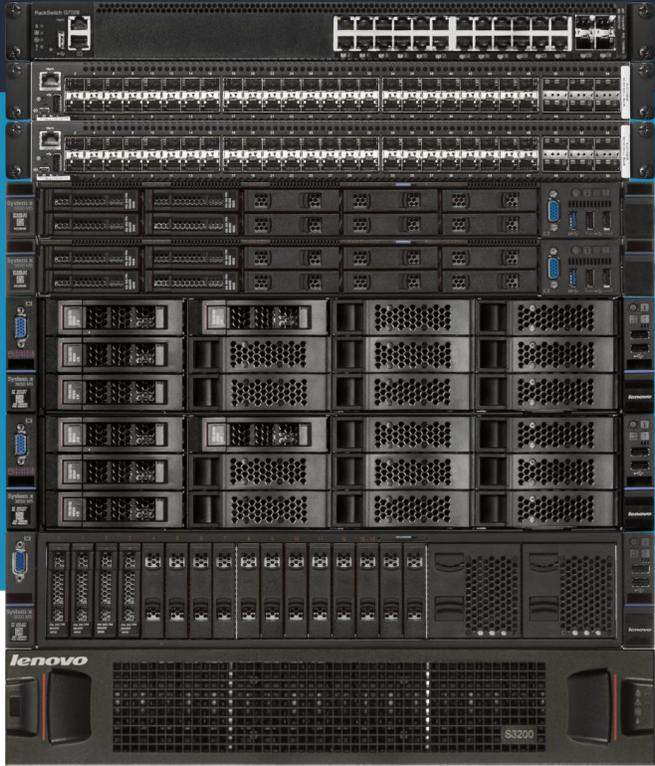


Red Hat® Enterprise Linux® OpenStack® Platform on **Lenovo® Performance Rack Servers**



A detailed guide to help
you deploy and manage
your private cloud solution



A Principled Technologies deployment guide commissioned by Lenovo. (Revised)

TABLE OF CONTENTS

Executive summary	3
Lenovo + Red Hat + OpenStack	3
Hardware summary	4
Software summary.....	5
Hardware in this solution	6
The Lenovo System x3650 M5 and System x3550 M5 rack servers ...	6
The Intel Xeon processor E5-2600 v3 product family.....	8
The Lenovo RackSwitch G8272 switch.....	8
The Lenovo RackSwitch G7028 switch.....	9
The Lenovo Storage S3200 SAN storage appliance	9
Software in this solution	10
Lenovo XClarity Administrator.....	10
Red Hat Enterprise Linux OpenStack Platform	10
Red Hat CloudForms	15
Planning and deploying your private cloud	16
After the install – Working with Red Hat Enterprise Linux OpenStack Platform, Red Hat CloudForms, and Lenovo XClarity Administrator	25
Validating the OpenStack deployment	31
Summary and conclusions	38
Appendix A – Installing the Lenovo hardware: Network setup	39
Appendix B – Deploying Red Hat Enterprise Linux OpenStack Platform Director 7	42
Task 1 – Deploying the OS to the Red Hat Enterprise Linux OpenStack Platform Director.....	42
Task 2 – Configuring the Lenovo System x3550 M5 Performance Rack Server for Red Hat Enterprise Linux OpenStack Platform Director installation	42
Task 3 – Installing Red Hat Enterprise Linux OpenStack Platform Director	45
Appendix C – Registering nodes	47
Example: instackenv.json.....	47
Appendix D – Assigning deployment roles	49
Appendix E – Configuring and deploying the Overcloud	50
Appendix F – Finalizing the Operational cloud	53
About Principled Technologies	55

EXECUTIVE SUMMARY

Businesses are embracing private cloud solutions in their environments for many reasons. Datacenter resources, such as Infrastructure-as-a-Service, are offered and shared among a large number of users demanding IT services in the private cloud. The users can then access and manage resources through online portals. One of the key reasons to implement a private cloud is scalability—the ability to add and adjust resource growth to meet user growth and subsequent demand.

It's even easier to have a private cloud for your business now that Lenovo and Red Hat have collaborated to offer their hardware and software together. In the Principled Technologies datacenter, we used the Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform to deploy and manage a private cloud. We found the Lenovo System x3650 M5 and x3550 M5 rack servers, powered by the Intel® Xeon® processor E5-2600 v3 family, can serve as a solid foundation for the customized Red Hat Enterprise Linux OpenStack Platform private cloud environment of your business.

This guide will show you how to prepare, deploy, provision, and manage a Red Hat Enterprise Linux OpenStack Platform 7 private cloud environment managed by XClarity Administrator and built on two-socket Lenovo System x3650 M5 and x3550 M5 rack servers.¹

LENOVO + RED HAT + OPENSTACK

OpenStack has evolved from an open-source project and is now a mature and constantly evolving cloud platform. It combines compute, advanced software-defined networking, block storage, object storage, and other components. Red Hat and OpenStack have an established and developed relationship, as Red Hat Enterprise Linux OpenStack Platform 7 is the fifth generation of OpenStack distribution from Red Hat. This latest version has features to address your business' crucial OpenStack requirements: deployment and management of OpenStack. Red Hat Enterprise Linux OpenStack Platform 7 offers Red Hat Enterprise Linux OpenStack Director, an appliance used for deploying OpenStack, which we deployed using Lenovo Performance Rack Servers with Lenovo XClarity Administrator.

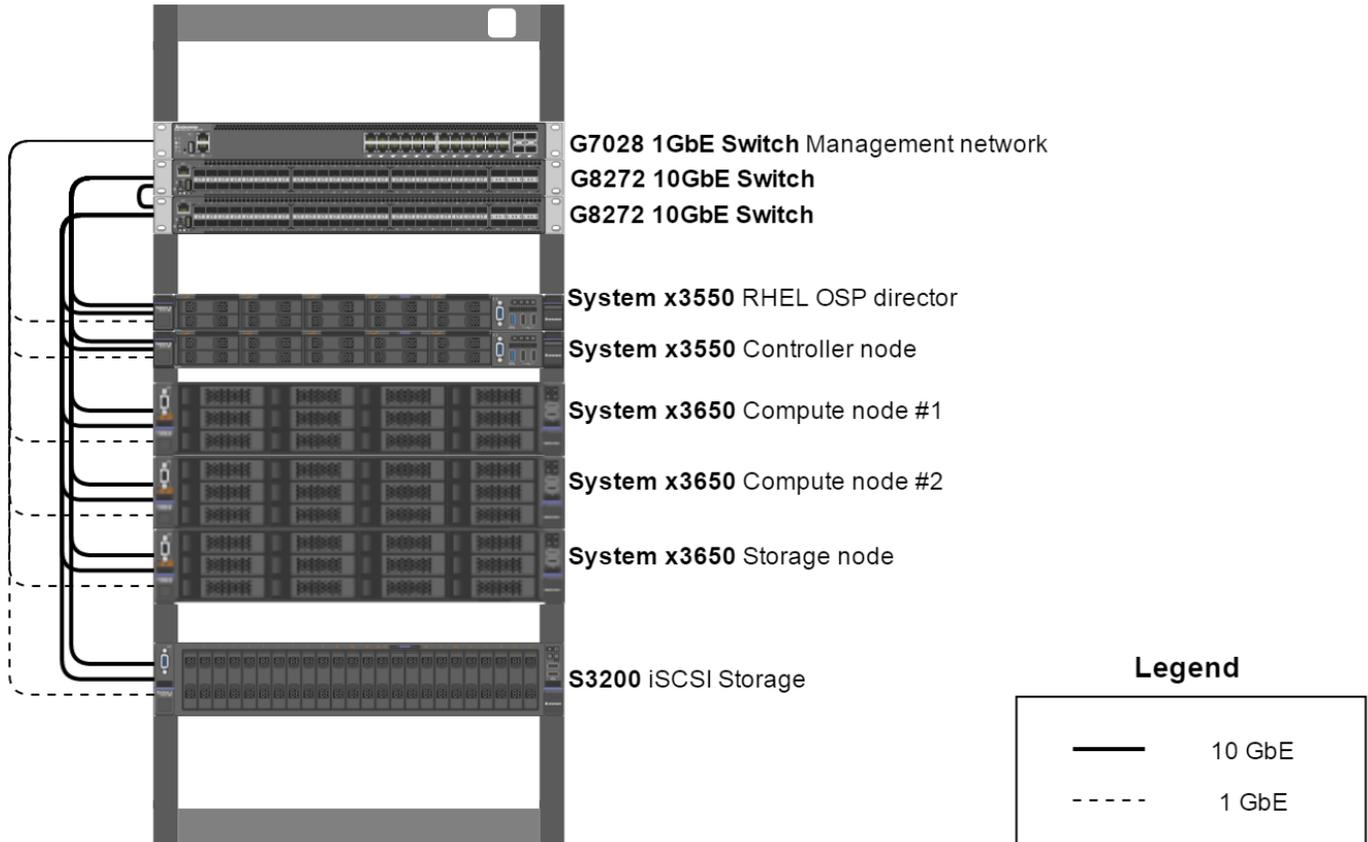
¹ For detailed information on our deployment process, see [Appendix B](#).

Hardware summary

Figure 1 shows the components of our solution. Our environment included the following hardware:²

- **Lenovo System x3650 M5 two-socket server** with versatile storage in 2U of space for a wide range of workloads
- **Lenovo System x3550 M5 rack servers**, a 1U, two-socket server that can handle complex workloads, including Big Data and virtualization
- **Two top-of-rack Lenovo RackSwitch G8272 switches** for 72 SFP+ 10Gb Ethernet ports to improve your infrastructure’s reliability
- **One top-of-rack Lenovo RackSwitch G7028 switch** as a management switch, offering 24 GbE ports and four 10 GbE ports
- **One Lenovo Storage S3200** appliance for iSCSI storage

See [Hardware in this solution](#) for more details.



Lenovo Red Hat Enterprise Linux OpenStack Platform 7 test bed

Figure 1: Our Red Hat Enterprise Linux OpenStack Platform 7 deployment environment.

² The Lenovo S3200 was part of the hardware solution but wasn't used in this study.

Software summary

Our environment included the following software:

- **Lenovo XClarity Administrator** to deploy the operating system for the Red Hat Enterprise Linux OpenStack Director and manage datacenter resources with a simple and automated process
- **Red Hat Enterprise Linux OpenStack Director** to deploy the Red Hat Enterprise Linux private cloud
- **Red Hat Enterprise Linux OpenStack Platform 7**, which uses features from the OpenStack software set including Controller, Compute, and Cinder to form the Overcloud
- **Red Hat CloudForms®**, a virtual appliance with management services for OpenStack components

Lenovo XClarity Administrator overview

Lenovo XClarity Administrator is a hardware resource-management solution that can simplify and automate infrastructure tasks. It's designed to integrate with Lenovo x86 rack servers, Lenovo High End servers, and the Flex System converged infrastructure platform including the Flex System Chassis Management Module, x86 compute nodes, and I/O modules. Lenovo XClarity Administrator is installed as a virtual appliance and features:

- A web-based graphical user interface (GUI)
- Automated firmware and configuration management
- The ability to deploy operating systems and hypervisors to bare-metal servers and compute nodes
- Support for integration into various external or higher level management, automation, and orchestration tools through Representational State Transfer (REST) Application Program Interfaces (APIs); and
- Control over hardware resources through scripting and commands by using Microsoft® Windows® PowerShell®

Red Hat Enterprise Linux OpenStack Platform 7 overview

Red Hat Enterprise Linux OpenStack Platform 7 is an Infrastructure-as-a-Service (IaaS) cloud software solution based on the OpenStack community "Kilo" release. Red Hat Enterprise Linux OpenStack Platform 7 can be used to build a private, public, or hybrid cloud.³

Version 7 includes several new features including simplified deployment and management through Red Hat Enterprise Linux OpenStack Platform Director, high

³ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/preface

availability, granular security control over network traffic, flexibility from IPv6 enhancements, and support for snapshot-based backups.

Red Hat CloudForms overview

Red Hat CloudForms exists as a virtual appliance that can be deployed on Red Hat Enterprise Linux OpenStack Platform, Red Hat Enterprise Virtualization, or VMware® vSphere®. Once deployed, management capabilities are provided through the Red Hat CloudForms Management Engine Console, a web interface that supports browsers including Mozilla® Firefox®, Internet Explorer® 8 or higher, and Google Chrome™ for Work.⁴

HARDWARE IN THIS SOLUTION

The Lenovo System x3650 M5 and System x3550 M5 rack servers

There are a number of advantages in choosing Lenovo System x3650 M5 and x3550 M5 servers with Intel Xeon E5-2600 v3 processors and XClarity Administrator to deploy and manage your Red Hat Enterprise Linux OpenStack Platform 7 cloud environment:

1. Computing capacity for your Software-Defined Infrastructure (SDI) from the Intel® Xeon® E5-2600 v3 processor family
2. Combined compute and storage hardware that allows virtualized pooling and customization for your specific workload needs
3. During runtime, resized compute and storage resources offer scalability as needed
4. Management and monitoring for kernel-based virtual machines (KVMs)

As seen in Figure 2, the Lenovo System x3650 M5 is a 2U, two-socket rack server designed with versatile storage configurations. Among the suggested uses for these servers are database, cloud computing and virtualization, enterprise applications, collaboration/email, business analytics and Big Data, and Microsoft RemoteFX® applications.⁵

The Lenovo System x3550 M5 (Figure 3) is a 1U, two-socket rack server available in many storage configurations to handle workloads ranging from web server to Big Data.

The Lenovo Performance Rack Servers come equipped with features designed to minimize your infrastructure costs while potentially delivering effective performance for your demanding workloads. For example, their dual-fan zones were designed to allow the system to continue operations in environments that reach up to 40 degrees Celsius.

⁴ access.redhat.com/documentation/en-US/Red_Hat_CloudForms/3.2/html/Deployment_Planning_Guide/index.html

⁵ www.redbooks.ibm.com/technotes/tips1193.pdf, page 1

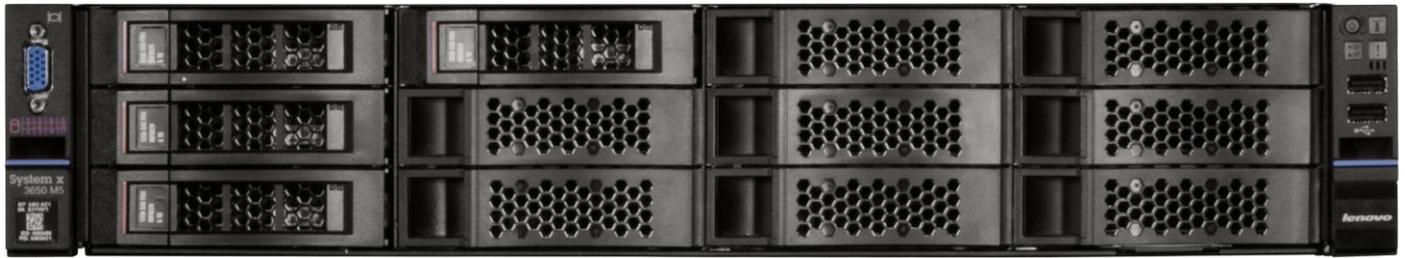


Figure 2: The Lenovo System x3650 M5 server with 3.5-inch drive bays.



