



Reference Architecture: Lenovo Open Cloud

Last update: 6 May 2019
Version 1.0

Describes the components of the Lenovo Open Cloud infrastructure for infrastructure software suite

Describes the management cluster and provides examples of deployable workload clusters

Provides the physical and logical network design following Lenovo's best practices

Contains Bill of Materials for both hardware and software

Feng Xia
Jay Bryant
Lijun Gu
Miroslav Halas
Mike Perks



Table of Contents

1	Introduction	1
2	Business problem and business value	2
2.1	Business problem	2
2.2	Business value	2
3	Requirements	3
3.1	Functional requirements	3
3.2	Non-functional requirements	4
4	Architectural overview	5
5	Component model	6
5.1	Hypervisor runtime service	7
5.2	Platform management services	7
5.3	Storage management services	14
5.4	Cloud management services	14
6	Operational model	15
6.1	LOC deployment	15
6.2	LOC management cluster	16
6.3	LOC workload clusters	17
6.4	Building a storage workload cluster	19
6.5	Building a Red Hat OpenStack Platform workload cluster	21
6.6	Networking	24
7	Appendix A: Lenovo bill of materials	32
7.1	Management cluster server BOM	32
7.2	Three-node RHHI-V cluster software BOM	33
7.3	Three-node HCI Ceph and OpenStack software BOM	33
7.4	Three-node Ceph only software BOM	33

7.5 Three-node OpenStack software BOM.....	33
Resources	34
Acknowledgements.....	35

1 Introduction

The target audience for this Reference Architecture (RA) is system administrators or system architects who are looking for an integrated environment that makes deployment of an infrastructure such as Ceph and OpenStack, either for a lab environment or for production, easier, quicker, better automated, better supported, than the traditional, manually intensive approach.

Lenovo Open Cloud (LOC) is infrastructure for infrastructure (iFi) software that utilizes Lenovo ThinkSystem servers and switches. It allows the administrator to deploy both for virtualized and bare-metal workload clusters, dynamically add or remove servers, and just as quickly delete the cluster and reuse the hardware for a different use case. Lenovo Open Cloud, using a bootstrapping process, can also deploy itself.

LOC is designed to combine the best technologies in the market today into a coherent user experience for the following types of users:

- **VM users:** LOC supports hardware virtualization in its core. Virtual machine users and applications can be migrated onto the platform while minimizing dependency on the underlying hardware environment.
- **Cloud users:** LOC provides on-premise cloud computing environments based on Red Hat OpenStack Platform.
- **Container users:** DevOps has continuously pushed the boundary to merge development and production into a single, consistent experience. LOC supports continuous integration/continuous deployment using Red Hat OpenShift Container Platform.

This reference architecture describes the system architecture of the LOC. It focusses on LOC's management services, and how users can use them to build infrastructure such as Red Hat Ceph Storage (RHCS) clusters, and Red Hat OpenStack Platform (RHOSP) clusters.

The core of LOC is based on Red Hat Hyperconverged Infrastructure for Virtualization (RHHI-V). Some experience with the Red Hat products such as RHOSP, RHCS, and RHHI-V may be helpful, but it is not required.

See also the following related Lenovo reference architecture documents:

- Red Hat Hyperconverged Infrastructure for Virtualization - lenovopress.com/lp1148
- Red Hat Ceph Storage - lenovopress.com/lp1147
- Red Hat OpenStack Platform - lenovopress.com/lp0762
- Red Hat OpenShift Container Platform - lenovopress.com/lp0968

2 Business problem and business value

This chapter outlines the value proposition of the Lenovo Open Cloud with Lenovo hardware.

2.1 Business problem

IT infrastructure is the backbone of all other IT activities. But at any given time there are always competing technologies that require different, if not conflicting, infrastructure to support — the stacks are different (more so on the software side than on the hardware side), the groups of users and developers are different, and the workloads running on top often have different architectural genes, and sometimes completely different design paradigms. The problem is how to build an infrastructure such as an OpenStack cluster on demand and how to lower the barrier of operating these different infrastructures using some common “best practices”.

Furthermore, infrastructure means different things to different people, just as the term “cloud” does. What does not change is that all software, regardless the form, language, design pattern, exists on a set of hardware, or be more precise, hardware resources — servers, storage and networks. From an end-user’s perspective, these resources magically appear whenever needed, are always available and always functional. From the IT team’s perspective, managing, monitoring, and maintaining all of these moving parts can be very difficult.

Finally, hardware breaks but yet software applications need to be always available. Total cost of ownership (TCO) can be reduced by sharing hardware and improving its utilization, but this makes it more difficult to troubleshoot when things go wrong.

2.2 Business value

Lenovo Open Cloud meets these challenges. Lenovo recognizes the need to support multiple platforms on the same set of hardware and believes that the infrastructure should enable business and application developers to have the freedom to choose the best tools for the job, not hinder them.

Out of box, the Lenovo Open Cloud has a fully deployed control plane to build a hyperconverged virtual cluster. The hyperconverged cluster provides a small footprint cluster (3-node is the default) and has robust functions to satisfy compute and storage needs, especially for VM-based workloads. The process of adding new hardware resources and on boarding them for workloads is fully automated using LOC’s management services.

Further, each of these platforms is defined in descriptive configuration files that can be used templates, for different deployments at a click of a button — an OpenStack 10 environment for production, another OpenStack 13 environment for the development team, or a Kubernetes container environment for the Northeast region only.

By streamlining the deployment process and treating the infrastructure as code, LOC can reduce the overhead of coordination and communications between the different IT teams (hardware networking, etc.) and get resources ready for the next step in the workflow. The result is that an iteration that used to be measured in days can now be achieved in hours:

3 Requirements

The functional and non-functional requirements for this reference architecture are described below.

3.1 Functional requirements

The primary use case is to build on-premise infrastructure clusters (clouds) to execute workloads; OpenStack, Ceph, Kubernetes, and hypervisor virtualized clusters as depicted in Figure 1 are based on commercial offerings from Red Hat.

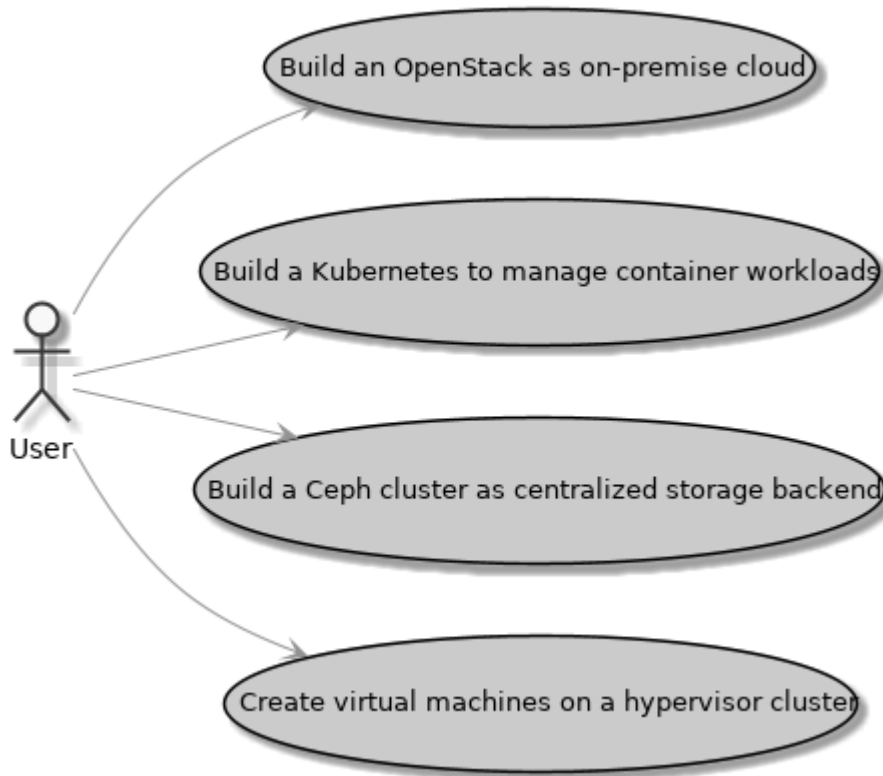


Figure 1: Cluster building use cases

The infrastructure needs to support four types of users — cloud users, container users, storage users, and virtual machine users.

