This is expected to be possible with the REST API that will be supported in the next release of Red Hat Enterprise Virtualization. Information about the capabilities of the API is available at https://fedorahosted.org/rhevm-api/.

6.2 Virtual Machine Management from the Matrix Operating Environment

The Red Hat Enterprise Virtualization REST API provides many capabilities beyond simply registering new hypervisors. It can be used to create, delete, start, stop, pause, unpause, and migrate virtual machines, as well as managing virtual machine snapshots. By leveraging these capabilities, the Matrix Operating Environment could be extended to offer a “single pane of glass” for managing both physical and virtual resources.

7 Downloads

The HP Matrix Operations Orchestration template and workflow files that were developed for the proof of concept are available for download from http://redhat.com/hpopencloud.

7.1.1 Contents of RHCF-HP.zip

The RHCF-HP.ZIP file contains the following files:

Filename	Description
README.txt	Additional information
Template.zip	Contains RHCF-RefArch.xml to be imported into HP Insight Orchestration
WorkFlow.zip	Contains workflows to be imported into HP Operations Orchestration, embedded edition
XSL.zip	Contains RHCFFindPrimaryIPAddressByLogicalSubnet.xsl to be placed in C:\Program Files\HP\Insight Orchestration\conf\OO\ on the Matrix management server