Figure 3: The Lenovo System x3550 M5 server with 2.5-inch drive bays.

To support your environment's compute needs, each Performance Rack Server is configured with two Intel Xeon processor E5-2600 v3 CPUs for a maximum of 36 cores, and 72 threads, per server. Performance Rack Servers can achieve desirable memory performance by offering configurations that support two RDIMMs per channel.⁶ In addition to RDIMM, the servers support SK Hynix 64 GB TruDDR4™ Memory LRDIMMs. For your networking requirements, each server also offers four integrated 1Gb Ethernet ports with optional 10Gb/40Gb ports for faster networking.⁷

The two Performance Rack Servers differ primarily in their available storage options. The Lenovo System x3650 M5 server has several external drive configurations to deliver storage resources including:

- up to 26 2.5" hot-swap SAS/SATA bays
- up to 14 3.5" and two 2.5" hot-swap SAS/SATA drive bays
- up to 16 2.5" Simple Swap SATA bays
- up to 8 3.5" Simple Swap SATA bays

The x3550 M5 has up to 10 front and two rear 2.5-inch HDD or SSD drive bays, and up to four 3.5-inch HDD bays. You can configure the x3650 M5 server with up to 100 TB of internal storage and can configure the x3550 M5 server with up to 46 TB of internal storage.

For your I/O-intensive applications, the x3650 M5 offers a 12Gbps SAS RAID controller for a faster data transfer rate than the more common 6Gb SAS solutions.⁸ The

⁶ www.redbooks.ibm.com/technotes/tips1193.pdf, page 1

⁷ www.redbooks.ibm.com/technotes/tips1193.pdf, page 2

⁸ www.redbooks.ibm.com/technotes/tips1193.pdf, page 2

x3550 M5 offers a 12 Gbps SAS RAID controller with support for hardware RAID-0, -1, and -10.

The Intel Xeon processor E5-2600 v3 product family

According to Intel, the Intel Xeon processor E5-2600 v3 product family “helps IT address the growing demands placed on infrastructure, from supporting business growth to enabling new services faster, delivering new applications in the enterprise, technical computing, communications, storage, and cloud.” It can deliver benefits in performance, power efficiency, virtualization, and security.

The E5-2600 v3 product family has up to 50 percent more cores and cache than processors from the previous generation. Other features include the following:

- Intel Advanced Vector Extensions 2 (AVX2)
- Intel Quick Path Interconnect link
- Up to 18 cores and 36 threads per socket
- Up to 45 MB of last-level cache
- Up to 1.5 TB of next-generation DDR4 memory support
- Intel Integrated I/O providing up to 80 PCIe® lanes per two-socket server
- Intel AES-NI data encryption/decryption

The Intel Xeon processor E5-2600 v3 product family also uses Intel Intelligent Power technology and Per-core P states to maximize energy efficiency.

Learn more at www.intel.com/content/www/us/en/processors/xeon/xeon-e5-brief.html.

The Lenovo RackSwitch G8272 switch

The Top-of-Rack (ToR) 1U Lenovo RackSwitch G8272 offers 48 SFP+ 10 Gigabit Ethernet (GbE) ports and six QSFP+ 40 GbE ports (each 40 GbE port can be split into four additional 10 GbE ports). In addition, the SFP+ ports can support 1GbE. With redundant power supplies and cooling fans, the RackSwitch G8272 can add or improve reliability in your infrastructure.

For your private cloud environment, the Lenovo RackSwitch G8272 features unified fabric port (UFP) for use with virtual NICS and OpenFlow to help create software-defined virtual networks (SDN). The RackSwitch G8272 also supports VXLAN for network virtualization through overlays. Figure 4 shows our network switch from testing.

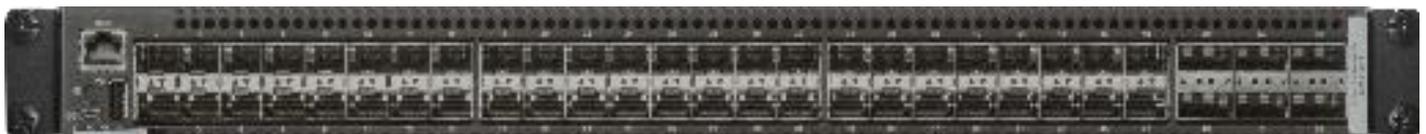


Figure 4: The Lenovo RackSwitch G8272.

The Lenovo RackSwitch G7028 switch

The entry-level, Top-of-Rack (ToR) 1U Lenovo RackSwitch G7028 offers four SFP+ 10 Gigabit Ethernet (GbE) ports and 24 10/100/1000BASE-T RJ-45 GbE ports. In addition, the SFP+ ports can support 1GbE. Its rear-to-front airflow allows for flexible mounting in a rack and can help deliver savings in cooling costs. Figure 5 shows our network switch from testing.



Figure 5: The Lenovo RackSwitch G7028.

The Lenovo Storage S3200 SAN storage appliance

Figure 6 shows the Lenovo Storage S3200 appliance. The Lenovo Storage S3200 array offers hybrid configurations to run in many environments, real-time tiering through flash hybrid storage for your demanding workloads, a simple GUI, thin provisioning, the ability for virtual snapshots, high availability configurations, and more. Designed to bring enterprise-level performance to any datacenter, the Lenovo Storage S3200 can meet the requirements for Red Hat Enterprise Linux OpenStack Platform 7 while supporting business growth, though we did not use it in this study.

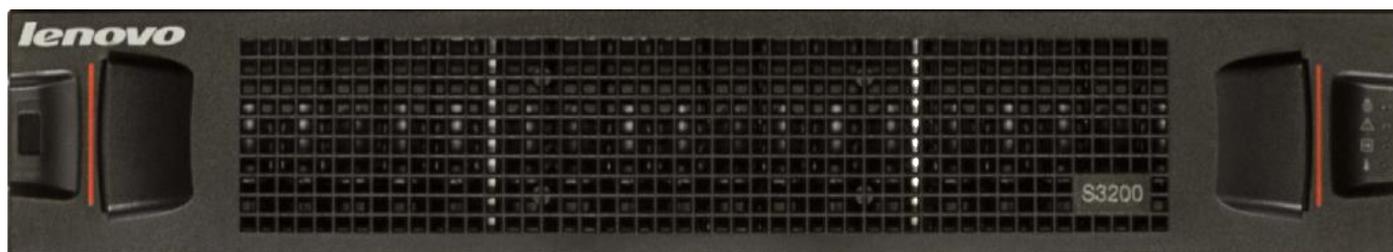


Figure 6: The Lenovo Storage S3200 SAN storage appliance.

SOFTWARE IN THIS SOLUTION

Lenovo XClarity Administrator

Lenovo XClarity Administrator is a hardware resource management solution for Lenovo Performance Rack Servers. Having the right management tools ensures the individual components of your Red Hat Enterprise Linux OpenStack Platform can deploy and run smoothly.

XClarity Administrator offers out-of-band management, which means managed Red Hat Enterprise Linux OpenStack Platform 7 endpoints do not need special software agents, driver installation, or maintenance. The key benefits to being agentless allow XClarity Administrator to remove operating system dependency, potentially simplifying deployment and maintenance of your Lenovo and Red Hat Enterprise Linux OpenStack Platform solution. The agentless approach helps your CPUs and RAM, too—they're not spending time on agent execution at the endpoint level.

Lenovo offers XClarity Administrator with a one-year, three-year, or five-year software subscription and support and on a per managed server or per managed Chassis Basis; the per chassis licenses offer a more cost-effective way of purchasing licenses for the Flex environment.

Red Hat Enterprise Linux OpenStack Platform

Red Hat Enterprise Linux OpenStack Platform 7 allows you to build a cloud platform that includes fully distributed object storage, persistent block-level storage, virtual machine provisioning engine and image storage, authentication and authorization mechanisms, integrated networking, and a web browser-based interface accessible to users and administrators.⁹ There are two key components to your Red Hat Enterprise Linux OpenStack Platform private cloud: the Undercloud, which must be deployed first, and the Overcloud.

The Undercloud

The Red Hat Enterprise Linux OpenStack Platform 7 Undercloud resides on the primary Director node and includes provision and management components for your other server nodes in your OpenStack environment. Setting up the Undercloud is the first step in creating your private cloud environment, using either a web-based GUI or a terminal-based command line interface. Undercloud components:

- Provide environment planning functions to assign all your necessary Red Hat Enterprise Linux OpenStack Platform roles such as Compute and Controller

⁹ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/preface

- Use the Intelligent Platform Management Interface (IPMI) to control power management and to provide a PXE-based service that registers hardware and installs OpenStack to each of your Lenovo System x3650 M5 and x3550 M5 nodes
- Provide and read a set of YAML templates to create your OpenStack environment

The Undercloud uses a nested approach, whereby OpenStack components are used to install the OpenStack cloud later.

The Overcloud

By first installing the Undercloud onto your Lenovo Performance Rack Servers-based solution, you can create the Overcloud, the actual Red Hat Enterprise Linux OpenStack Platform private cloud environment for your business. The nodes of the Overcloud:

- Provide administration, networking, and high availability for your OpenStack cloud environment (Controller)
- Provide the virtualization resources for your OpenStack environment (Compute)
- Provide the storage resources for your OpenStack environment (Ceph – storage clusters, Cinder – block storage, and Swift – object storage)

The number of virtual machines (VMs) you will need to deploy will dictate how many Controllers you will need for your Overcloud, but you will need to deploy at least one Controller. We installed only one. The same goes for the Compute node—you will need to deploy at least one. The Compute nodes act as a hypervisor and provide processing capabilities for your VMs.

Regarding storage, you should choose which kind of storage is best for your environment. We ran Cinder. After choosing which storage will best suit your Red Hat Enterprise Linux OpenStack Platform 7 cloud environment, consider how many storage nodes you will need. In our case, we installed one, but most environments will have many storage nodes.

Red Hat Enterprise Linux OpenStack Platform Director

Red Hat Enterprise Linux OpenStack Platform Director is a new feature in Red Hat Enterprise Linux OpenStack Platform 7 that enables deployment of your private cloud environment. Red Hat Enterprise Linux OpenStack Platform Director brings together several existing OpenStack components such as the Foreman lifecycle management tool and Triple O, “an open-source project aimed at installing, upgrading and operating OpenStack clouds using OpenStack's own cloud facilities as the

foundations - building on Nova, Neutron, and Heat to automate fleet management at datacenter scale.”¹⁰

Red Hat Enterprise Linux OpenStack Platform Director offers either a GUI or command line scripts to deploy and then manage your private cloud environment.

Red Hat Enterprise Linux OpenStack Platform toolset

Deploying and managing Red Hat Enterprise Linux OpenStack Platform 7 will require the understanding of the OpenStack core service. Running commands with these services allows for further customization of OpenStack clouds.

Red Hat Enterprise Linux OpenStack Platform 7 leverages the following OpenStack components in this deployment: **Horizon, Keystone, Neutron, Cinder, Nova, Glance, Swift, Ceilometer, and Heat.**¹¹

OpenStack components

Horizon

Horizon provides a GUI for users and admins to use for the OpenStack deployment. Also known as OpenStack Dashboard, Horizon provides a modular design, allowing integration with other products and additional management tools. The dashboards and panels available to a user are determined by the role of the user on login.¹² See Figure 7 for an example of a user’s dashboard view in Red Hat Enterprise Linux OpenStack Platform 7.

¹⁰ wiki.openstack.org/wiki/TripleO

¹¹ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/architecture-guide/

¹² access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-dashboard

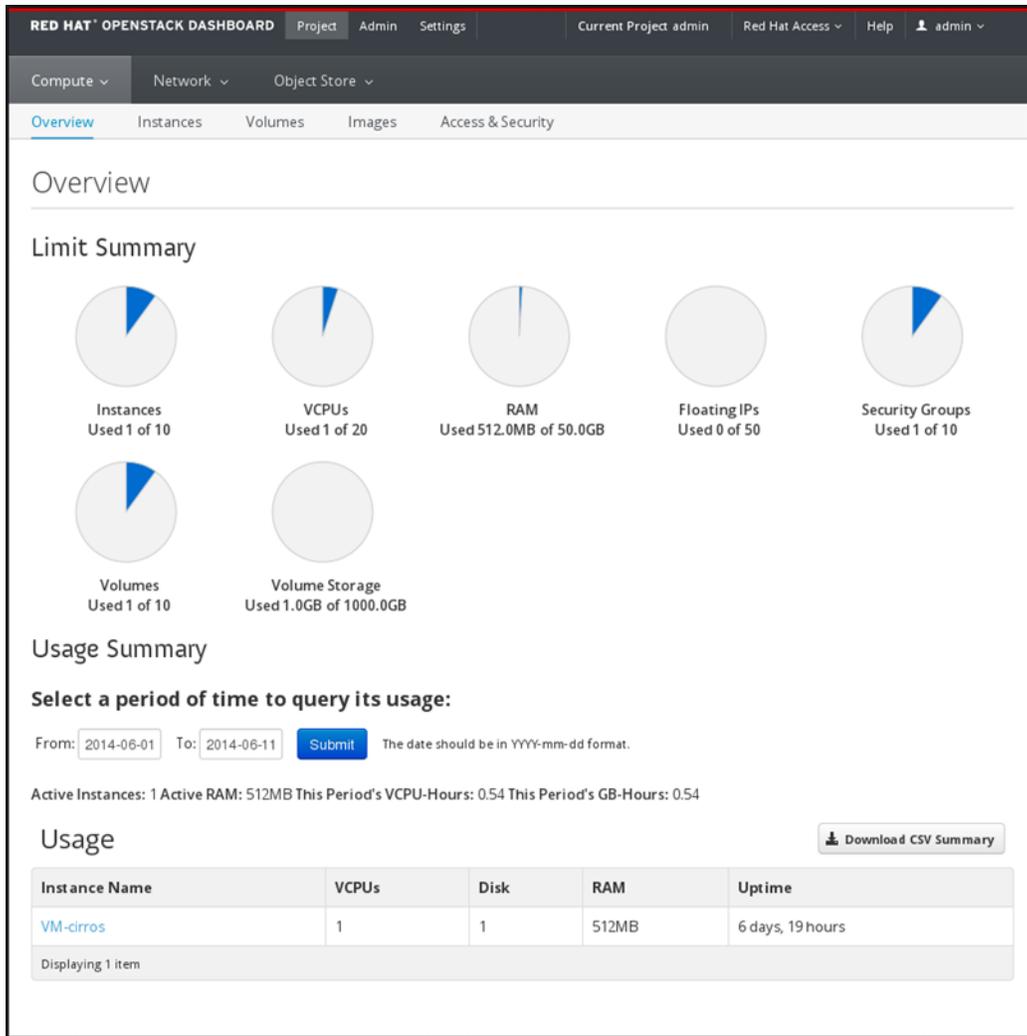


Figure 7: The Red Hat Enterprise Linux OpenStack Platform 7 dashboard view.