1. As a cloud user, I want to build at minimum a three node OpenStack cluster, so that I can offer an on-premise cloud to my users.
2. As a storage user, I want to build at minimum a three node Ceph cluster and use it as storage backend for OpenStack and Kubernetes, so that data persistence and their life cycle is centrally managed.
3. As a container user, I want to build at minimum a three node Kubernetes container cluster, so that container workloads can be executed on it
4. As a virtual machine user, I want to build at minimum a three node hypervisor cluster, so that I can deploy applications in virtual machines that are portable to VMware and KVM.

Derived from these high-level requirements, are management and implementation use cases:

1. As a user, I want to manage a physical server without installing an operating system, so that I can upgrade/downgrade firmware, select boot device and boot order, turn power on/off, set the out-of-band management IP, manage components including network interface cards (NIC), RAID controllers, disks, and monitor this hardware for failures.
2. As a user, I want to provision an operating system on a server with or without using DHCP, and have OS images managed centrally, so that I can manage versions of these images, add a new server to cluster, or re-commission a server for a different use.
3. As a user, I want manage the software packages on provisioned servers, so that I can install, upgrade or downgrade a version of a software and ensure that any dependencies are also resolved.
4. As a user, I want to detect a server as soon as it is plugged into a monitored network, so that I automate OS provisioning and start to actively use the server.
5. As a user, I want to build hardware inventory of my data center in its current state and to include newly procured hardware before it arrives, so that I can plan for the hardware location, rack position, usage, and configurations.
6. As a user, I want to orchestrate a workflow that supports serial execution and parallel execution of programming scripts, so that I can develop functions and compose them into a workflow to accomplish complex automation.
7. As a user, I want to simplify administration by having a single portal that aggregates management tasks and data of the cloud into a single user interface.
8. As a user, I want to run tests against the cloud infrastructure, so that I can define and validate the health of the infrastructure.

3.2 Non-functional requirements

The infrastructure should be built with maintenance in mind.

3.2.1 Scalability

Infrastructure resources can expand or shrink. This is especially true for a cloud computing platform, which by definition, provides such elasticity.

1. As a cloud/storage/container/VM user, I want to add new server to a cluster for scaled up.
2. As a cloud/storage/container/VM user, I want to remove server from a cluster.
3. As a user, I want to re-purpose a server using automation.

3.2.2 Infrastructure as code (IaC)

Traditionally, an infrastructure does not change much once it is deployed, and rebuilding is often a daunting task. Changing configurations is always risky because the infrastructure lacks built-in support of best practices such as version control, automated testing, and continuous integration/continuous deployment (CI/CD).

1. As a user, I want to define the infrastructure as code, so that I can manage its end state descriptively, and rebuild/re-configure it, at will, with a deterministic result.

4 Architectural overview

This chapter gives an architectural overview of Lenovo Open Cloud. Figure 2 shows the architecture of Lenovo Open Cloud using a three node cluster of Lenovo ThinkSystem servers.

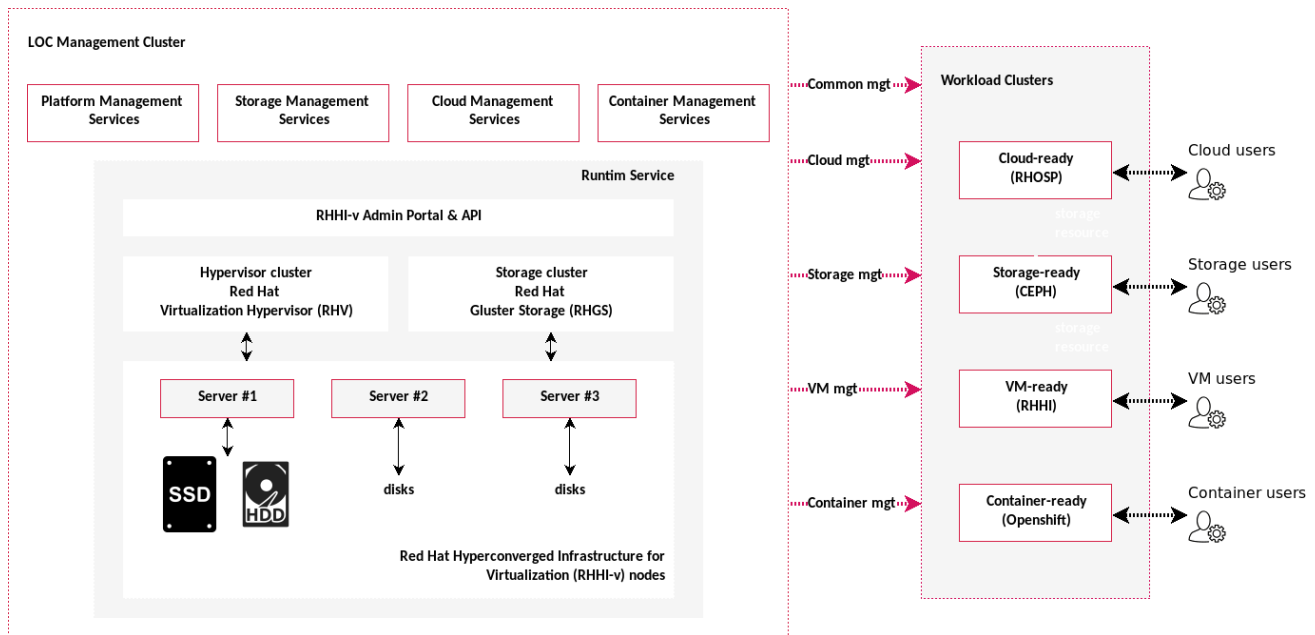


Figure 2: Architecture of Lenovo Open Cloud

Lenovo Open Cloud (LOC) contains two types of clusters: a management cluster and one or more workload clusters.

The management cluster includes four types of management services:

- Platform management services include a runtime service that supports VMs and services to manage infrastructure hardware and software. The services cover a wide range of common use cases including server discovery and provisioning, software package management, configuration driven automation, workflow, inventory planning and tracking, and validation, monitoring and storage performance testing.
- Storage management services such as Ceph to manage storage infrastructure.
- Cloud management services such as Red Hat OpenStack Platform to manage the cloud infrastructure.
- Container management services such as Red Hat OpenShift to manage the container infrastructure.

The workload cluster refers to the software and applications that run either on top of LOC or in parallel to LOC, while their management software, such as OpenStack controllers are deployed as LOC services.

5 Component model

Lenovo Open Cloud (LOC) consists of four layers:

- Hypervisor runtime service
- Management services (platform, storage and container)
- Configuration templates and deployed instances
- Interfaces

Figure 3 gives a high-level overview of the LOC components.