Keystone

Keystone is OpenStack's Identity engine that provides user authentication and authorization to all levels of an OpenStack deployment. It supports several different authentication mechanisms including username and password credentials, token-based systems, and Amazon Web Services (AWS)-style log-ins. Keystone also supports multiple back ends to provide token, catalog, policy, and identity information, allowing users to have multiple authentication systems such as Lightweight Directory Access Protocol (LDAP) and Structured Query Language (SQL) concurrently. It also offers federation with Security Assertion Markup Language (SAML), allowing for trust between identity providers and the services being provided to end users.¹³

¹³ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-identity

Neutron

OpenStack Networking (Neutron) controls the virtual networking infrastructure in the OpenStack cloud. Such elements include networks, subnets, and routers, as well as firewalls and virtual private networks (VPNs). Neutron also allows admins to decide which services can run on which physical systems giving each a unique host, or allowing it to be replicated for redundancy. Advantages include user-created networks, traffic control, flexible networking models, and dedicated or floating Internet protocol addresses (IPs).¹⁴

Cinder

Cinder is OpenStack's block storage feature that creates virtual pools of block storage for users to request and allocate for various tasks. The user isn't required to know where the storage is physically located, or where their application is deployed.¹⁵

Nova

Nova, also known as OpenStack Compute, is the core of the cloud services that provides virtual machines on demand. Nova typically utilizes the KVM hypervisor features to create VMs based on images. Admins can restrict images and the number of VMs created based on users, projects, and user quotes. KVM hosts in the OpenStack Compute deployment can share common resources such as storage and networking. Benefits include load balancing, instance distribution, physical isolation and redundancy, labeling for groups of servers with common attributes, and separation of hardware classes.¹⁶

¹⁴ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-networking

¹⁵ wiki.openstack.org/wiki/Cinder

¹⁶ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-compute

Glance

Glance, or OpenStack Image, provides a registry for tracking virtual disk images, pairs with Nova to create new images, and tracks snapshots of existing images. Images and snapshots can optionally be stored in the Object Storage service (Swift) as well as other locations. Supported image disk formats include aki/ami/ari, iso, qcow2, raw, vhd, vdi, and vmdk. Other accepted formats for containers include bare, ova, and ovf.¹⁷

Swift

Swift is OpenStack's object storage feature that allows users to store large amounts of data such as videos, images, VM images, etc. The architecture supports horizontal scaling, failover redundancy, and asynchronous and eventual consistency replication.¹⁸ Ceph can also be used for object storage. Note we did not deploy Swift (or Ceph) in this study.

Ceilometer

Ceilometer, or OpenStack Telemetry, collects data from the other OpenStack components to use for billing, system monitoring, and alerts. Telemetry also offers a plug-in system that administrators can use to add additional monitors to the stack to allow for visibility throughout the entire deployment.¹⁹

Heat

OpenStack Orchestration, also known as Heat, provides template creation to manage or create cloud resources such as applications, storage, networking, and instances. Heat works with all OpenStack core services with auto-scaling and basic high availability. Advantages include a single template that can access all underlying APIs in an OpenStack environment; modular, resource-oriented templates; templates that can be recursively defined and reused; and pluggable resource implementation.²⁰

Red Hat CloudForms

Red Hat CloudForms is an enterprise-focused cloud management solution that complements virtual private cloud infrastructures with a unified interface for providing IaaS. In addition to supporting virtual platforms from VMware, Microsoft, and the public cloud offering from Amazon, Red Hat CloudForms integrates with OpenStack and Red

¹⁷ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-image

¹⁸ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-swift

¹⁹ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-telemetry

²⁰ access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/red-hat-enterprise-linux-openstack-platform-7-architecture-guide/chapter-1-components-comp-orchestration

Hat Enterprise Linux OpenStack Platform Director installations to provide self-service provisioning and management at a granular level, user governance controls, lifecycle management for virtual machines, and capacity planning for your datacenter's resources. Red Hat CloudForms extends the functionality of OpenStack by combining virtual infrastructures into highly scalable enterprise-grade clouds through integration of management services, agents, and compute processes into a single portal for delivering services to users.

PLANNING AND DEPLOYING YOUR PRIVATE CLOUD

Preparing for deployment

This guide assumes you have an existing infrastructure to house the Lenovo System x3650 M5 and x3550 M5 servers, powered by the Intel Xeon processor E5-2600 v3 family. The steps and length of the entire process may vary slightly depending on your infrastructure and scale.

Deployment considerations

Before beginning the deployment, consider the following points:

- **Internet connectivity.** To execute OpenStack installation, you'll need to make sure your environment has Internet access to register your Red Hat subscription, and to download installation and update packages for the Linux operating systems. After Red Hat Enterprise Linux OpenStack Platform Director is deployed, you may also need to obtain Overcloud and discovery images you can use for registering and provisioning nodes.
- **Instance sizing constraints.** In cases where OpenStack default X-Large or custom sized instances will be used, you should add external storage and configure additional OpenStack Cinder servers dedicated for storage provisioning. The addition of external storage is outside the scope of this deployment guide.
- **Traffic isolation.** PXE services are used for deploying servers and applications within your Red Hat Enterprise Linux OpenStack Platform Director-managed hardware. PXE traffic should be isolated from production data traffic either by using separate data networks or by confining PXE traffic to a dedicated VLAN. Refer to the OpenStack TripleO Baremetal Networking guidelines and the Red Hat Enterprise Linux OpenStack Platform 7 Director Networking Requirements in planning your environment's networking.^{21, 22}

²¹ docs.openstack.org/developer/tripleo-docs/environments/baremetal.html

²² access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Director_Installation_and_Usage/sect-Networking_Requirements.html

Deploying the Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform and Red Hat Enterprise Linux OpenStack Platform

There are six steps to deploying this Lenovo solution, as Figure 8 shows.

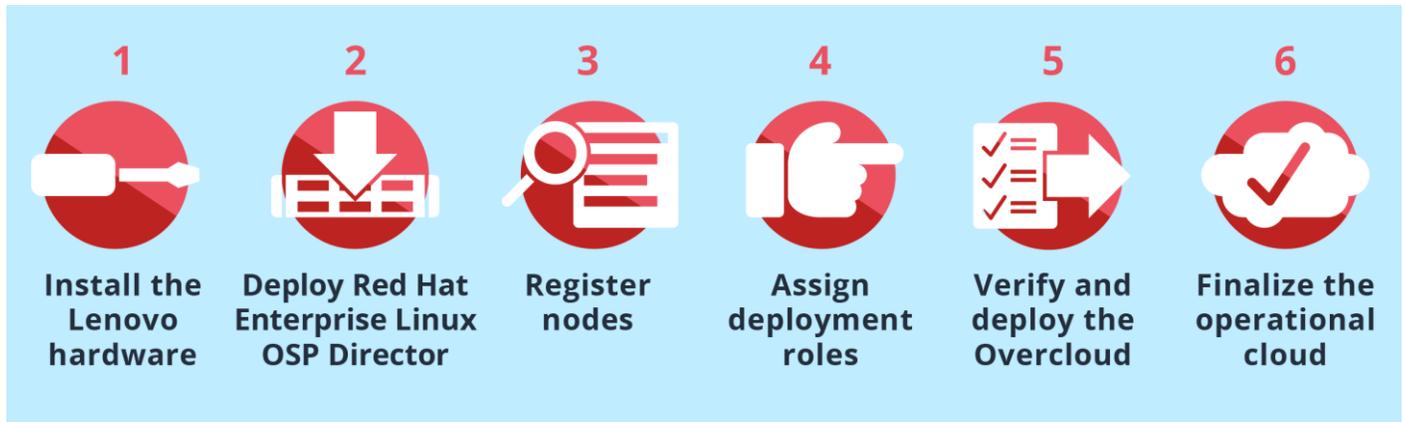


Figure 8: The Red Hat Enterprise Linux OpenStack Platform 7 deployment process for the Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform.

This portion of the guide will walk through each of these six steps to identify the high-level actions required to accomplish each step. For more information on deploying Red Hat Enterprise Linux OpenStack Platform, see access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Director_Installation_and_Usage/index.html.

Step 1: Install the Lenovo hardware

Our hardware stack consisted of three Lenovo System x3650 M5 servers, two Lenovo System x3550 M5 servers, one Lenovo RackSwitch G7028 1Gb ToR management switch, and two Lenovo RackSwitch G8272 10Gb switches. You'll need multiple connections from your stack, and you'll need to configure the RackSwitch G8272 switches with the correct configuration to accommodate the different networks required for the OpenStack Deployment:

1. Connect the dedicated Integrated Management Module (IMM) ports of each server to ports on the management switch, and configure those switch ports with the VLAN for the management network. Your XClarity Administrator system uses this same network for server management and operating system (OS) deployment.
2. Connect NIC1 of each server to ports on the management switch. Configure these ports with VLAN for the management network.

3. Configure the 10Gb switches using the Network Setup Methodology found in [Appendix A](#). After its deployment, the Red Hat Enterprise Linux OpenStack Platform Director node will use these NICs for deploying the Overcloud nodes.
4. Connect one of the 10Gb NIC ports from each node to one of the configured ports on the first Lenovo RackSwitch G8272 switch. Connect the second 10Gb NIC port from each node to one of the configured ports on the second Lenovo RackSwitch G8272 switch.

Step 2: Deploy Red Hat Enterprise Linux OpenStack Platform Director

There are three major tasks required to deploy the Red Hat Enterprise Linux OpenStack Platform Director.

Task 1: Deploy the OS to the Red Hat Enterprise Linux OpenStack Platform Director node

You'll use XClarity Administrator to deploy a Red Hat Enterprise Linux 7.1 image to the Red Hat Enterprise Linux OpenStack Platform Director node. For our deployment, we selected a Lenovo System x3550 M5 for our Director node. Lenovo XClarity Administrator automates deployment of both firmware and the Red Hat Enterprise Linux OS to bare-metal compute nodes or servers. Administrators can deploy the OS or hypervisor onto a single node or onto multiple nodes simultaneously using XClarity Administrator.

First, use XClarity Administrator to discover all the Lenovo Performance Rack Servers in your stack, and then ensure they are up to date with the latest firmware (see Figure 9). If you need to deploy firmware, perform the following steps:

1. Log on to the Lenovo XClarity Administrator console.
2. In the Lenovo XClarity Administrator console, click Provisioning.
3. Click Apply / Activate.
4. In the Firmware Updates: Apply / Activate screen, check all units, change their assigned policy to the policy you created, and click Perform Updates.
5. In the Update summary, select the following:
 - a) Update Rule: Stop all updates on error
 - b) Activation Rule: Immediate activation
6. Click Perform Update.
7. If you get a confirmation window warning you that the endpoint might be restarted, click OK.

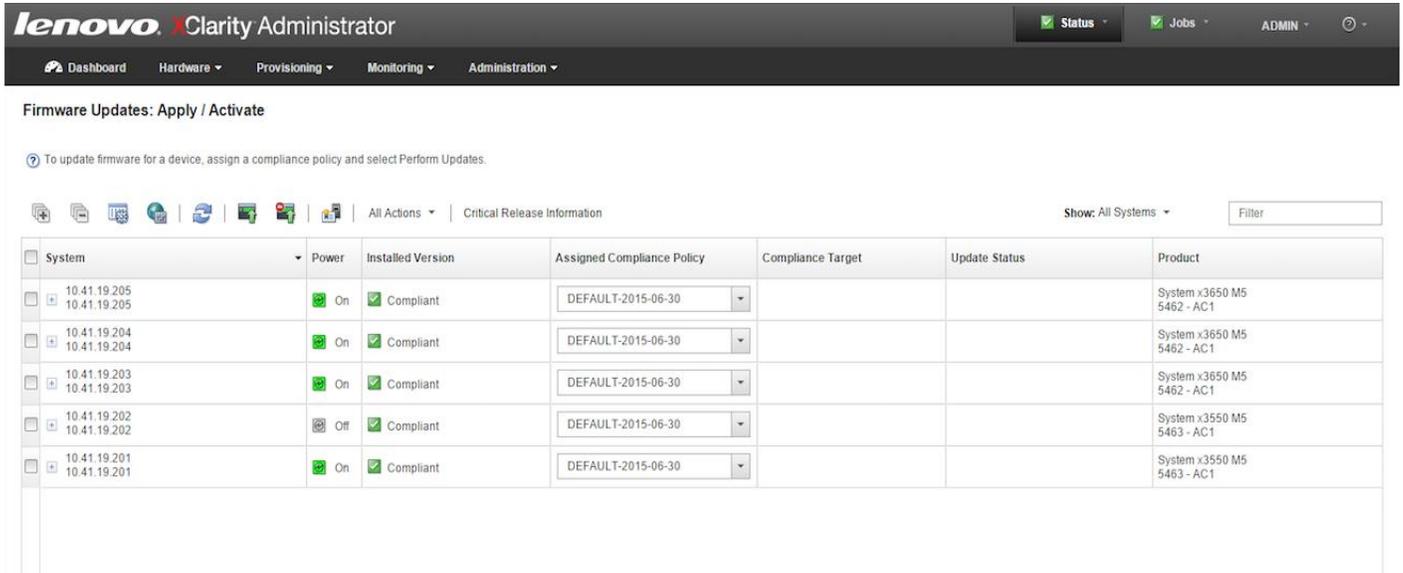


Figure 9: Checking for firmware updates with Lenovo XClarity Administrator.

Next, import your Red Hat Enterprise Linux image (see Figure 10):

1. Log in to the XClarity Administrator.
2. From the top menu, select Provisioning → Deploy Operating Systems → Managing OS Images.
3. Click the Import OS Image icon.
4. Click Browse and locate the Red Hat Enterprise Linux 7.1 image you want to import. Click Open. Click Import.

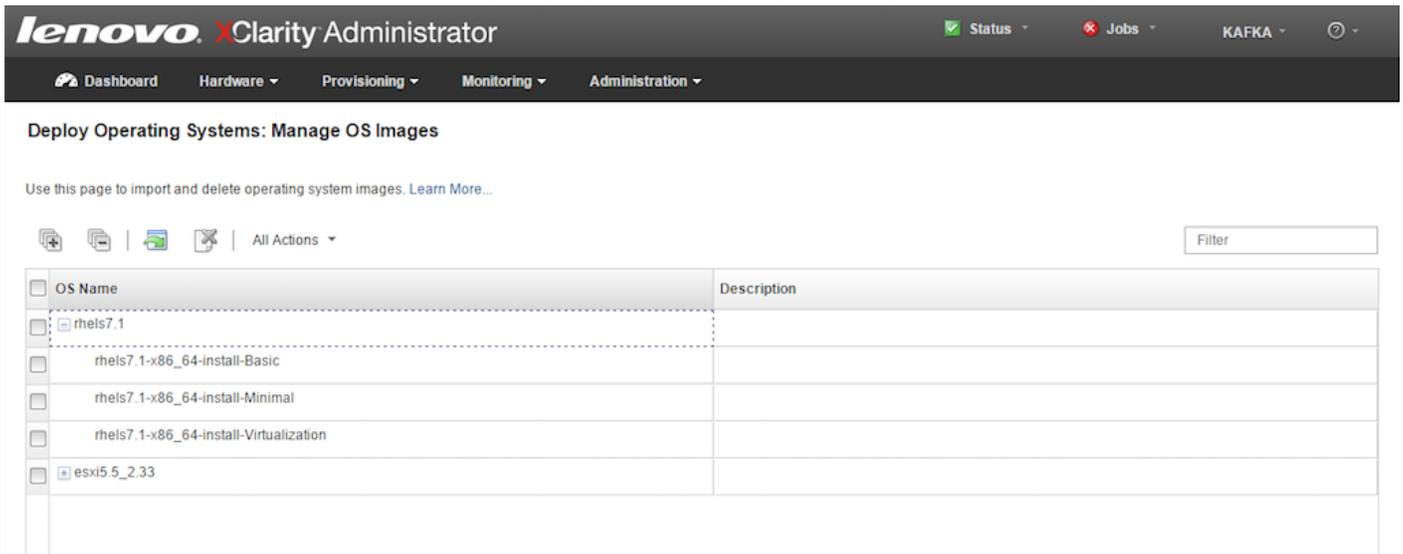


Figure 10: Managing OS images in Lenovo XClarity Administrator.

Finally, configure the global settings and deploy the operating system using the following procedures (see Figure 11 for guidance):

1. Go to the Deploy OS Images screen, and select Provisioning → Deploy OS Images from the top menu.
2. Associate the OS image with the Director node by checking the box at the start of its line, and use the pull-down menu in the Image to Deploy column to select the image rhel7.1rhels7.1-x86_64-install-Basic.
3. To open the Network Settings page, click Edit in the last column.
4. On the Edit Network Settings screen, type `direct01` for the Hostname, `192.0.2.1` for the IP Address, `255.255.255.0` for the Subnet Mask, and `192.0.2.254` for the Gateway. From the pull-down menu, select the MAC Address for the server's first 1GbE network port. Click OK.
5. On the Deploy OS Images page, select Global Settings.
6. In the Global Settings pop-up window, select Credentials, and enter the root password twice.
7. In the same window, select IP Assignment, select Assign static IP address (IPv4), and click OK.
8. On the Deploy OS Images page, select Deploy Images. To start the OS installation, click Deploy on the pop-up screen.
9. We monitored the progress of the installation in a console window by selecting Remote Control.

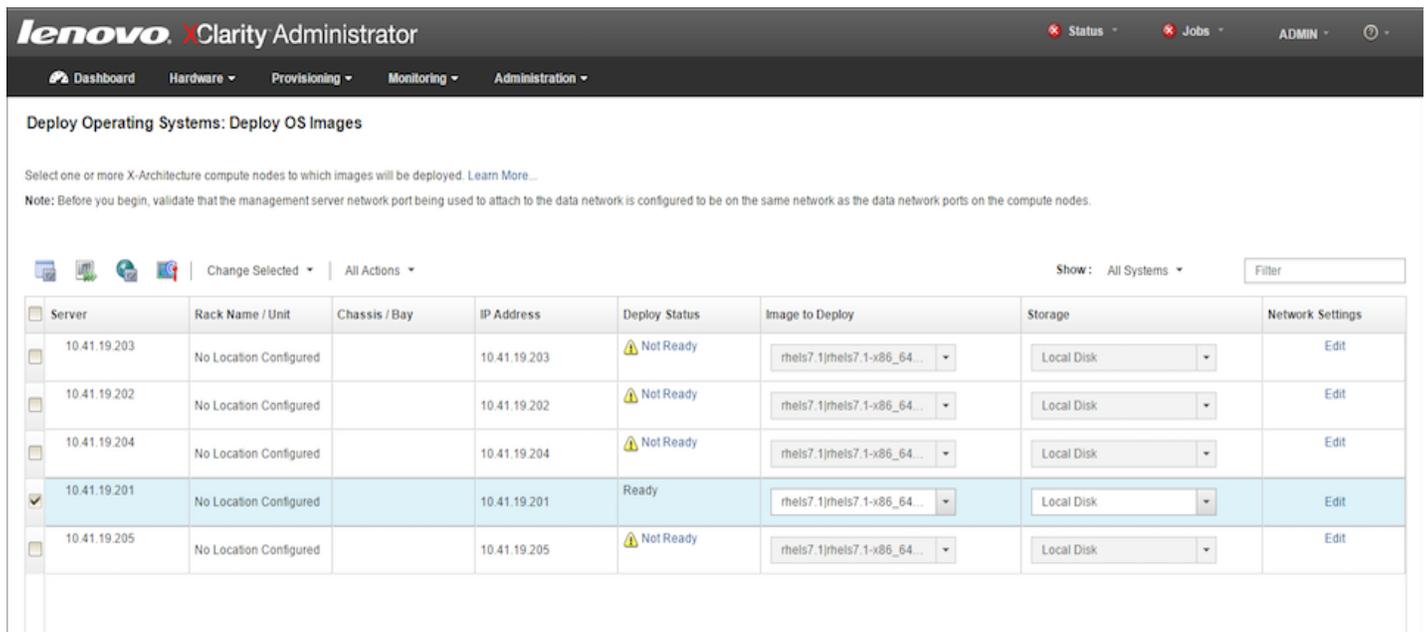


Figure 11: Deploying OS images in Lenovo XClarity Administrator.

Once the installation of Red Hat Enterprise Linux 7 is complete, access the server via Secure Shell (SSH) and the IP address defined during installation, and log in with the root credentials. Alternately, you can click the Remote Control icon in XClarity Administrator and log on as root.

Task 2: Configuring the x3550 M5 Performance Rack Server for Red Hat Enterprise Linux OpenStack Platform Director installation

Next, complete the basic configuration of the Director node. Then, register the system for updates and installation packages and download the necessary components to install Red Hat Enterprise Linux OpenStack Platform Director. Our configuration scripts can streamline this process and are available in [Appendix B](#).

1. Modify the parameters of the second onboard 1Gb NIC interface to start and connect to the management network automatically, as shown in [Basic configuration of the OpenStack Platform Director node](#).
2. Execute Script-1 to perform the following tasks:
 - a. For execution of the OpenStack Platform Director installation command, create a non-root user (we used `stack` as the username)
 - b. Set the OpenStack Platform Director hostname
 - c. Register the server with Red Hat for access to subscriptions
3. Identify the subscription containing OpenStack 7 entitlements and attach the node to it.
4. Execute Script-2 to perform the following tasks:
 - a. Configure software repositories
 - b. Install pre-requisite software
 - c. Update all packages on the system
5. After a reboot, execute Script-3 to perform the following tasks:
 - a. Create an encryption certificate for OpenStack
 - b. Configure time synchronization using Chrony
6. Install the Director configuration and installation tools.
7. Log in as the `stack` user, and create the template for the Undercloud configuration, as shown in [Appendix B](#).

Task 3: Installing Red Hat Enterprise Linux OpenStack Platform Director

With the previous changes, Red Hat Enterprise Linux OpenStack Platform Director is ready to be installed. In our datacenter, we installed Director on our Lenovo System x3550 M5 with the following steps:

1. From the non-root user, run the command `openstack undercloud install | tee director-install.log`. This will launch a configuration script that installs additional packages and configures services as defined in the `undercloud.conf` file. After the script is finished, two files will be generated in the non-root user's home directory:
 - a. `stackrc`, a set of initialization variables that give the non-root user access to the Director's command line tools, and

- b. `undercloud-passwords.conf`, a list of all the passwords generated for the Director's core services.
2. From the non-root user, run the `source` command and specify the `stackrc` file location to initialize the user's ability to use Director's command line tools. We executed `"source ~stack/stackrc"`.

Before installing the Overcloud onto the OpenStack nodes, you'll need to obtain OS images for discovery and deployment, import those images into the Director, and define a nameserver for the Overcloud.

1. First, download the following images from the Red Hat Enterprise Linux OpenStack Platform by following the instructions on the Red Hat Customer Portal:²³
 - a. Deployment Ramdisk for Red Hat Enterprise Linux OpenStack Platform Director 7
 - b. Discovery Ramdisk for Red Hat Enterprise Linux OpenStack Platform Director 7
 - c. Overcloud Image for Red Hat Enterprise Linux OpenStack Platform Director 7

Copy these files into the `/home/stack` directory on the Director node.

2. Extract the contents of these images into `~stack/images`.
3. Modify the Overcloud image for consistent network device naming.
4. Next, change the directory to `~stack/images` and import the images into the Director using the command `"openstack overcloud image upload --image-path ~stack/images/"`.
5. Finally, define a nameserver for the Overcloud. Overcloud nodes require a nameserver so that they can resolve hostnames through DNS.²⁴ On the OpenStack Platform Director node, use the command `"neutron subnet-update $(neutron subnet-list | \awk '/192.0.2.0/ {print $2}')" --dns-nameserver 10.128.0.10"` and replace the last address with the IP address of a DNS server for the environment specific to the Neutron subnet. For our testing environment, we used our infrastructure nameserver.

²³ access.redhat.com/downloads/content/191/ver=7/rhel---7/7/x86_64/product-downloads

²⁴ access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Director_Installation_and_Usage/sect-Setting_a_Nameserver_for_the_Overcloud.html

With the steps above completed, the Undercloud installation is complete. The next sections detail creating the Overcloud.

Step 3: Register the nodes

To deploy the Overcloud, you will first need to register the remaining nodes with the Red Hat Enterprise Linux OpenStack Platform Director. You do this by performing the following steps. See [Appendix C](#) for more details.

1. Log on to the Director node as the stack user.
2. Create the node definition file, which contains hardware and interface information for each of the nodes you want to import. This file is named "instackenv.json". See [Appendix C](#) for an example.
3. Import the nodes by issuing the following commands:
 - a. `openstack baremetal import --json ~stack/instackenv.json`
 - b. `openstack baremetal configure boot`
4. Gather the hardware configuration of Overcloud nodes by issuing the command:

```
openstack baremetal introspection bulk start
```

Step 4: Assign deployment roles

Flavors are hardware definitions that determine whether a server is suitable for a given role. We defined flavors that Red Hat Enterprise Linux OpenStack Platform Director used to define how our hardware would be utilized. Once a server is imported, use the OpenStack ID strings to assign the correct flavor to each node.

Perform the following tasks to assign the deployment roles:

1. Create flavors for the default, compute, controller, and Cinder storage nodes. See [Appendix D](#) for details.
2. List and note the ID strings assigned to the Overcloud nodes by executing the following command:

```
"openstack baremetal list --long"
```
3. Using the ID strings generated by the list command, assign the correct flavor to each node. See [Appendix D](#) for examples.

Step 5: Configure and Deploy the Overcloud

Red Hat Enterprise Linux OpenStack Platform Director (the Undercloud) uses OpenStack's tools to build out an OpenStack environment (the Overcloud). Now that the Undercloud has been installed and configured, use the commands found in [Appendix E](#) to complete the following steps. Make sure to replace the IP addresses with ones suitable for your environment.

1. Copy the OpenStack Heat templates for deploying an Overcloud with bonded interfaces and VLANs to the local directory.
2. Create an environment file that defines the parameters for this template set.
3. Modify the Heat template for the controller node and define which interfaces will be bonded for use with multiple VLANs .
4. Modify the Heat template for the compute nodes and define which interfaces will be bonded for use with multiple VLANs.
5. Modify the Heat template for the Cinder storage node and define which interfaces will be bonded for use with multiple VLANs.
6. Deploy the Overcloud.

Step 6: Finalize the Operational Cloud

After you've deployed all OpenStack components, some post-installation configuration work must be done to make sure your private cloud is ready to deploy virtual machine instances. We have provided detailed examples in [Appendix F](#).

1. Define the networks for use within the tenant – these networks define IP addresses that are useable by the virtual machines within a tenant.
2. Create a router that connects the networks.
3. Add external interfaces to the router, which provide access to resources outside your cloud environment.
4. Allocate additional storage to Cinder.
5. Modify network parameters as needed to function within your environment.
6. Obtaining cloud-ready OS distributions for use in your cloud deployment.

After this post-installation work has completed, you're ready to start deploying virtual machine instances in your cloud. Log in to your OpenStack Dashboard using "admin" as both the username and password.

AFTER THE INSTALL – WORKING WITH RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM, RED HAT CLOUDFORMS, AND LENOVO XCLARITY ADMINISTRATOR

Deploying a private cloud is one thing, but using it and effectively managing it can be a challenge for even experienced administrators. The Red Hat Enterprise Linux OpenStack Platform solution powered by Lenovo servers provides intuitive, graphical based tools to help administrators and users make the most of their private cloud deployments. In this section, we'll discuss several administrative tools that make ongoing management easier. Topics of discussion:

- **Topic 1: Red Hat Enterprise Linux OpenStack Platform Dashboard** – the web GUI that provides administrators with tools for deploying and managing their OpenStack private cloud resources.
- **Topic 2: Red Hat Enterprise Linux OpenStack Platform Director** – when you've outgrown your initial cloud deployment, you can use the Red Hat Enterprise Linux OpenStack Platform Director to quickly scale out your cloud with new hardware resources.
- **Topic 3: Lenovo XClarity Administrator** – a hardware monitoring and management system used for ongoing maintenance and support of the private cloud hardware.
- **Topic 4: Red Hat CloudForms** – a software solution administrators can use to discover and manage multiple clouds under a single pane, enabling self-service provisioning and accounting methods for tracking resource usage for billing purposes.

Topic 1: Red Hat Enterprise Linux OpenStack Platform Dashboard

The Red Hat Enterprise Linux OpenStack Platform Dashboard is the portal used once the OpenStack cloud has been deployed to manage cloud resources, build virtual machines, define networks, create users, and assign permissions to the cloud users. Shown in Figure 12, the dashboard divides resources into categories, and provides administrators with a quick overview of the resource utilization.