Figure 3: Lenovo Open Cloud Components

5.1 Hypervisor runtime service

Lenovo Open Cloud uses the Red Hat Hyperconverged Infrastructure for Virtualization to provide the hypervisor and runtime environment for virtual machines (VMs). For the details, see the Lenovo Reference Architecture for Red Hat Hyperconverged Infrastructure for Virtualization: lenovopress.com/lp1148.

5.2 Platform management services

Table 1 lists the ten platform management services that provide administrative functions to support the operation of LOC.

Table 1. Platform management services and versions

Service	Implementation Example	Version
Software repo	Red Hat Satellite	6.4
Automation	Red Hat Ansible Tower	3.4.2
Discovery	Lenovo Confluent	2.2.0
Inventory planning	Digital Ocean Netbox	Lenovo fork
OS deployment	Lenovo xClarity	2.3.0
Configuration repo	Gitlab CE	8.1.4
Validation	Jenkins, Elasticsearch, Logstash, Kibana	J(2.150.3), ELK (6.5.4)
OS Image	Lenovo distribution point	n/a
Single pane of glass	Red Hat Cloudforms	5.10.0.18
Launchpad	Red Hat Enterprise Linux	7.5

These platform management services are required for the deployment and management of a cloud environment to perform services such as extending the cluster with new hardware resources, provisioning of bare-metal, deploying management software, maintaining configurations, and administration with a single view of truth. Together these services provide comprehensive coverage and automation has been developed for many of these services to help speed up the deployment and management of a cloud environment.

All LOC platform management services except the Discovery Service and the Launchpad have an admin web portal and a RESTful API. Integration of services are orchestrated through the Automation service. Automation code, template, and default configs are maintained by the Config Repo service. The Automation Service is synchronized with the Config Repo, and has the option to execute automation based on version, branch, or even a specific commit. This makes it possible to fine tune automation and try out new development.

The Discovery service and Inventory Planning service work together to maintain hardware inventory including procured future inventory and new hardware just plugged into the grid for the first time. These hardware resources can then be used based on their characteristics. In addition, a pre-determined automation path can be activated as soon as the hardware is detected so that its life cycle starts with little operator intervention.

The Software Repo service and OS Image service provide the baseline of software packages for all other services. OS Images are mostly static and have no version control. Software Repo, on the other hand, is

highly sophisticated and can integrate with upstream sources including vendor release points and open source projects providing that the package format is compatible.

The Lenovo Open Cloud contains a set of recipes to orchestrate these services to work together to achieve highly sophisticated automation such as building an OpenStack cluster as soon as the compute nodes have been powered on without even setting the out-of-band IP address. Planning is still required, but these services and LOC's recipes will drive these services to carry out the end-to-end automation of many common use cases, once the configuration templates have been completed with necessary information.

5.2.1 Software repository and life cycle management service

This service provides the central repository for software packages and is the upstream source for all installers. It handles software licenses, updates, and versioning for all third-party software. In Lenovo Open Cloud, it manages Red Hat subscriptions for all LOC hardware and software. A user can also use it to manage subscriptions for workloads, making it a single source-of-truth of software licenses.

Out-of-box, all native LOC hardware and VMs are registered to this service, and it provides a matching number of subscriptions and entitlements to support the life cycle of these subscribers. The Software repo service is also the central management for Docker images. It can be viewed as a local Docker registry so LOC infrastructure can support container-based applications out of the box.

This service is built on the Red Hat Satellite. Satellite is an on-premise alternative to trying to download all of your content from the Red Hat content delivery network or managing your subscriptions through the Customer Portal. From a performance side, it reduces hits to your network bandwidth because local systems can download everything they need locally; from a security standpoint, it can limit the risks of malicious content or access, even enabling entirely disconnected environments. Satellite is composed of a centralized Satellite Server. Depending on your data center setup, organization design, and geographic locations, you can have local Capsule Servers, which are proxies that locally manage content and obtain subscriptions, registration, and content from the central Satellite Server. See access.redhat.com/products/red-hat-satellite for more details.

The key concepts of the service are:

- **Life cycle environment:** Logical group to make different versions of the same software available to members in the group. The name implies that the environments are defined to represent the *life cycle* of the application, e.g. development, test, and production.
- **Content view:** Logical group to segment software sources in the repo. For example, a view that includes the OpenStack files, but excludes OpenShift.
- **Subscriptions:** These are software licenses that controls the number of endpoints supported by one license and its expiration date.

Life cycle environment

Application life cycles are divided into life cycle environments, which represent each stage of the application life cycle. Life cycle environments are linked to form an environment path. Content can be promoted along the environment path to the next life cycle environment when required.

The LOC Software repo service offers three environments: development, test, and production with a default path of development → test → production.

- The “development” environment usually maintains the newest version of packages. Once a version satisfies development, it version can be promoted to “test” for more rigorous testing and broader integration.
- The “test” environment is a staging area for QA and CI/CD. Versions included in this environment are usually promoted from development but are not necessarily the latest version.
- The “production” environment supports a stable version of applications.

In LOC, all hosts and virtual machines of the management cluster are registered to the production life cycle environment. A new resource, created by the Automation service is registered to the development environment by default. An automation parameter can specify a different environment.

Subscriptions

Table 2 lists the subscriptions needs to support a three node management cluster. Additional subscriptions are needed for infrastructure such as Red Hat Ceph storage and Red Hat OpenStack Platform.

Table 2. Software repo and life cycle subscriptions of a three node management cluster

Subscription	Quantity
Red Hat Satellite	1
Red Hat Virtualization	3
Red Hat Gluster Storage	3
Red Hat Ansible Tower	1
Red Hat CloudForms	1
Red Hat Ceph Storage	1
Red Hat Enterprise Linux	13
Red Hat OpenStack Platform	3
OpenShift	1

Content views

Content views are managed selections of content, which contain one or more repositories (yum, puppet, or containers) with optional filtering. These filters can be either inclusive or exclusive, and tailor a system view of content for life cycle management. They are used to customize content to be made available to client systems.

Content views are used to make subscriptions available to managed hosts and virtual machines. Table 3 lists the content views provided by LOC.

Table 3. Software repo and life cycle management Red Hat Satellite content views

Content View	Subscription
platform	Red Hat Gluster Storage and Red Hat Virtualization
rhel	Red Hat Enterprise Linux
satellite	Red Hat Satellite
dashboard	Red Hat CloudForms
ansible	Red Hat Enterprise Linux

Activation keys

An activation key ensures that the content host is associated with the correct host group and that the content host consumes the correct subscription. Activation keys store a set of subscriptions that can be associated with content hosts without attaching these subscriptions directly.

LOC names activation keys using the pattern `<content view><life cycle> key`. Subscriptions under each key is consistent with the content view. Table 4 lists the activation keys provided by LOC.

Table 4. Software repo and life cycle management activation keys

Activation Key	Content View	Environment	Subscription
rhel development key	rhel	development	Red Hat Enterprise Linux
rhel test key	rhel	test	Red Hat Enterprise Linux
rhel production key	rhel	production	Red Hat Enterprise Linux
satellite development key	satellite	development	Red Hat Satellite
satellite test key	satellite	test	Red Hat Satellite
satellite production key	satellite	production	Red Hat Satellite
dashboard development key	dashboard	development	Red Hat CloudForms
dashboard test key	dashboard	test	Red Hat CloudForms
dashboard production key	dashboard	production	Red Hat CloudForms
ansible development key	ansible	development	Red Hat Enterprise Linux
ansible test key	ansible	test	Red Hat Enterprise Linux
ansible production key	ansible	production	Red Hat Enterprise Linux
platform development key	platform	development	Red Hat Gluster Storage and Red Hat Virtualization
platform test key	platform	test	Red Hat Gluster Storage and Red Hat Virtualization
platform production key	platform	production	Red Hat Gluster Storage and Red Hat Virtualization

5.2.2 Automation service

The Automation service is an execution orchestrator built on Red Hat Ansible Tower. It supports common workflow patterns such as sequential, parallel, mixed, and cron. The execution status and outputs are logged and trackable through this service. See access.redhat.com/products/ansible-tower-red-hat for more details.