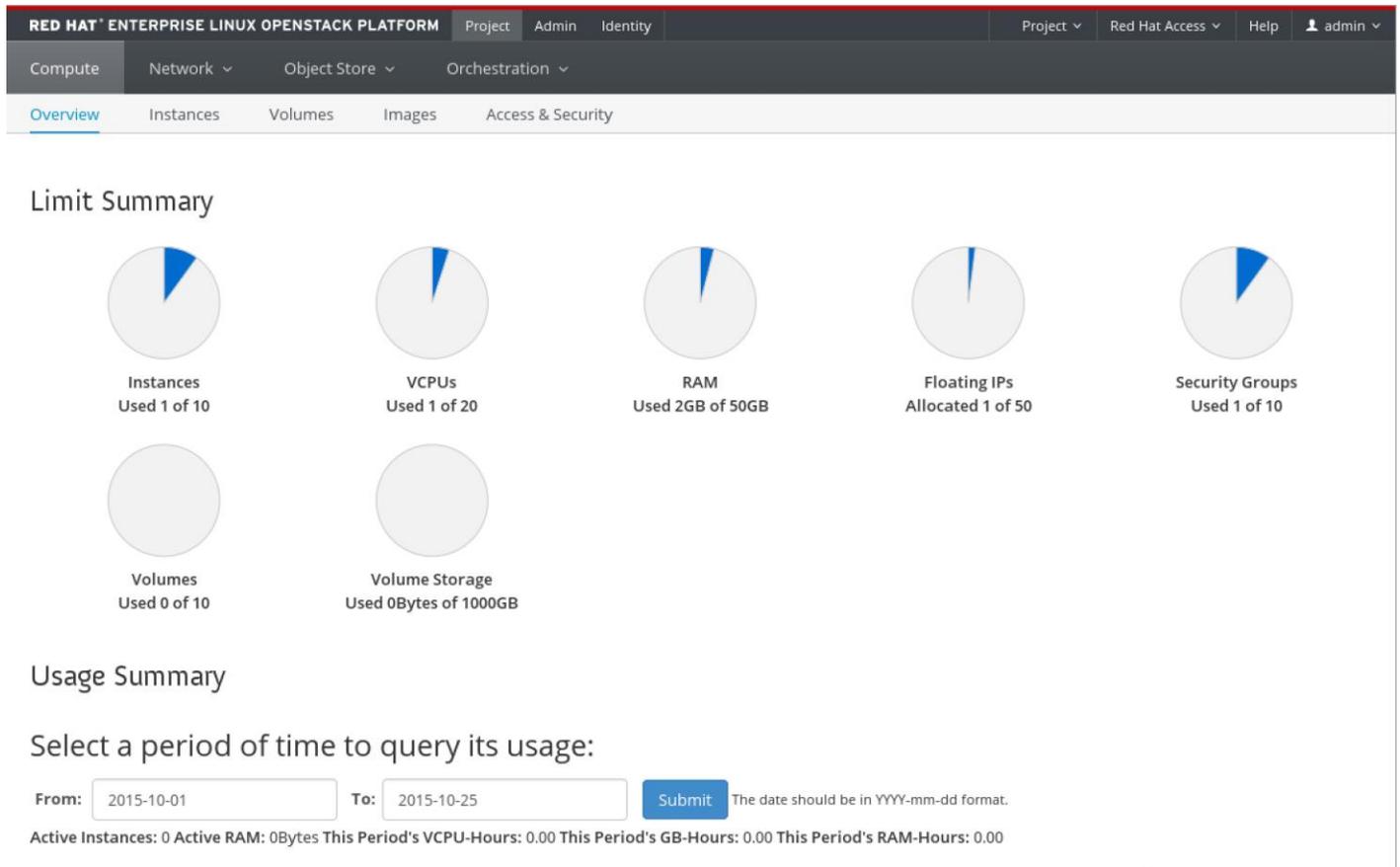


Figure 12: The Red Hat Enterprise Linux OpenStack Platform GUI displaying how resources were divided in our private cloud environment.

Topic 2: Red Hat Enterprise Linux OpenStack Platform Director

After deploying your Red Hat Enterprise Linux OpenStack Platform 7 solution, you may decide to add additional compute resources, storage servers, or other components to enhance the functionality of your cloud environment or workload. Scaling your environment is as simple as deploying additional Lenovo System x3650 M5 and x3550 M5 servers and registering the hardware with Red Hat Enterprise Linux OpenStack Platform Director using a JSON file. Then run a discovery job to characterize the resources of the new servers and assign flavor(s) to them. Finally, re-run the overcloud template command to increase the resources available to your cloud deployment.

Topic 3: Lenovo XClarity Administrator

In addition to deploying the initial firmware and operating systems necessary to get your cloud up and running, Lenovo XClarity Administrator helps administrators with ongoing management tasks, utilizing methods such as policy defined firmware baselines, integrated firmware and software repositories, and the ability to manage systems using

the Integrated Management Module (IMM). Additionally, XClarity Administrator provides hardware alert and event tracking, ensuring that administrators have a quick, easy method for discovering issues as soon as they occur.

Managing the Lenovo hardware with XClarity Administrator

As with other hardware platforms, the servers you use in your private cloud will on occasion require updates to ensure optimal performance and to maintain the highest levels of security. Manually applying firmware updates on a server-by-server basis can mean several hours of work, and creates a risk of inconsistency among your cloud servers, which can contribute to performance issues when virtual machines are migrated from one node to another.

Lenovo XClarity Administrator gives administrators an automated way of managing firmware updates, with the ability to deploy to a single server, or multiple servers at the same time. Additionally, XClarity Administrator maintains an internal firmware repository, and with the ability to define baseline policies that can be applied to a group of servers as shown in Figure 13, you can make sure all of the servers with a particular deployment role have the same firmware for maximum consistency.

The screenshot shows the 'Firmware Updates: Apply / Activate' page in the XClarity Administrator. At the top, there is a navigation bar with 'Dashboard', 'Hardware', 'Provisioning', 'Monitoring', and 'Administration'. Below the navigation bar, there is a header 'Firmware Updates: Apply / Activate' and a sub-header 'To update firmware for a device, assign a compliance policy and select Perform Updates.' Below this, there is a toolbar with various icons and a search filter. The main content is a table with the following data:

System	Power	Installed Version	Assigned Compliance Policy	Compliance Target	Update Status	Product
10.41.19.205 10.41.19.205	On	Compliant	DEFAULT-2015-06-30			System x3650 M5 5462 - AC1
10.41.19.204 10.41.19.204	On	Compliant	DEFAULT-2015-06-30			System x3650 M5 5462 - AC1
10.41.19.203 10.41.19.203	On	Compliant	DEFAULT-2015-06-30			System x3650 M5 5462 - AC1
10.41.19.202 10.41.19.202	Off	Compliant	DEFAULT-2015-06-30			System x3550 M5 5463 - AC1
10.41.19.201 10.41.19.201	On	Compliant	DEFAULT-2015-06-30			System x3550 M5 5463 - AC1

Figure 13: Checking firmware versions of servers in our solution.

Monitoring the Lenovo hardware with XClarity Administrator

Using Lenovo XClarity Administrator, users can leverage hardware monitoring and alerting. All hardware and management events are tracked in the XClarity Administrator event log, while all user actions are stored in the audit log.

Events can be filtered or excluded if deemed unnecessary by the administrator and forwarded to help centralize and allow for aggregation of hardware events. Events in XClarity Administrator also provide information on Lenovo-recommended actions to resolve them, as shown in Figure 14.

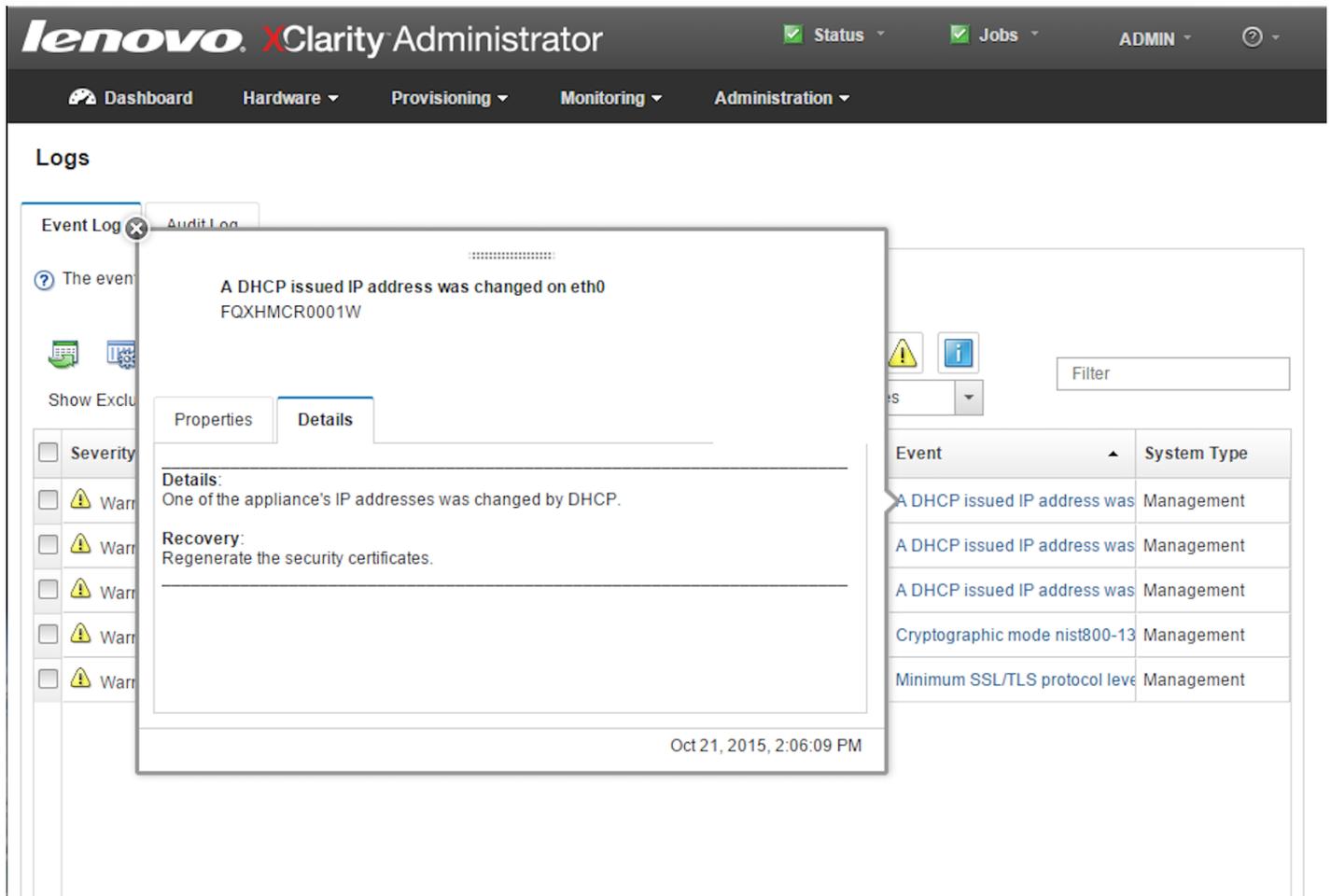


Figure 14: An alert in the XClarity Administrator GUI.

Alerts in XClarity Administrator mean hardware or configuration settings require investigation or other administrative action. Like events, alerts can be filtered or excluded, and contain some information on appropriate actions to resolve the alert, as shown in Figure 14.

Topic 4: Red Hat CloudForms

Red Hat CloudForms provides specific integrations for Red Hat Enterprise Linux OpenStack Platform Director, allowing the discovery of Director nodes and their deployed resources. After multiple clouds are imported into Red Hat CloudForms as infrastructure providers, resources can be combined and offered to users as a self-serve IaaS. Figure 15 shows Red Hat CloudForms discovering OpenStack in our solution.

The screenshot displays the Red Hat CloudForms Management Engine interface. The top navigation bar includes 'RED HAT CLOUDFORMS MANAGEMENT ENGINE' and a user profile 'Administrator | EVM'. Below this, a secondary navigation bar lists various sections: 'Cloud Intelligence', 'Services', 'Clouds', 'Infrastructure', 'Control', 'Automate', 'Optimize', and 'Configure'. A third navigation bar shows sub-sections: 'Providers', 'Clusters', 'Hosts', 'Virtual Machines', 'Resource Pools', 'Datastores', 'Repositories', 'PXE', 'Requests', and 'Configuration Management'. The main content area is titled 'Add New Infrastructure Provider' and is divided into two sections: 'Basic Information' and 'Credentials'. The 'Basic Information' section contains a form with the following fields: 'Name' (OpenStack Director Test), 'Type' (OpenStack Platform Director), 'Host Name (or IPv4 or IPv6 address)' (10.41.19.121), 'API Port' (5000), and 'Zone' (default). The 'Credentials' section has three tabs: 'Default', 'AMQP', and 'RSA key pair'. The 'Default' tab is active and contains a form with 'Username' (admin), 'Password', and 'Verify Password' fields, along with a 'Validate' button. A note below the form states: 'Required. Should have privileged access, such as root or administrator.' At the bottom right of the form, there are 'Add' and 'Cancel' buttons. The footer of the interface shows the date and time: '10/23/15 19:42 UTC'.

Basic Information	
Name	OpenStack Director Test
Type	OpenStack Platform Director
Host Name (or IPv4 or IPv6 address)	10.41.19.121
API Port	5000
Zone	default

Credentials	
Default	
Username	admin
Password
Verify Password

Figure 15: Using Red Hat CloudForms to discover Red Hat Enterprise Linux OpenStack Platform Director.

Beyond the managing of individual clouds, Red Hat CloudForms also offers agentless discovery, assessment, monitoring, and tracking of virtual machines. Red Hat CloudForms provides a management portal from which your administrators can observe VM configurations, resource utilization, and performance. Your admins can generate reports and monitor events to provide insight into stability and capacity. Figure 16 shows Red Hat CloudForms with an OpenStack instance available as IaaS.

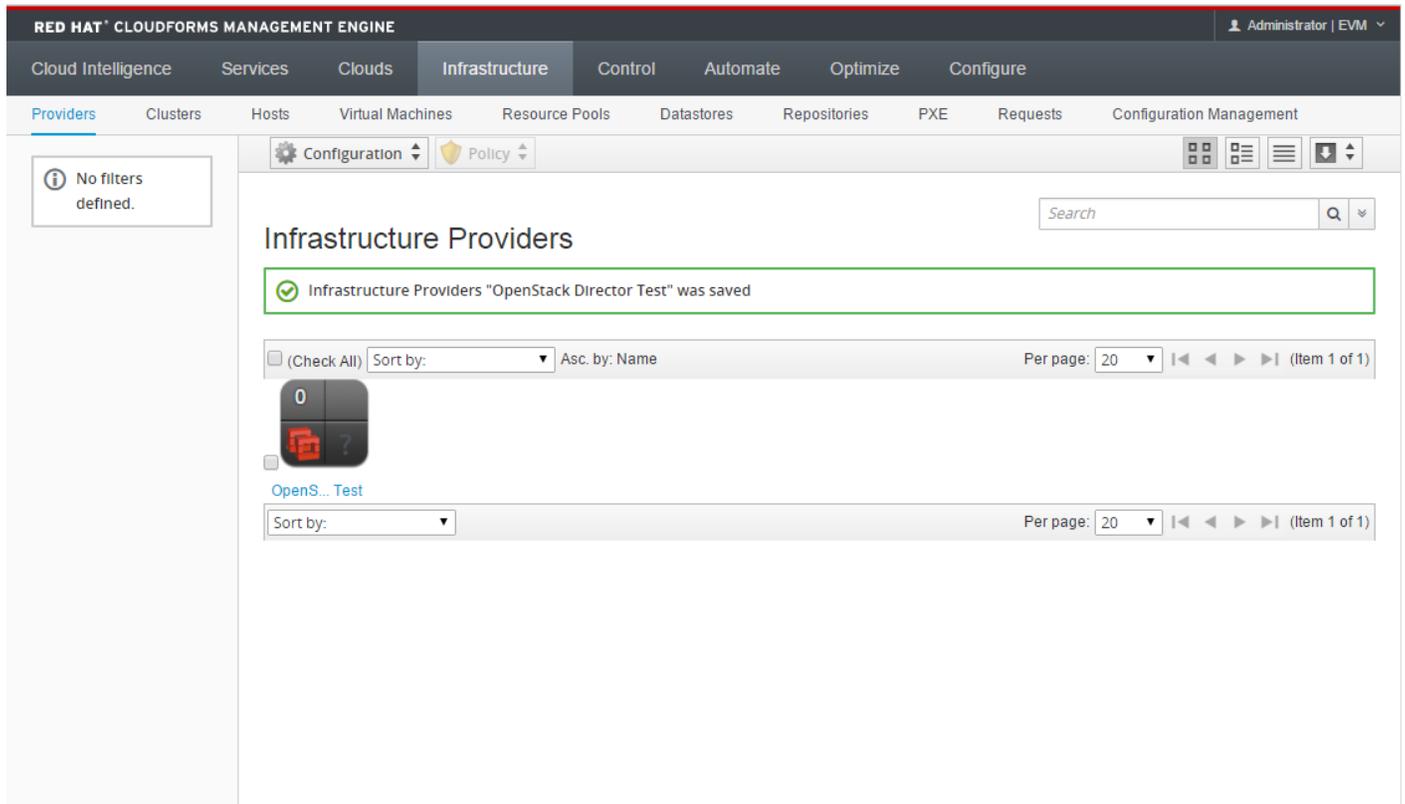


Figure 16: Using Red Hat CloudForms with Red Hat Enterprise Linux OpenStack Platform Director discovered.