The Automation service fully integrates with the Config Repo service to have access to code, scripts, templates, and template instances; all managed by the Config Repo service. By changing where it points to, the service can run a version of a file using a git branch for testing a feature implementation or using a git commit for testing or debugging a specific code change.

Lenovo Open Cloud is shipped with a list of pre-defined automation that makes managing the infrastructure easy and efficient.

5.2.3 Discovery service

The Discovery service is a helper service. As shown in Figure 4, it continuously monitors the OOB network for new Lenovo hardware resources. When used together with the Inventory Planning service, it can initialize a device inventory of hundreds even thousands of devices in a matter of seconds.

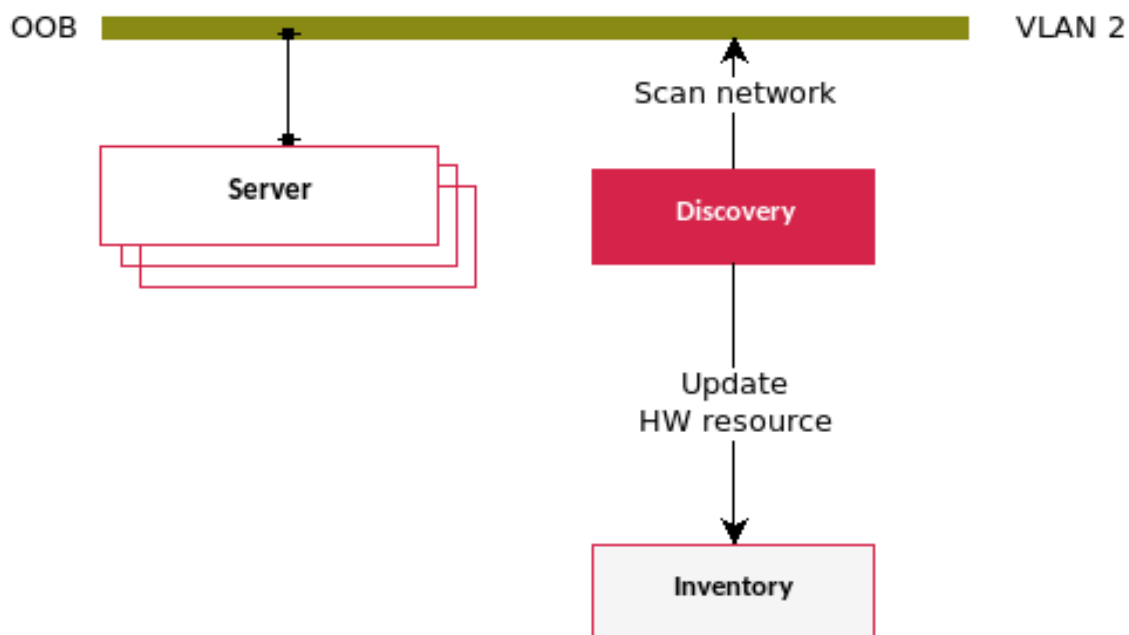


Figure 4: Discovery service overview

A role-based discovery can further reduce operator task by matching a device UUID with the inventory record, and start a pre-defined server configuration as soon as the device is connected to the network such as configuring an IP address for the ThinkServer XCC.

5.2.4 Inventory planning service

This service is the source-of-truth of infrastructure planning that handles the pre-on-board phase of hardware such as tracking device attributes (manufacturer, part number, model, etc.), allocating rack position, assigning it to a tenant, organizing it by site or by organization, designating a role if using role-based LOC discovery automation (see “Discovery service” for details).

A unique feature of this service is to be the source-of-truth of the cable schema between LOC servers and switches. It can both take a schema config from the Configuration repository service to apply to the switch, or create a schema configuration based on the hardware inventory.

5.2.5 Server Configurator and OS deployment service

This service configures hardware including Lenovo ThinkServer bare-metal servers and Lenovo switches. Common tasks such as firmware updates, boot order, and switch port configuration are managed by this service.

Following the data driven philosophy of Lenovo Open Cloud, this service is integrated with the Config Repo Service in which a config file, called profile, defines the state of hardware, and the service applies the profile to configure the server or deploy the operating system.

5.2.6 Configuration and automation repository service

The Config repo service is the source-of-truth of configuration templates and scripts. All published automation scripts are saved and version-controlled by this service.

LOC implements a data-driven design whereby configuration templates are inputs to LOC automation. Examples are rebuilding a LOC service, rebuilding an entire LOC instance, creating a new VM, or creating a new network for a VLAN. A configuration template filled with a set of values is called an instance of that template, and these instances can also be managed by the Config repo service.

5.2.7 Validation service

The Validation service monitors the status of LOC hardware and software to keep LOC in a good shape. It can be customized to launch user tests that development and administration often require. It integrates seamlessly with the Config Repo Service and the Automation Service, bringing writing test code and running test code into a cohesive workflow.

Based on an ELK stack (elastic.co/elk-stack), the Validation service perform two important tasks:

- It integrates with the Config Repo Service, and runs all tests defined in the LOC validation test suite; it continuously monitors the health of LOC instance.
- It has pre-configured storage performance tests (based on SNIA PTS) that can be executed against any storage solution such as Red Hat Hyperconverged Infrastructure for Virtualization. See snia.org/tech_activities/standards/curr_standards/pts for more information.

5.2.8 OS Image service

The OS image service is the source-of-truth of OS images including VM images.

It is a common task to provision a bare-metal server with an operating system, or to create a VM using a cloud image or a VM snapshot. These OS images are large and do not change often. It is crucial that they are managed properly so their contents are well known and can be deployed in a deterministic manner.

Instead of relying on Internet resources to download these images on the fly, the OS image service provides a single point of management to maintain the images, whether for licensing purposes or to upgrade to a new release.

Table 5 lists the images are that preloaded with LOC.

Table 5. Preloaded Images

Filename	Description
rhel-server-7.4-x86_64-dvd.iso	For server provisioning
rhel-server-7.5-x86_64-dvd.iso	For server provisioning
Invgy_sw_lxca_204-2.1.0_kvm_x86-64.qcow2	Installer for the OS Deployment service
rhel-server-7.5-x86_64-kvm.qcow2	Size SMALL RHV virtual machine base image
rhel-server-7.5-x86_64-kvm-500G.qcow2	Size MEDIUM RHV virtual machine base image
cfme-rhev-5.10.0.18-1.x86_64.qcow2	Installer for the Single Pane of Glass service

5.2.9 Single pane of glass

This service is the LOC admin portal. It takes advantage of the RESTful APIs of the services and a flexible plug-in system, to provide a unified user experience through the single viewpoint of the Lenovo Open Cloud. All LOC platform management services are integrated, and some common actions such as creating a new VM resource can even be directly configured and launched from the service. The single pane of glass also has a RESTful API, which allows integration with other customer tools.