VALIDATING THE OPENSTACK DEPLOYMENT

To verify our deployment and demonstrate proof-of-concept based on the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#), we deployed and exercised a number of VMs across our private cloud test infrastructure with actual database loads running within each VM. As a reminder, our private cloud test infrastructure consisted of one Lenovo System x3550 M5 server for OpenStack controller services, two Lenovo System x3650 M5 servers for OpenStack compute nodes, and a single Lenovo System x3650 M5 server for OpenStack Cinder storage. All servers used Intel Xeon E5-2600 v3 family processors.

From the OpenStack dashboard, we launched 20 VM instances, splitting compute resources evenly by placing 10 VMs on each of the two compute nodes. The allocated storage for the VMs resided on the single Cinder storage node. We defined the virtual hardware parameters of each VM with a custom flavor (or template) we created via the OpenStack dashboard. The OS for each VM was spawned from a Red Hat Enterprise Linux 7 image and each VM ran one instance of a PostgreSQL database.

After launching all the VMs via the OpenStack dashboard, we ran the DVD Store database benchmark to exercise each VM's compute resources. During the test runs, we captured performance data for all instances as well as the performance metrics for the Lenovo System x3650 M5 compute nodes and the Lenovo System x3650 M5 Cinder storage node.

The following sections present details of this validation process.

Launching OpenStack instances

Users can launch OpenStack instances from the Launch Instance dialog in the Red Hat Enterprise Linux OpenStack Platform dashboard (see Figure 17). This menu lets you define key parameters for each VM, including the desired flavor, the boot image to deploy and network connections.

Flavors are virtual hardware templates for defining the allocation of resources such as RAM, disk space, and number of virtual CPUs among other parameters. With Red Hat Enterprise Linux OpenStack Platform there are five default VM flavors: tiny, small, medium, large, and x-large. For our validation exercise, we created a custom flavor with 4GB RAM, 1 vCPU, and a 20GB root disk. Our custom flavor shares features common with the standard small and medium flavors.

An OpenStack image contains the base operating system and any customized applications that you wish to deploy. This image is by default stored on the OpenStack controller. The OpenStack image we used contained Red Hat Enterprise Linux, a PostgreSQL database instance, and a corresponding 10GB database for the DVD Store

utility. To create the 20 identical VMs, we chose our custom flavor and our image, and requested 20 VMs.

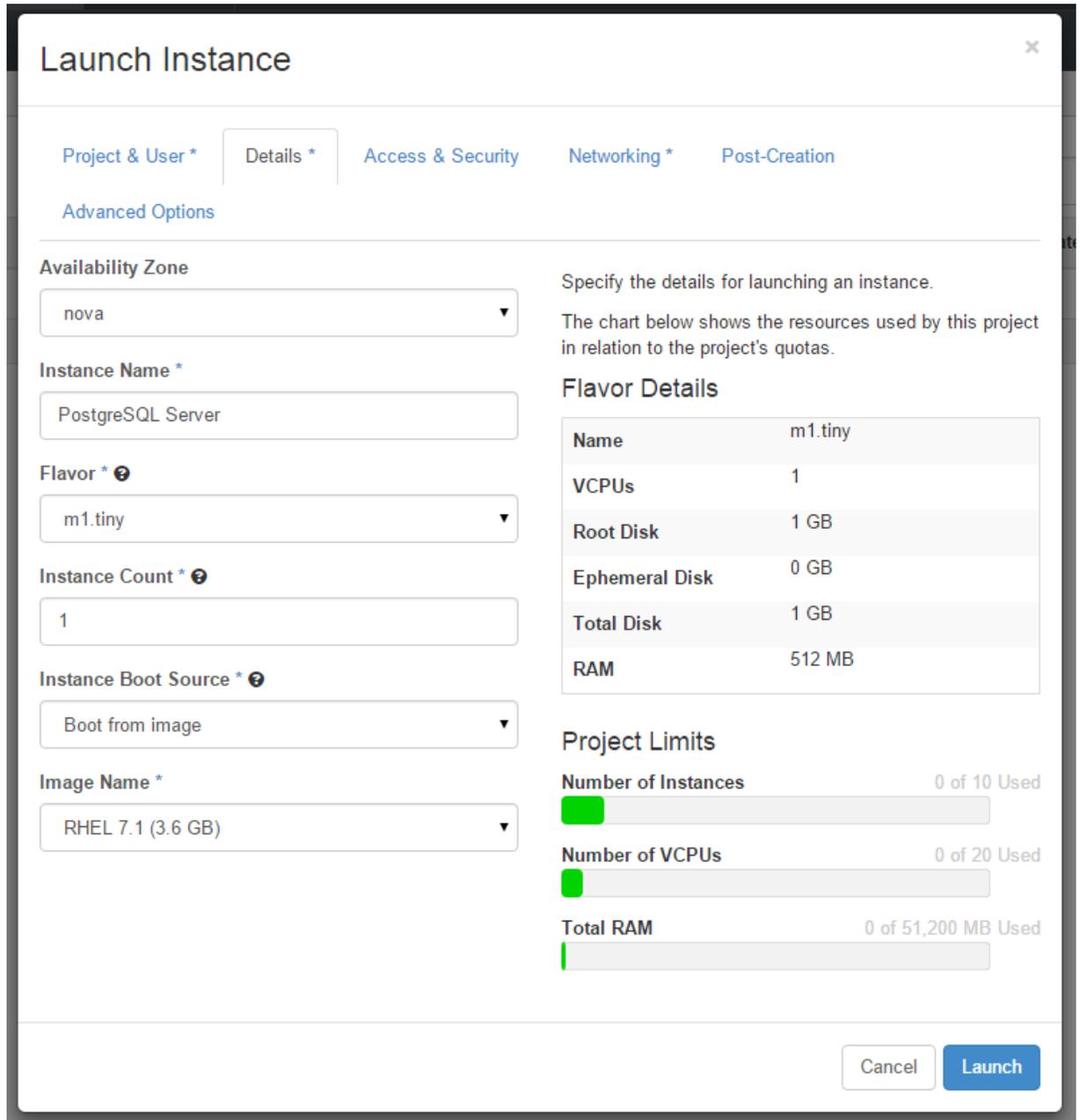


Figure 17: Example of launching an OpenStack Tiny flavor instance from the Red Hat Enterprise Linux OpenStack Platform dashboard. For our validation testing, we deployed a custom flavor.

After we initiated the OpenStack launch request, deployment of the VMs began almost immediately. The OpenStack cluster used our selected image, extracted it, and deployed it 20 times – splitting the compute resources evenly and placing the VMs’ storage resources onto the Cinder node. When the VM deployment finished the

OpenStack instances were marked as active in the OpenStack dashboard. For consistency and repeatability in our testing, we launched one VM at a time, ensuring that 10 VMs were on each compute node.

Operational validation using a database workload

We used DVD Store Version 2 as the benchmark to exercise the PostgreSQL workload on each of the VMs in the private cloud test infrastructure. DVD Store simulates an e-commerce application where users buy items from an online store and place orders. It measures order output rate in orders per minute (OPM). As is typical of most database applications, this application employed a client/server model. In this case, the PostgreSQL workload ran on the 20 OpenStack VM instances. The DVD store client software ran on 20 separate and corresponding VMs, which existed outside of the private cloud test infrastructure. The PostgreSQL database workloads operating in tandem with the DVD Store test utilities simulated a common two-tier client/server architectural approach.

For this validation exercise, we ran a test of the PostgreSQL database workload and then repeated the test cycle twice so that we would have three samples of the same test run. Each test cycle ran for 30 minutes. Prior to each 30-minute measurement period, each test cycle included 15-minute warmup period. The purpose of three tests was to verify test repeatability, and validate functionality of the private cloud test infrastructure. Once the three test cycles were completed, we analyzed the DVD Store OPM and used the median test cycle of the three. The remaining two test cycles were unused.

VM CPU utilization

To execute these test cycles, we launched 20 client machines running outside the OpenStack environment, targeted our custom OpenStack instances, and started the PostgreSQL workloads. In using DVD Store as the benchmark test utility, we could nearly maximize the CPU utilization level of each VM instance. To ensure the VMs reached a high CPU utilization level, we used the parameters available in the DVD Store benchmark test utility, specifically thread count and think time. By doing this, the DVD Store load level pushed CPU utilization for each VM between 98 and 100 percent capacity throughout the course of the test measurement period.

Orders per minute

As noted previously, we paired each OpenStack VM instance running PostgreSQL with a client VM running the DVD Store test utility. In this benchmark utility, using a single execution thread with a 60-second think time would result in one order per minute. As threads increase, think time decreases, or a combination of those two adjustments are made, the target database system will have a greater amount of work

to do, and therefore the VM, in this case, attains more orders per minute and higher utilization levels.

To accomplish this, we configured each DVD Store instance to run with eight threads, each targeting its corresponding OpenStack VM. Additionally, we set the think time for each DVD Store thread to 100-milliseconds to represent the time between tasks associated with a user “thinking” or keying in data. Each environment, application, and user is different, so we cannot necessarily equate these settings to a certain user count. However, as any normal human user would have much greater “think” time than 100ms, we can safely say this configuration of eight threads with 100-millisecond think time simulates a high number of concurrent users.

So, how did the Lenovo reference architecture as deployed in this private cloud test infrastructure stand up? As Figure 18 shows, the number of orders per minute across the two compute nodes exceeded 32,300 OPM. This represents the number of orders combined across all 20 VMs using the median 30 minute test cycle.

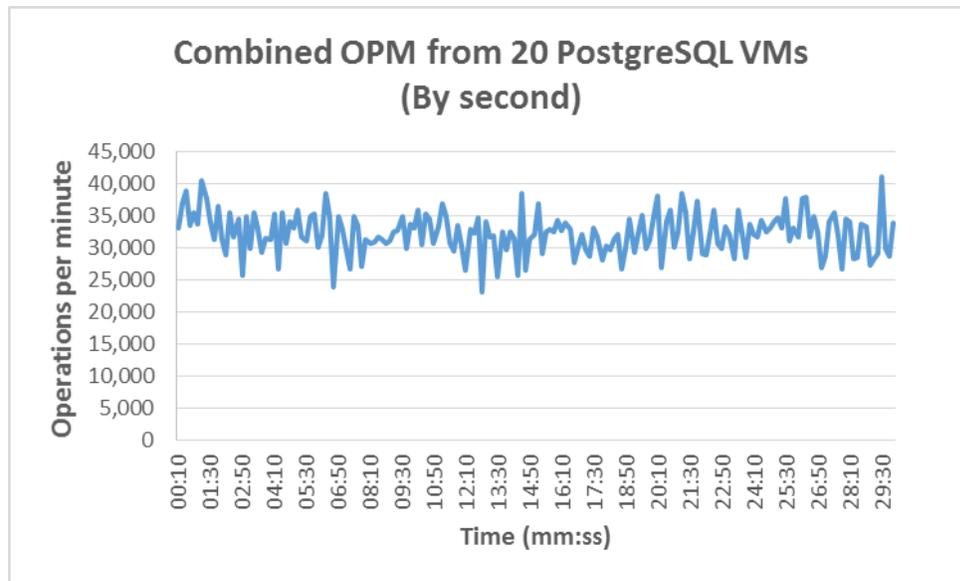


Figure 18: Combined OPM of all 20 OpenStack PostgreSQL VMs for the median 30-minute cycle.

Compute node CPU utilization

Testing demonstrated OpenStack compute node 1 had an average CPU utilization level of 57.8 percent. Likewise, OpenStack compute node 2 averaged 61.1 percent CPU utilization. Figures 19 and 20 show the overall CPU usage across the median test cycle on compute nodes 1 and 2, respectively. According to the OpenStack dashboard, each compute node reported only 40.5GB of memory used out of the physically available 125.5GB. This is a direct result of the VM sizing and the OpenStack flavor selected, as each VM had only 4GB of RAM (see Figure 21).

The proof-of-concept infrastructure we tested is an entry-level configuration that matches the non-high availability deployment noted in section 6 of the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#). Matching the non-HA configuration noted in that document, we used a single Lenovo System x3550 for a Red Hat Enterprise Linux OpenStack Platform Director server, a single Lenovo System x3550 as a controller node, and a single Lenovo System x3650 as a Cinder storage node. Additionally, our environment contained two Lenovo System x3650 compute nodes. See the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#) for other sizing and high availability options.

Despite the minimal configuration, this hardware configuration had more resources to offer in terms of CPU and RAM - approximately 40 percent of the CPU capacity and over half the RAM remained available on each compute node for additional VMs and additional workloads. We could still deploy additional RAM-hungry applications on this small private cloud infrastructure. As this is a scale out architecture and not a scale up architecture, the size of this configuration can be increased within the [Reference Architecture](#) design by simply adding more nodes.

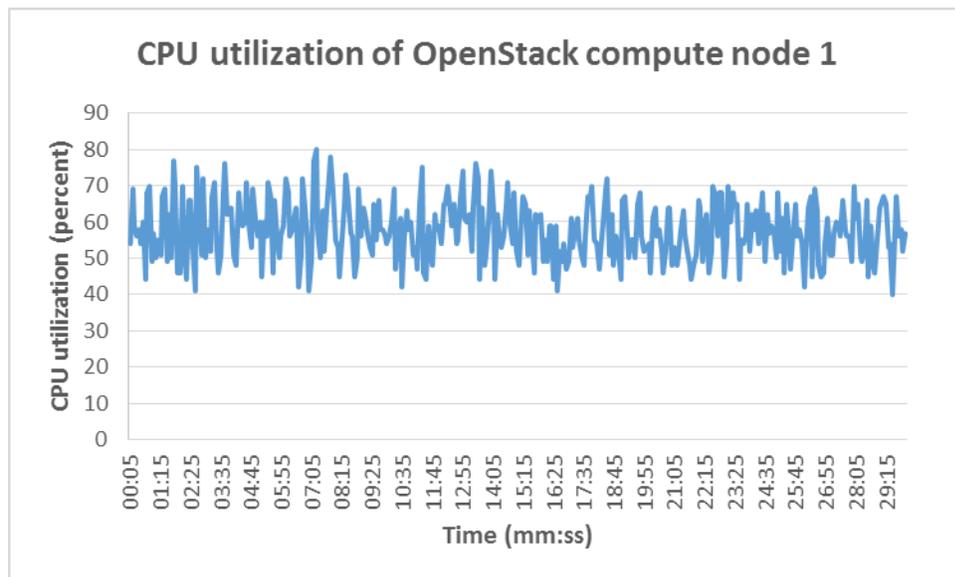


Figure 19: Average CPU utilization on compute node 1.

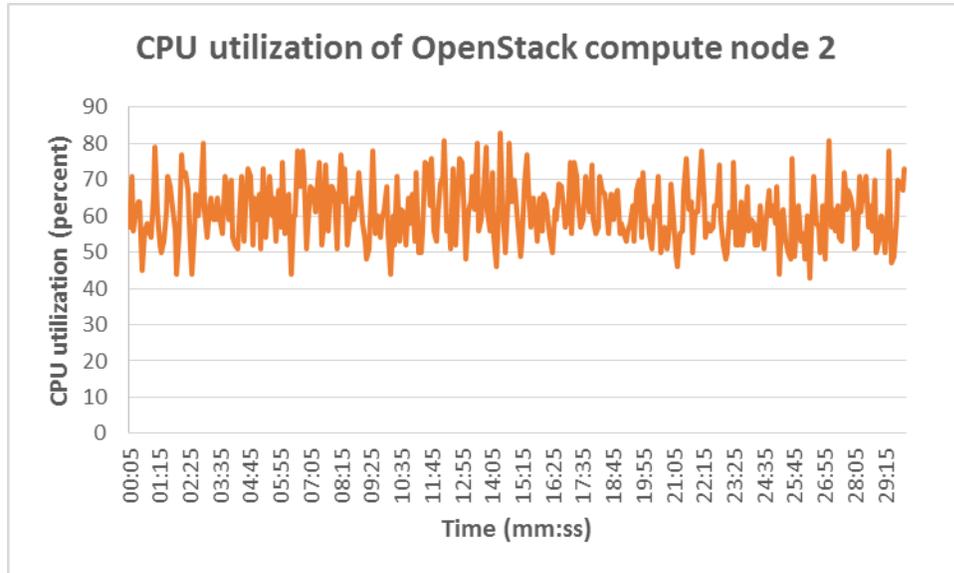


Figure 20: Average CPU utilization on compute node 2.

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM Project Admin Identity Project Red Hat Access Help admin

System

Overview Resource Usage Hypervisors Host Aggregates Instances Volumes Flavors Images Networks

Routers Defaults Metadata Definitions System Information

Hypervisor Summary

VCPU Usage
Used 20 of 80

Memory Usage
Used 81GB of 251GB

Local Disk Usage
Used 400GB of 1.8TB

Hypervisor Compute Host

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Local Storage (used)	Local Storage (total)	Instances
overcloud-compute-1.localdomain	QEMU	10	40	40.5GB	125.5GB	200GB	935GB	10
overcloud-compute-0.localdomain	QEMU	10	40	40.5GB	125.5GB	200GB	935GB	10

Displaying 2 items

Figure 21: Screenshot of the Hypervisor Summary in the OpenStack dashboard showing the total resources used on both compute nodes.

Even in this minimal configuration, the average number of DVD Store orders per minute, running on PostgreSQL database, performed across the two compute nodes

exceeded 32,300 orders per minute. Within the scope of the [Lenovo Cloud Reference Architecture](#), we used just one Lenovo System x3650 server as the OpenStack Cinder node which was not even using a full complement of disks (16 x 2.5" 300GB, 15,000 RPM HDDs). Even with this limited storage configuration, the Lenovo System x3650 handled the high I/O requirements of the PostgreSQL databases from each VM. This proof-of-concept test demonstrates the private cloud infrastructure based on the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#) was operational and had plenty of CPU and RAM resource headroom for additional growth. This test effort also demonstrates how distributed storage resources can be leveraged within this Lenovo Cloud Reference Architecture based testbed.

SUMMARY AND CONCLUSIONS

The proof-of-concept private cloud test infrastructure matched the minimal non-high availability configuration in the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#). Our testbed consisted of two Lenovo System x3650 M5 compute nodes, a single Lenovo System x3650 M5 Cinder node, a Lenovo System x3550 M5 OpenStack controller node, and a Lenovo System x3550 M5 Red Hat Enterprise Linux OpenStack Platform Director server. We used this small-scale private cloud test infrastructure to demonstrate the setup and functionality of the Lenovo OpenStack reference architecture. As it is a scale out architecture, it can grow to a significant size and can support additional VMs and workloads, simply by adding more compute and storage nodes. See the [Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform](#) for other sizing and high availability options.

Also worth noting is the validation testing performed as a part of this deployment guide. Our relational database workload was I/O intensive, which placed particular stress on the single Cinder storage node. Even so, this entry-level configuration showed it had room to support additional RAM and CPU-focused workloads beyond what was tested. A scalable cloud architecture, such as this Lenovo OpenStack reference architecture, is capable of supporting many different types of workloads, such as DevOps applications, Big Data applications (e.g. Hadoop® and Apache™ Spark), and distributed NoSQL database applications (e.g. Cassandra and MongoDB®). Given the flexibility of the Lenovo Cloud Reference Architecture for Red Hat Enterprise Linux OpenStack Platform, if one needed additional capacity, one could simply add additional compute nodes for compute-intensive applications or add additional Cinder storage nodes for storage-intensive applications.

A private cloud infrastructure can bring your business flexibility and scalability while providing greater control over your infrastructure, applications, and data. Within this reference architecture, Lenovo's Performance Rack Servers, Lenovo XClarity Administrator tool and Red Hat OpenStack Platform software are designed to come together to deliver a customizable private cloud solution to meet a wide variety of business workloads. As demonstrated in this guide, deploying and managing Red Hat Enterprise Linux OpenStack Platform 7 with XClarity Administrator on your Lenovo System x3650 M5 and x3550 M5 servers is a straightforward process that can result in a scalable, flexible, and reliable private cloud infrastructure.

APPENDIX A – INSTALLING THE LENOVO HARDWARE: NETWORK SETUP

1. Log in to the first Lenovo G8272 switch from its serial console.
2. Enter privileged mode, and then configuration mode.

```
en
conf terminal
```
3. Configure the switch's name.

```
hostname "RS G8272-01"
```
4. Configure the management IP Address, enable ssh access, and enable NTP time synchronization.

```
int ip 128
ip address 10.41.19.1 255.255.0.0
enable
exit
ssh enable
ntp enable
ntp primary-server 10.41.0.5 MGT
```
5. Define the OpenShift VLANs.

```
vlan 131
name "OPS External Network"
vlan 201
name "OPS Internal API"
vlan 202
name "OPS Storage"
vlan 203
name "OPS Storage Management"
vlan 204
name "OPS Tenant"
```
6. On the first switch, configure the VLAG links to the second switch with access to the same set of VLANs that the server ports will use (compare with step 8).

```
spanning-tree mode pvrst
int port 47-48
switchport mode trunk
switchport trunk allowed vlan add 131,201-204
lacp mode active
lacp key 200
exit
vlag isl adminkey 200
vlag tier-id 10
vlag enable
```
7. Configure the second switch, following steps 1 through 6, but with these switch-specific changes in steps 3 and 4.

```
hostname "RS G8272-02"
int ip 128
ip address 10.41.19.2 255.255.0.0
```

```
enable
exit
```

8. On the both switches, configure the 10GbE ports to the servers in trunk mode with access to VLANs 131 and 201-204 (tagged).

```
int port 2,4,6,8,10,47-48
switchport mode trunk
switchport trunk allowed vlan add 131,201-204
no shutdown
exit
```

9. Configure VLAG health check on both switches.

```
! On the first switch
vlag hlthchk peer-ip 10.41.19.2
```

```
! On the second switch
vlag hlthchk peer-ip 10.41.19.1
```

10. Configure the uplink port to the external network.

```
int port 44
switchport mode trunk
switchport trunk allowed vlan add 131
no shutdown
exit
```

11. Configure the static port channels to aggregate the two links to the servers.

Preparing the director and OpenStack nodes for PXE installation.

We configured the first 1GbE on-board interface on the director and OpenStack nodes for the isolated PXE network.

1. On the Lenovo RackSwitch G7028, we configured the six 1GbE ports to be in their own VLAN.

```
en
conf terminal
vlan 3002
  name "PXE"
exit
int port 1-6
  switchport access vlan 3002
  spanning-tree portfast
  exit
exit
```

2. Log onto the Lenovo IMM II interface for the first OpenStack server-node from your browser with a User name and Password of USERID and PASSWORD.
3. From the pull-down menu, select Server Management -> Adapters
4. Click on the first onboard Ethernet controller listed.
5. On the Controller Properties pop-up screen, select Configuration for the first Ethernet Controller.

6. Under the Main Configuration Page and MBA Configuration Menu, select PXE for the Legacy Boot Protocol.
7. Click Save.
8. On the Controller Properties pop-up screen, select Port Details.
9. Select the Port Details for the first Ethernet controller.
10. Make a note of the value in the Network Address column.
11. Click Close.
12. Select System Status from the pull-down menu.
13. Select Power Actions -> Restart the Server Immediately.
14. Log out from the IMM II interface.
15. Repeat steps 2-14 for the remaining three OpenStack server-nodes.

APPENDIX B – DEPLOYING RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM DIRECTOR 7

Task 1 – Deploying the OS to the Red Hat Enterprise Linux OpenStack Platform Director

First, we installed the Red Hat Enterprise Linux Server 7 operating system on the Director node using Lenovo XClarity Administrator.

Import the RHEL 7.1 image

1. Log on to the XClarity Administrator.
2. To go to the operating system images management screen, and select Provisioning→Managing OS Images from the top menu.
3. To import the Red Hat Enterprise Linux 7.1 installation ISO, select the import image icon.
4. In the Import OS Image pop-up screen, click browse, use the browsing tool to select the installation ISO, and click Import.

Deploy the operating system

1. Go to the Deploy OS Images screen, selecting Provisioning→Deploy OS Images from the top menu.
2. To associate the OS image with the Director node, click the box at the start of its line, and select the image `rhel7.1rhels7.1-x86_64-install-Basic` using the pull-down menu in the Image to Deploy column.
3. To open the Network Settings page, click Edit in the last column.
4. On the Edit Network Settings screen, type `direct01` for the Hostname, `192.0.2.1` for the IP Address, `255.255.255.0` for the Subnet Mask, and `192.0.2.254` for the Gateway. Select the MAAC address for the server's first 1GbE network port from the pulldown menu. Click OK.
5. On the Deploy OS Images page, select Global Settings.
6. In the Global Settings pop-up window, select Credentials, and enter the root password twice.
7. In the same window, select IP Assignment, select Assign static IP address (IPv4), and click OK to close the window.
8. On the Deploy OS Images page, select Deploy Images, and then Deploy in the pop-up screen to start the OS installation.
9. To open a console window to monitor progress of the installation, select Remote Control.

Task 2 – Configuring the Lenovo System x3550 M5 Performance Rack Server for Red Hat Enterprise Linux OpenStack Platform Director installation

Basic configuration of the Red Hat Enterprise Linux OpenStack Platform Director node

1. From the XClarity Administrator console, log in as root with the chosen password.
2. Stop the NetworkManager service for this session.

```
systemctl stop NetworkManager.service
```
3. Modify the IP configuration for the second 1GbE network interface (eno2) by editing the file `/etc/sysconfig/network-scripts/ifcfg-eno2` to start automatically, and connect to your management network. For example, we used this configuration:

```
TYPE=Ethernet  
BOOTPROTO=none  
IPV4_FAILURE_FATAL=no
```

```
IPV6INIT=no
NAME=eno2
UUID=7d179b30-968b-43e5-8095-7ae186e25c95
DEVICE=eno2
ONBOOT=yes
IPADDR=10.41.19.121
PREFIX=16
```

4. Assign the default gateway on your management network by adding an entry similar to the following to the file `/etc/sysconfig/network`.
`GATEWAY=10.41.0.1`
5. Assign the Director server's DNS server by adding an entry similar to the following to the file `/etc/resolve.conf`.
`nameserver 10.41.0.10`
6. Restart networking to apply these changes, and open access to the external network.
`systemctl restart network.service`

Next, we followed the OpenStack Director 7 installation guide (Director Installation and Usage) to prepare the server for installation of OpenStack Director.

1. We ran the following script to configure the stack user, set server hostname, and to register the server with Red Hat subscriptions.

SCRIPT-1

```
#!/bin/bash
# os-local-01.sh -- initial server configuration
useradd stack
echo "password" | passwd --stdin stack
echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
chmod 0440 /etc/sudoers.d/stack

echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
sysctl -p /etc/sysctl.conf

mkdir /home/stack/{images,templates}
chown stack:stack /home/stack/{images,templates}

hostnamectl set-hostname direct01
hostnamectl set-hostname --transient direct01

timedatectl set-timezone America/New_York

sed -i 's|^^(127.0.0.1)|\1 direct01 direct01.localdomain|' /etc/hosts

echo "Enter your subscription username and password."
subscription-manager register
subscription-manager list --available --all | tee subs.txt \
| egrep '^(Subscription Name|Pool ID):'
```

2. Examine the contents of the file `subs.txt` to determine the pool identification for your OpenStack 7 subscriptions (for example, `8a85f9814ff0134a014ff710053466b7`).

3. Attach the OpenStack Platform subscription, using the pool id found in the step 2.

```
subscription-manager attach --pool=8a85f9814ff0134a014ff710053466b7
```

4. We ran a second script to configure the software repositories, install prerequisite software, VNC server and browser, and update all packages on the system.

SCRIPT-2

```
#!/bin/bash
# os-local-02.sh
subscription-manager repos --disable=*
subscription-manager repos --enable=rhel-7-server-rpms \
  --enable=rhel-7-server-optional-rpms \
  --enable=rhel-7-server-extras-rpms \
  --enable=rhel-7-server-openstack-7.0-rpms \
  --enable=rhel-7-server-openstack-7.0-director-rpms

yum install -y yum-plugin-priorities yum-utils

for i in openstack-7.0 "" optional extras \
  openstack-7.0-director; do
  j="rhel-7-server-$i-rpms"
  yum-config-manager --enable $j --setopt="$j.priority=1"
done
yum install -y libguestfs-tools-c
yum update -y
```

5. Reboot the server.

```
shutdown -r now
```

6. After the server reboots, log in as root.

7. We ran this third script to create an encryption certificate for OpenStack and complete miscellaneous server configuration. Note: Replace the IP address of the example NTP server (10.41.0.5) in the next-to-last line with yours.