The integrated services are:

- hardware: by integrating with the “Server Configurator and OS deployment service”.
- software: by integrating with the “Software repository and life cycle management service”.
- virtual machines: by integrating with the “Hypervisor runtime service”.
- automation: by integrating with the “Automation service”.
- inventory: by integrating with the “Inventory planning service”.
- OS images: by integrating with the “OS Image service”.
- LOC health: by integrating with the “Validation service”.

5.2.10 Launchpad

Launchpad represents an isolated automation environment. Decoupling the automation scripts and where they are executed makes it flexible to compose different environments for different automation needs, such as version, library, and network requirement.

5.3 Storage management services

Storage management services provide additional capabilities to manage external storage outside of the hyperconverged cluster storage provided by Red Hat Hyperconverged Infrastructure for Virtualization.

5.3.1 Ceph dashboard

The Lenovo Ceph dashboard, developed by Lenovo, is the recommended management software for Ceph Storage and supports many monitoring and management functions for a Ceph cluster including:

- Cluster management including showing the topology and performance graphs
- Management of Ceph nodes, disks, network interfaces
- Management of OSDs
- Management of pools, placement groups and volumes
- Management of RADOS Block Devices (RDBs)
- Management of snapshots
- Back up of incremental data or full data to a remote Ceph cluster, a S3 cluster or an AWS S3 cluster.
- Alerts and alert history

See the Lenovo Reference Architecture for Red Hat Ceph Storage for more details: lenovopress.com/lp1147.

5.4 Cloud management services

Cloud management services provide additional capabilities to manage

5.4.1 Red Hat Cloud Director

The Red Hat OpenStack Platform uses Director as the toolset for installing and managing a production OpenStack environment, e.g. overcloud. Each overcloud instance requires a Director instance for installation, management and maintenance operations.

For Lenovo Open Cloud, the Red Hat OpenStack Platform Director is deployed as a VM in the management cluster. Additional information about using Director can be found in [Director Installation and Usage](#).

5.4.2 Cloud controllers

OpenStack controllers are nodes that provide administration, networking, and high availability for the OpenStack environment. The Controller nodes are deployed with a collection of interacting services that control compute, storage, and networking resources, such as OpenStack nova, horizon, neutron, glance, etc. and other software components like MariaDB, HAProxy and pacemaker.

Lenovo recommends three of these nodes deployed as VMs in the management cluster.

6 Operational model

This chapter describes the options for mapping the logical components of Lenovo Open Cloud (LOC) to Lenovo ThinkSystem servers and network switches. Initial deployment can start with three Red Hat Hyperconverged Infrastructure for Virtualization nodes.

6.1 LOC deployment

Lenovo Open cloud consists of a management cluster and zero or more workload clusters. This management cluster serves as virtualized control plane for LOC.

The user can then choose to activate the appropriate workload control planes to build the following types of workload clusters using additional servers:

- Red Hat Ceph Storage
- Red Hat OpenStack Platform

The value of LOC is to construct these infrastructures quickly and with a flexible configuration. It is common to have LOC build multiple OpenStack instances with different numbers of compute resources to cater for different user groups — a three-node OpenStack with virtual Ceph for development, and a larger deployment with hundreds of compute nodes and external Ceph storage for production.

In addition, it is also possible to build the management cluster using a bootstrapping process.

6.1.1 Hardware

The hardware and BOMs for the management cluster and workload clusters are described in the respective Lenovo Reference Architecture document and are not repeated here:

- Red Hat Hyperconverged Infrastructure for Virtualization - lenovopress.com/lp1148
- Red Hat Ceph Storage - lenovopress.com/lp1147
- Red Hat OpenStack Platform - lenovopress.com/lp0762
- Red Hat OpenShift Container Platform - lenovopress.com/lp0968

6.1.2 Licenses

By default, all platform management services are enabled and covered by proper licenses. Ceph management services and cloud management services are installed but disabled. This means that they cannot yet build Ceph and OpenStack workload clusters without first installing licenses. The minimum required licenses are provided in the following BOMs.

- To license Ceph only storage infrastructure, see “Three-node Ceph only software BOM”.
- To license OpenStack only infrastructure, see “Three-node OpenStack software BOM”.
- To license HCI Ceph and OpenStack infrastructure, see “Three-node HCI Ceph and OpenStack software BOM”.

6.2 LOC management cluster

Figure 5 shows the LOC management services deployed on Red Hat Hyperconverged Infrastructure for Virtualization cluster.

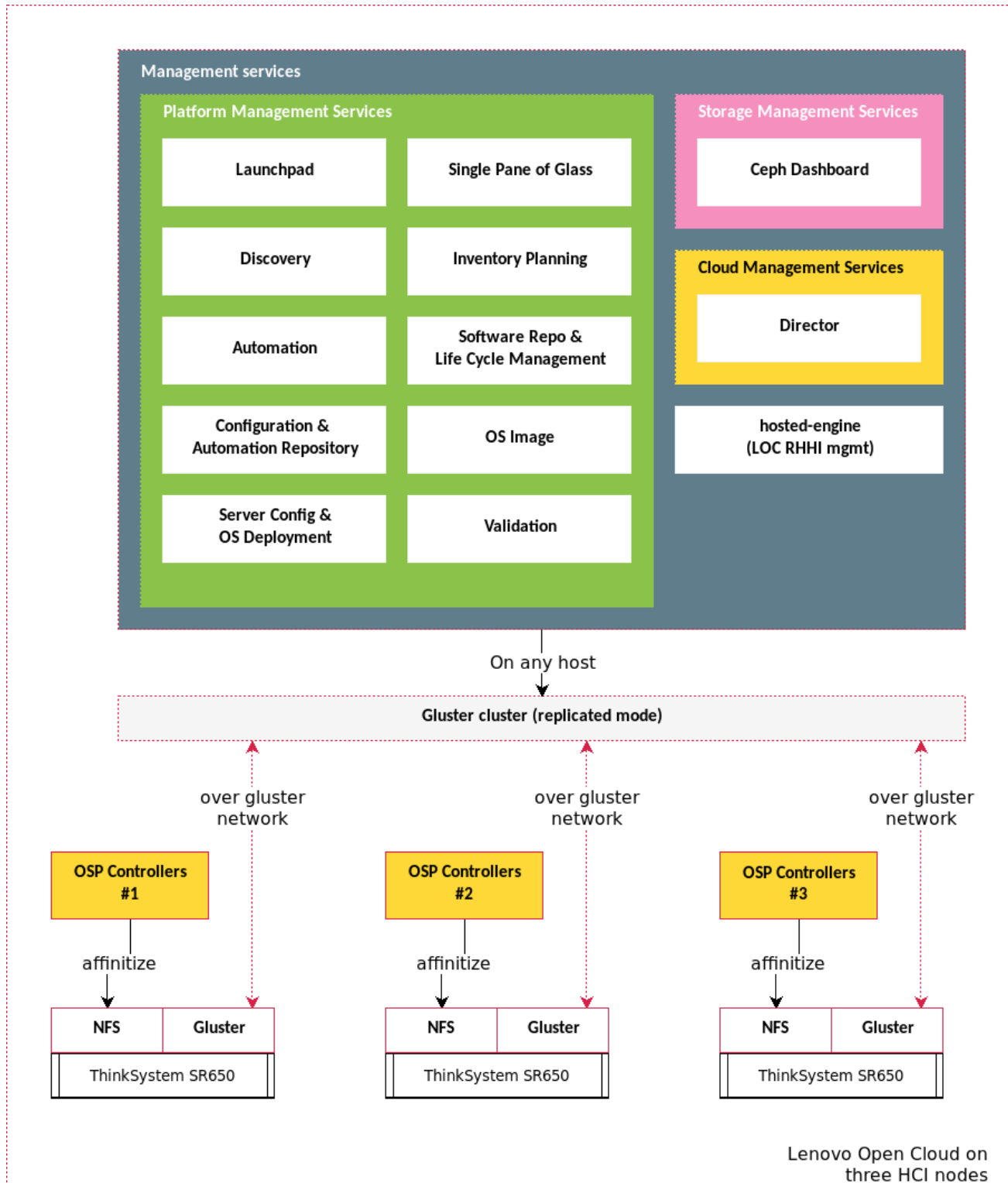


Figure 5: LOC service deployment on Red Hat Hyperconverged Infrastructure for Virtualization

6.2.1 LOC platform management service VMs

Table 6 lists the recommended VM resource requirements for the platform management services.