SCRIPT-3

```
#!/bin/sh
# os-local-03.sh
openssl genrsa -out privkey.pem 2048
echo "Enter your organization's information for OPS certificate"
openssl req -new -x509 -key privkey.pem -out cacert.pem -days 365
cat cacert.pem privkey.pem > undercloud.pem

sudo mkdir /etc/pki/instack-certs
sudo cp ~stack/undercloud.pem /etc/pki/instack-certs/.
```

```
sudo semanage fcontext -a -t etc_t "/etc/pki/instack-certs(/.*)?"
sudo restorecon -R /etc/pki/instack-certs
```

```
sed -i '/^server [^ ]* iburst/d' /etc/chrony.conf
echo "server 10.41.0.5 iburst" >> /etc/chrony.conf
systemctl restart chronyd
```

8. Install the Director configuration and installation tools.

```
yum install -y python-rdomanager-oscplugin
```

9. Log in in as the stack user.

```
su - stack
```

10. Create the template for the Undercloud configuration by creating the file `~stack/undercloud.conf` with the following contents:

Undercloud Configuration Template

```
[DEFAULT]
image_path = .
local_ip = 192.0.2.1/24
undercloud_public_vip = 192.0.2.2
undercloud_admin_vip = 192.0.2.3
undercloud_service_certificate =
local_interface = eno1
masquerade_network = 192.0.2.0/24
dhcp_start = 192.0.2.5
dhcp_end = 192.0.2.24
network_cidr = 192.0.2.0/24
network_gateway = 192.0.2.1
discovery_interface = br-ctlplane
discovery_iprange = 192.0.2.100,192.0.2.120
discovery_runbench = false
undercloud_debug = true

[auth]
# use defaults as generated during configuration
```

Task 3 – Installing Red Hat Enterprise Linux OpenStack Platform Director

1. Run the following command to complete the installation and configuration of OpenStack Platform Director

```
openstack undercloud install | tee director-install.log
```
2. Download three tar files, containing images for installation of Overcloud nodes, by following [these instructions](#). We obtained and used `deploy-ramdisk-ironic-7.1.0-39.tar`, `discovery-ramdisk-7.1.0-39.tar`, and `overcloud-full-7.1.0-39.tar`.
3. Copy the files to `/home/stack` on the Director server.
4. Extract the contents of these archives into `~stack/images`.

```
for i in ~stack/*.tar; do
    tar -C ~stack/images -x -f $i
done
```

5. **Modify the Overcloud image so that its uses "consistent network device naming."**

```
sudo -s
cd ~stack/images
export LIBGUESTFS_BACKEND=direct
virt-customize -a overcloud-full.qcow2 \
    --edit /etc/default/grub:s/net.ifnames=0/net.ifnames=1/
virt-customize -a overcloud-full.qcow2 \
    --run-command 'grub2-mkconfig --output=/boot/grub2/grub.cfg'
exit
```

6. **Upload the images into OpenStack Platform Director as user stack.**

```
cd ~stack/images
openstack overcloud image upload --image-path ~stack/images/
```

7. **Configure the DNS nameserver for the overcloud nodes.**

```
neutron subnet-update $(neutron subnet-list | \
    awk '/192.0.2.0/ {print $2}') --dns-nameserver 10.41.0.10
```

APPENDIX C – REGISTERING NODES

1. Log on to the Director node as user stack.
2. Create the node definition file (~stack/instackenv.json), using the MAC addresses for the provisioning network (the first 1GbE port), and the IP address and login credentials for the Lenovo IMM. Be sure to replace the MAC addresses (highlighted in gray, in the example below) with the values you found in Step 10 in section "Preparing the director and OpenStack nodes for PXE installation".

Example: instackenv.json

```
{
  "nodes": [
    {
      "mac": [
        "00:0A:F7:26:68:30",
      ],
      "cpu": "40",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "pxe_ipmitool",
      "pm_user": "USERID",
      "pm_password": "PASSWORD",
      "pm_addr": "10.41.19.202"
    },
    {
      "mac": [
        "68:05:CA:37:E5:80",
      ],
      "cpu": "40",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "pxe_ipmitool",
      "pm_user": "USERID",
      "pm_password": "PASSWORD",
      "pm_addr": "10.41.19.203"
    },
    {
      "mac": [
        "68:05:CA:37:E5:20",
      ],
      "cpu": "40",
      "memory": "6144",
      "disk": "40",
      "arch": "x86_64",
      "pm_type": "pxe_ipmitool",
    }
  ]
}
```

```

    "pm_user": "USERID",
    "pm_password": "PASSWORD",
    "pm_addr": "10.41.19.204"
  },
  {
    "mac": [
      "68:05:CA:37:E3:48",
    ],
    "cpu": "40",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "USERID",
    "pm_password": "PASSWORD",
    "pm_addr": "10.41.19.205"
  }
]
}

```

3. Add these nodes to the Overcloud.

```

openstack baremetal import --json ~stack/instackenv.json
openstack baremetal configure boot

```

4. Gather the hardware configuration of the Overcloud nodes.

```

openstack baremetal introspection bulk start

```

APPENDIX D – ASSIGNING DEPLOYMENT ROLES

1. Create flavors for the default, compute, controller, and Cinder nodes.

```
openstack flavor create --id auto --ram 4096 --disk 40 \  
  --vcpus 1 baremetal  
  
for flavor in compute control cinder-storage ; do  
  echo $flavor  
  openstack flavor create --id auto --ram 6144 --disk 40 \  
    --vcpus 4 $flavor  
  openstack flavor set --property "cpu_arch"="x86_64" \  
    --property "capabilities:boot_option"="local"\  
    --property "capabilities:profile"="${flavor}" $flavor  
done
```

2. List and note the ID strings assigned to the Overcloud nodes.

```
openstack baremetal list --long
```

3. Using these ID strings, assign the correct flavor to each node. For example, we used the following:

```
ironic node-update 5c285968-8aa8-46e4-8069-17c1d1bbe461 add \  
  properties/capabilities='profile:control,boot_option:local'  
ironic node-update 3b02a1e7-cca7-41cc-9299-44fa9dee1796 add \  
  properties/capabilities='profile:cinder-storage,boot_option:local'  
ironic node-update e20599c0-99ca-451b-b22f-729dc6c46f21 add \  
  properties/capabilities='profile:compute,boot_option:local'  
ironic node-update a6c642df-21ad-4a99-967f-07162f0ff3fd add \  
  properties/capabilities='profile:compute,boot_option:local'
```

APPENDIX E – CONFIGURING AND DEPLOYING THE OVERCLOUD

1. Copy the Heat templates for deploying the Overcloud with bonded interfaces and VLANs to the local directory.

```
cp -r /usr/share/openstack-tripleo-heat-templates/network/config/bond-with-vlan ~stack/templates
```

2. Create an environment file that defines the parameters for this set of Heat templates; that is, create the file `~stack/templates/net-environment.yaml` with the following contents:

```
resource_registry:
  OS::TripleO::BlockStorage::Net::SoftwareConfig:
    /home/stack/templates/bond-with-vlans/cinder-storage.yaml
  OS::TripleO::Compute::Net::SoftwareConfig:
    /home/stack/templates/bond-with-vlans/compute.yaml
  OS::TripleO::Controller::Net::SoftwareConfig:
    /home/stack/templates/bond-with-vlans/controller.yaml
  OS::TripleO::ObjectStorage::Net::SoftwareConfig:
    /home/stack/templates/bond-with-vlans/swift-storage.yaml
  OS::TripleO::CephStorage::Net::SoftwareConfig:
    /home/stack/templates/bond-with-vlans/ceph-storage.yaml

parameter_defaults:
  InternalApiNetCidr: 172.16.0.0/24
  TenantNetCidr: 172.17.0.0/24
  StorageNetCidr: 172.18.0.0/24
  StorageMgmtNetCidr: 172.19.0.0/24
  ExternalNetCidr: 10.131.0.0/16
  InternalApiAllocationPools: [{'start': '172.16.0.10', 'end':
'172.16.0.200'}]
  TenantAllocationPools: [{'start': '172.17.0.10', 'end':
'172.17.0.200'}]
  StorageAllocationPools: [{'start': '172.18.0.10', 'end':
'172.18.0.200'}]
  StorageMgmtAllocationPools: [{'start': '172.19.0.10', 'end':
'172.19.0.200'}]
  # Leave room for floating IPs in the External allocation pool
  ExternalAllocationPools: [{'start': '10.131.20.10', 'end':
'10.131.20.50'}]
  ExternalInterfaceDefaultRoute: 10.131.0.1
  ControlPlaneDefaultRoute: 192.0.2.254
  # The IP address of the EC2 metadata server. Generally the IP of the
Undercloud
  EC2MetadataIp: 192.0.2.1
  DnsServers: ["10.41.0.10", "10.41.0.11"]
  InternalApiNetworkVlanID: 201
  StorageNetworkVlanID: 202
  StorageMgmtNetworkVlanID: 203
  TenantNetworkVlanID: 204
  ExternalNetworkVlanID: 131
```

```

NeutronExternalNetworkBridge: ""
# bonding options
BondInterfaceOvsOptions: "bond_mode=balance-slb"

```

3. Modify the Heat template for the control node by editing the file `~stack/templates/bond-with-vlans/controller.yaml` and make the following replacements:

```

--- controller.yaml      2015-12-07 11:17:33.879901110 -0500
+++ controller.yaml-    2015-12-07 11:17:12.892970018 -0500
@@ -80,7 +80,7 @@
     network_config:
     -
       type: interface
-     name: eno1
+     name: nic1
       use_dhcp: false
       addresses:
     -
@@ -105,11 +105,11 @@
         members:
         -
           type: interface
-           name: ens1f0
+           name: nic2
           primary: true
         -
           type: interface
-           name: ens1f1
+           name: nic3
         -
           type: vlan
           device: bond1

```

4. Modify the Heat template for the compute nodes by editing the file `~stack/templates/bond-with-vlans/compute.yaml` and make the following replacements:

```

--- compute.yaml        2015-12-07 11:17:09.714980453 -0500
+++ compute.yaml-      2015-12-07 11:16:44.567062985 -0500
@@ -71,7 +71,7 @@
     network_config:
     -
       type: interface
-     name: eno1
+     name: nic1
       use_dhcp: false
       dns_servers: {get_param: DnsServers}
       addresses:
@@ -99,11 +99,11 @@
         members:
         -

```

```

-         type: interface
+         name: ens1f0
+         name: nic2
          primary: true
-
-         type: interface
-         name: ens1f1
+         name: nic3
-
-     type: vlan
-     device: bond1

```

5. Modify the Heat template for the Cinder nodes by editing the file `~stack/templates/bond-with-vlans/cinder-storage.yaml` and make the following replacements:

```

--- cinder-storage.yaml 2015-12-07 18:39:43.902349110 -0500
+++ cinder-storage.yaml-      2015-12-07 17:39:10.425306391 -0500
@@ -71,7 +71,7 @@
     network_config:
-
-         type: interface
-         name: eno1
+         name: nic1
+         use_dhcp: false
+         dns_servers: {get_param: DnsServers}
+         addresses:
@@ -99,11 +99,11 @@
         members:
-
-             type: interface
-             name: ens1f0
+             name: nic2
+             primary: true
-
-             type: interface
-             name: ens1f1
+             name: nic3
-
-         type: vlan
-         device: bond1

```

6. Deploy the Overcloud with the following command:

```

openstack overcloud deploy --templates -e /usr/share/openstack-tripleo-heat-
templates/environments/network-isolation.yaml \
-e ~stack/templates/net-environment.yaml \
--control-scale 1 --compute-scale 2 --block-storage-scale 1 \
--control-flavor control --compute-flavor compute \
--block-storage-flavor cinder-storage --ntp-server 10.41.0.5 \
--neutron-network-type vxlan --neutron-tunnel-types vxlan

```

APPENDIX F – FINALIZING THE OPERATIONAL CLOUD

1. After the overcloud deployment has completed, create two networks (and associated subnets) – one for the external network on VLAN 131 and the other to tenant/intra-instance network on VLAN 204.

```
. ~stack/overcloudrc
```

```
# external network/subnet
neutron net-create ext_131 --provider:network_type vlan \
  --provider:physical_network datacentre \
  --provider:segmentation_id 131 --router:external \
  --shared

neutron subnet-create --name ext_131_subnet --enable_dhcp \
  --allocation-pool start="10.131.20.101,end=10.131.20.199" \
  --gateway=10.131.0.1 --dns-nameserver 10.41.0.10 \
  ext_131 10.131.0.0/16
```

```
# tenant network/subnet
neutron net-create ten_204 --provider:network_type vlan \
  --provider:physical_network datacentre \
  --provider:segmentation_id 204 --router:external=False \
  --shared
```

```
neutron subnet-create --name ten_204_subnet --enable_dhcp \
  --allocation-pool start="172.17.0.201,end=172.17.0.249" \
  --gateway=172.17.0.99 --dns-nameserver 10.41.0.10 \
  ten_204 172.17.0.0/24
```

2. Create a router connecting the two networks.

```
TEN_ID="$(keystone tenant-list | awk '$4=="admin" {print $2}')"
neutron router-create --tenant-id $TEN_ID provider_router
```

3. Create two interfaces on the router: one to the external network and the second to the tenant network.

```
neutron port-create --name ten_gateway \
  --fixed-ip subnet_id=ten_204_subnet,ip_address=172.17.0.99
  ten_204
neutron router-interface-add provider_router port=ten_gateway
neutron router-gateway-set provider_router ext_131
```

4. In preparation of allocating additional storage to Cinder, we listed the IP address of the nodes.

```
( . ~stack/stackrc ; nova list )
```

We extended the amount of storage allocated to Cinder by logging on to Cinder node, as the Heat admin user, from the Director node and user stack, and performed the following commands on it.

```
sudo parted /dev/sdb --script mklabel gpt
sudo parted /dev/sdb --script -- mkpart primary 0 -1
sudo pvcreate /dev/sdb1
sudo vgextend cinder-volumes /dev/sdb1
sudo losetup -al
```

```
sudo vgreduce cinder-volumes /dev/loop2
sudo pvremove /dev/loop2
sudo losetup -d /dev/loop2
sudo rm /var/lib/cinder/cinder-volumes
```

5. We adjust the resource quotas for instances and the project.

```
TENANT_ID="$(keystone tenant-list | awk '$4=="admin" {print $2}')"
nova quota-update --instances 200 $TENANT_ID
nova quota-update --cores 200 $TENANT_ID
nova quota-update --ram 307200 $TENANT_ID
nova quota-update --security-groups 200 $TENANT_ID
```

6. Finally, the controller creates instances with network interfaces with small MTUs (1,400 bytes), which can cause problems. We configured the controller to deploy instances with network interfaces with MTU 1,500 bytes. Log on to the controller and perform these commands:

```
sudo sed -i 's/^\(dhcp-option-force=26,1400\)/#\1/' \
/etc/neutron/dnsmasq-neutron.conf
sudo sed -i 's/^\(neutron_dnsmasq_options: dhcp-option-force=26,1400\)/#\1/'
/etc/puppet/hieradata/controller.yaml
```

```
for i in openstack-nova-api openstack-nova-scheduler \
openstack-nova-conductor neutron-server neutron-dhcp-agent; do
sudo systemctl restart $i
done
```

ABOUT PRINCIPLED TECHNOLOGIES



Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.