Table 6. VM resources for Platform Management Services

LOC Management Service	CPU Cores	Mem (GB)	Storage (GB)
Software repo	4	20	500
Automation	2	4	100
Server config & OS deployment	4	8	200
Inventory planning	2	4	50
Discovery	2	4	50
Config repo	2	4	100
Single pane of glass	4	8	100
Validation	4	8	300
OS Image	2	4	500
Launchpad	2	4	50

The primary storage for each VM is GlusterFS. To promote high availability and failover, these VMs should be configured to enable auto migration, boot from local drive, and should not use local NFS storage or have affinity to a single node.

Workload clusters also need management VMs. Refer to “Storage virtual control plane” and “OpenStack virtual control plane” for more details.

6.3 LOC workload clusters

A Lenovo Open Cloud deployment can have zero or more workload clusters that are used to execute different customer workloads. Using multiple clusters, customers can easily deploy isolated clouds with different configurations for various business purposes.

Building the infrastructure for a workload cluster follows a standard pattern:

2. Resource planning — servers, networks, roles, and tenants
3. Configure server — firmware, RAID, boot order
4. Provision OS
5. OS customization — cloud-init, package/subscription manager, network (remote administration)
6. Infrastructure installer
7. Infrastructure post-install configuration
8. Infrastructure functional testing and validation, performance/stress testing and health monitoring.

The platform management services are designed to fit this pattern using the following these steps:

1. Save hardware info in the Inventory service, such as manufacturer, model, and serial number.

2. User uses Inventory service to plan for hardware's usage by assigning rack position, tenant it belongs to, and its network assignment in terms of IP addresses, VLANs, and which interface on which subnet.
3. User assigns a tenant, role and status to a server in the Inventory service:
 - tenant: e.g. cloud xyz, meaning that this server will be a resource for this tenant. The type of resource is defined in role.
 - role: e.g. OpenStack compute that determines which configuration and which automation procedure is applied to the server.
 - status == available: marks the server as new resource.
4. Automation service calls the Discovery service to scan the XCC network, and uses detected server UUID to query the Inventory service for its role and status.
5. If found matching servers that are available, Automation service calls the Inventory service for server's XCC network assignments, and apply them to the Discovery service to set up OOB management of the server.
6. Automation service retrieves config patterns by the role from the Config repo service, and applies it to the OS Deployment service to configure server — firmware upgrade/downgrade, RAID, boot order.
7. Inventory service retrieves server component information such as disks, RAID controller, network controller, firmware, as part of server's meta data.
8. Automation service retrieves OS config by the role from the Config repo service, and applies to the OS Image to setup OS image to use.
9. Automation service retrieves network assignment the server, such as IP addresses, root user name, password, from the Inventory service, and calls the OS Deployment service to provision the server, including cloud-init to set up user accounts. By this point, the server is provisioned and can be accessed using SSH.
10. Automation service optional registers the new server to the Software repo and life cycle management service if the server is to use this service as software package upstream.
11. Automation service changes server's status to *provisioned* in the Inventory service.
12. Automation retrieve infrastructure config files by the tenant from the Config repo.
13. Automation service starts a pre-defined workflow to install infrastructure or to extend existing infrastructure with these servers.
14. Automation service sets servers' status to in-use.
15. Automation service updates tenant's infrastructure config files in the Config repo.
16. Automation service updates the Inventory service w/ tenant config ID, type, and default test cases. The user can activate additional test cases for the infrastructure in the Inventory service. The test cases can set to run once or CRON.
17. Automation service retrieves infrastructure configs from the Config service, and passes it to the Validation service to run validation tests as sanity check.
18. Automation service retrieves test cases by tenant from the Inventory service, and calls Validation service to run tests.
19. For new infrastructure, the Automation service registers the infrastructure in the Single pane of glass.
20. Validation service runs CRON tests, which continuously monitor the health of the infrastructure.

Each new workload cluster receives a "cloud index" which allows support for multiple instances.

6.4 Building a storage workload cluster

The requirements for storage infrastructure in the cloud can vary widely depending on the application. LOC storage has flexible options that easily support a variety of needs. Because LOC is built upon Red Hat and OpenStack technologies, many different storage infrastructure options are available but this release supports both Gluster and Ceph storage. This section focusses on deploying a Ceph storage cluster and shows how the integration of storage is simplified.

6.4.1 Storage virtual control plane

The Lenovo Ceph dashboard monitors the status of Ceph cluster. It is a VM deployed in LOC management cluster. Table 7 lists the recommended VM resource requirements for the storage management services.

Table 7. VM resources for Storage Management Services

LOC Management Service	CPU Cores	Mem (GB)	Storage (GB)
Ceph dashboard	2	8	60

This VM should be configured to enable auto migration, boot from local drive, and should not use local NFS storage or have affinity to a single node.

6.4.2 Deploying a new Ceph cluster

Lenovo Open Cloud uses Red Hat Ceph Storage to provide scalable, open, software-defined storage to Red Hat OpenStack Platform. Like Gluster storage, Red Hat Ceph Storage is built from a number of Ceph storage nodes to provide fault-tolerant storage.

For more details on the Lenovo Reference Architecture for Red Hat Ceph Storage, see lenovopress.com/lp1147. For more details on Red Hat Ceph Storage integration into Red Hat OpenStack Platform, see access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/.

Figure 6 shows an overview of deploying a new Ceph cluster.

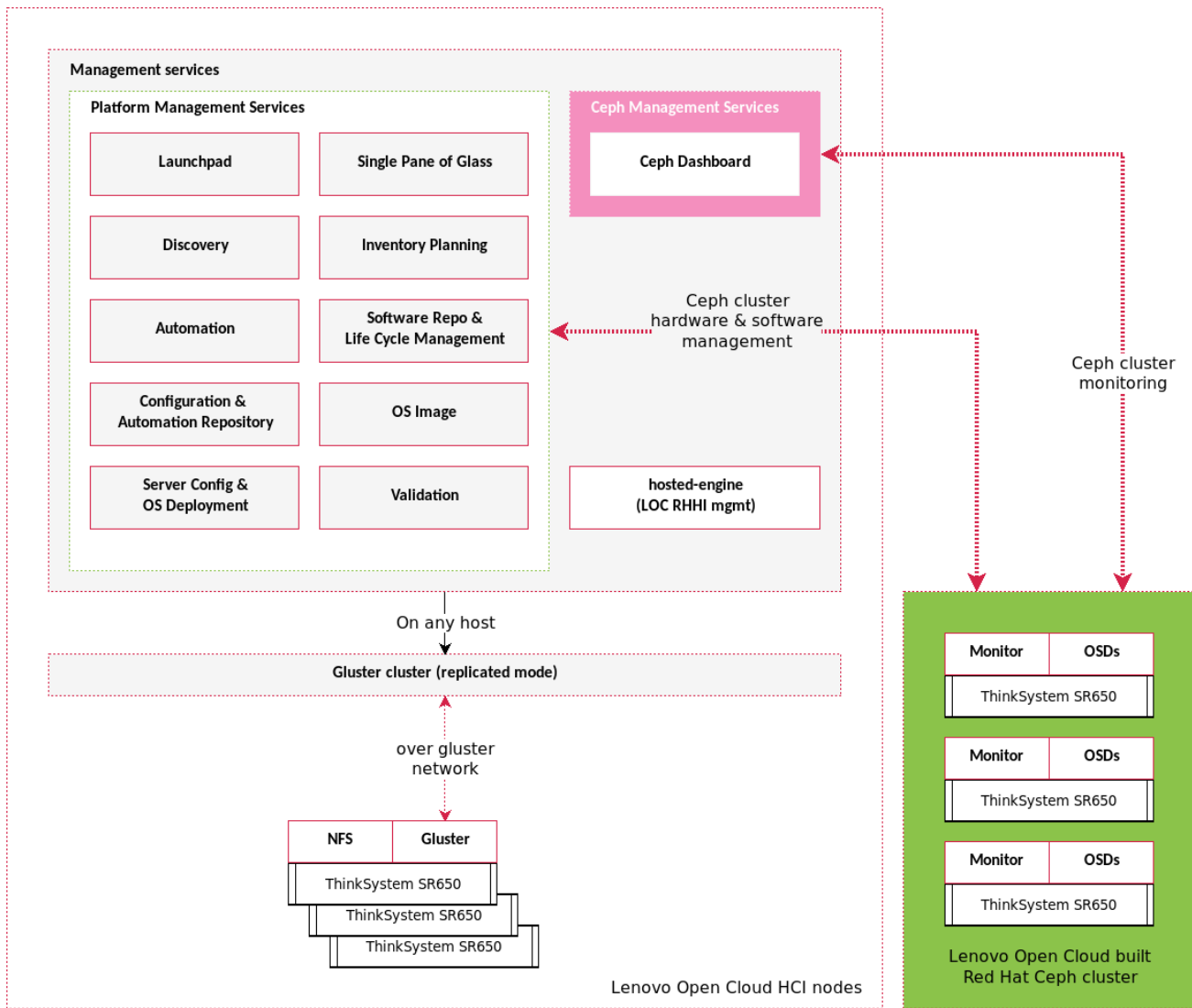


Figure 6: Deploying a new Ceph cluster

The deployment steps to create a Ceph cluster are as follows:

1. The Automation service can trigger an automated workflow on demand to create a Ceph cluster. The automation workflow pulls in the necessary configuration files from the Config repo to determine the size, disk configuration, monitor placement, and target OpenStack cloud for the requested cluster. The automation workflow then requests the Runtime service to create a Ceph Dashboard VM in the LOC management cluster. The Runtime service also requests bare-metal resources from the Inventory service to serve as the Ceph storage nodes.
2. Once the VM and bare-metal resources are ready, each storage node has an operating system deployed and the node's network interfaces and VLANs are configured.
3. After configuring each node as a Ceph storage node, the automation then enables the appropriate software repos to support installing Red Hat Ceph Storage. The binaries and libraries for Ceph support are installed and Ceph is deployed using the disks requested via the config repo in each node, creating a cluster across the storage nodes.

4. Once Lenovo's storage automation has verified that the Ceph cluster has been properly deployed, it creates the storage pools for each of the OpenStack components, which require Ceph storage. Ceph storage pools follow the cloud numbering convention so that a Ceph pool can be easily associated with a cloud. The following pools are created by default where "x" is the cloud number:
 - cloudx_nova - backs ephemeral storage for Nova instances to enable VM migration
 - cloudx_volumes - storage for the Cinder volume service
 - cloudx_images - storage for the Glance image service
 - cloudx_backups - storage to be utilized for backing up volumes
 - cloudx_metrics - storage utilized by OpenStack's Telemetry Service
5. After the pools have been created, the storage automation creates a client with a name that is indexed on the cloud number. For example, cloud 1 has the cloud1.client user created. A key is created for accessing the cluster and the appropriate authorities are configured for the cloud to access each pool.
6. All of the information necessary for accessing the newly created cluster is then added to the Config repo and may be accessed, based on the cloud index. This information is used to deploy a cloud.
7. The final step in deploying a Ceph cluster is to add the cluster to the Ceph Dashboard for monitoring using a scripted discovery process.

6.4.3 Extending an existing Ceph cluster

To extend an existing Ceph cluster much of the same process is followed as when creating a new Ceph cluster. The Runtime service will pull the updated configuration files from the config repo. Once the files are processed, a bare-metal resource is requested from the Inventory service. The new bare-metal resource is prepared to be a Ceph node and the node is then added into the Ceph cluster.

After the node has been successfully integrated, the storage is then rebalanced across all the nodes in the Ceph cluster, and the Ceph Dashboard is updated. Note, when extending a Ceph cluster, the pool and client creation steps do not need to be repeated.

6.5 Building a Red Hat OpenStack Platform workload cluster

Red Hat OpenStack Platform 13, based on the OpenStack Queens release, provides an integrated solution to create, deploy, and scale a secure and reliable OpenStack cloud. It is packaged so that available physical hardware can be turned into a private, public, or hybrid 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
- Web browser-based interface accessible to users and administrators

Lenovo Open Cloud supports deploying a workload cluster of Red Hat OpenStack Platform 13 with various storage configurations. The controller nodes can be installed to either bare-metal nodes or virtual machines (VMs) spread across multiple physical servers.

Deploying the OpenStack cloud controller nodes as VMs decreases cost and provides users with a scalable cloud-ready infrastructure with much easier operational management. The virtualized control plane of Open Stack cloud is easy to rebuild, maintain and replace. Virtualized controller node deployment is most suitable for medium scale clouds of less than forty nodes.

6.5.1 OpenStack deployment configurations

LOC supports three OpenStack deployment configurations:

- With Swift co-located storage (see Table 8)
- With an external Ceph storage cluster (see Table 9)
- With a hyperconverged Ceph storage cluster (see Table 10)

The next three tables list the server requirements for the three deployment options.

Table 8. Server requirements for Swift co-located storage configuration

Description	VM or bare-metal	Quantity
Director	VM	1
Cloud Controller with Swift service co-located (HA)	VM	3
Cloud compute	bare-metal	1

Table 9. Server requirements for external Ceph storage cluster configuration

Description	VM or bare-metal	Quantity
Director	VM	1
Cloud Controller (HA)	VM	3
Cloud compute	bare-metal	1
Ceph storage node	bare-metal	3

Table 10. Server requirements for hyperconverged Ceph storage cluster configuration

Description	VM or bare-metal	Quantity
Director	VM	1
Cloud Controller (HA)	VM	3
Hyperconverged Ceph node	bare-metal	3

6.5.2 OpenStack virtual control plane

Table 11 lists the recommended VM resource requirements for the OpenStack management services.

Table 11. VM resources for OpenStack Management Services

LOC Management Service	CPU Cores	Mem (GB)	Storage (GB)
Director	4	16	80
Cloud controllers (HA)	8	16	100
Cloud Controller with Swift service co-located (HA)	8	16	100

Table 12 lists the recommended additional attributes each VM.

Table 12. VM Attributes for OpenStack Management Services

LOC Management Service	Local NFS (GB)	Boot Options	Affinity (Y/N)	Enable Auto Migration
Director	N/A	Local drive	No	Yes
Cloud controllers (HA)	N/A	PXE as primary, and local drive as secondary	Yes	No
Cloud Controller with Swift service co-located (HA)	50 minimum	PXE as primary, and local drive as secondary	Yes	No

Auto migration is disabled and host affinity is enabled for the OpenStack cloud controller VMs. This ensures that all three cloud controller VMs execute on different physical hosts in the Management cluster to provide high availability. The cloud controllers need to be PXE booted for initial deployment from Director.

The hyperconverged Management cluster has Red Hat Gluster storage that is used by the majority of the VMs executing in the management cluster. Each node in the management cluster also has NFS storage space that is used by some workload management VMs such as the Cloud Controller with Swift service. This is because Swift already has built-in replica support and using the NFS local, Swift-replicated storage avoids a double replication penalty.

6.5.3 Deploy a Red Hat OpenStack Platform cluster

Figure 7 shows an overview of deploying a new Red Hat OpenStack Platform cluster with external Ceph storage.

The deployment steps to create an OpenStack cluster are as follows:

1. The automation service can trigger a cloud automation workflow on demand to deploy an OpenStack cloud. It loads the scripts and heat-templates required by Director from Config repo service. Config repo service also provides the platform service config that contains information on the endpoint info of platform services, and input from the system administrator for the expected cloud configuration, and VM and bare-metal requirements for cloud nodes.
2. The cloud deployment workflow then requests the Runtime service to create and configure the necessary VMs for the cloud controllers and Director. In parallel, the workflow uses the Inventory service to allocate and provision the bare-metal compute nodes for the cloud.

- After all VM and bare-metal resources are ready, Director is used to deploy the OpenStack cloud using the rendered heat-templates to create the expected cloud configuration.

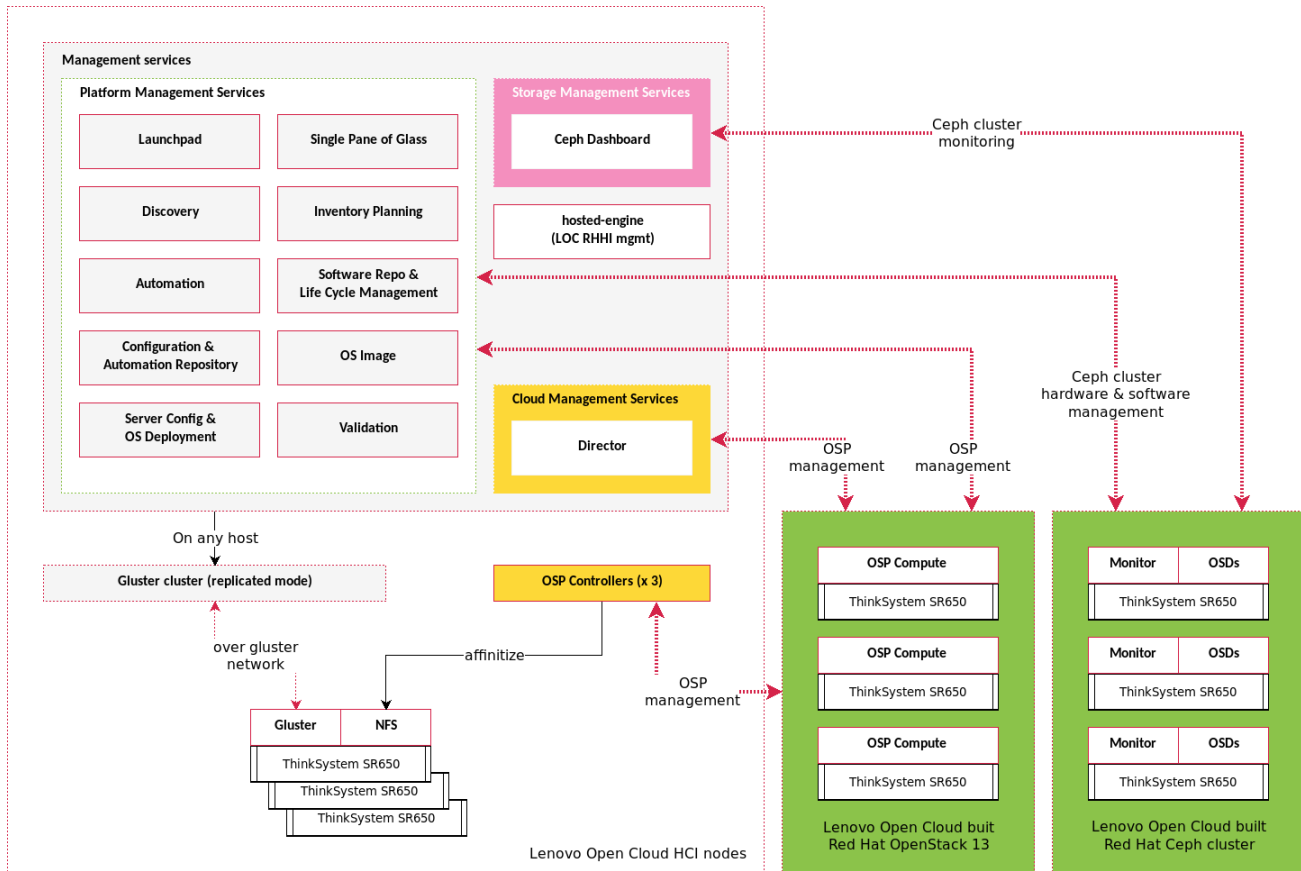


Figure 7: Deploying a new Red Hat OpenStack Platform cluster with external Ceph storage

6.6 Networking

The networking architecture is an important part of the Lenovo Open Cloud deployment. This section describes both the management cluster networking and the networking needed for the various workload clusters including storage and OpenStack.

6.6.1 Management cluster networking

This section describes the logical networks and VLANs used in the management cluster. These are the same as those used by the underlying Red Hat Hyperconverged Infrastructure for Virtualization cluster and are summarized in Table 13.

Table 13. VLAN mapping of logical networks

Logical Network	VLAN	Description
Campus internal	1	For public access of admin portal and API
OOB	2	OOB management
Physical server management	3	SSH admin access
Provisioning	10	New bare-metal server provisioning network
OVIRT management	100	Management portal and API
GlusterFS	400	Hyperconverged storage

Table 14 shows the VLANs needed by each of the services in the management cluster. VLAN 400 is reserved for the hyperconverged storage logical network. The numbers refer to the numbered notes after the table.

Table 14. Mapping of platform management services to VLANs

Service	VLAN				
	1	2	3	10	100
Runtime	1				3
Software repo	1		8		11
Automation	1		8		
Discovery	1	4			
Inventory	1, 2	5	9		
OS deployment	1	6			
Configuration repo	1				
Validation	1	7	10		12
OS Image	1				
Single pane of glass	13				11

1. Serving admin portal and its API endpoint
2. Accessing LOC switches for port configuration and cable schema mapping
3. Managing management cluster hyperconverged servers
4. Monitoring new bare-metal server
5. Out-of-band server query into XCC for building inventory attributes and cable schema
6. Server out-of-band management and OS provisioning
7. Run tests via XCC
8. Remote administration of physical servers
9. Query server networking interfaces for cable schema mapping
10. Execute tests via remote access such as SSH
11. Integration point with Runtime service

12. Run tests such as adding a new hardware resource
13. Integration point to other services except the Runtime service

6.6.2 Storage Networking

Lenovo Open Cloud supports Red Hat Ceph Storage workload clusters. A storage cluster can be shared among multiple workload clusters. Storage logical networks are categorized into two groups:

- Storage specific networks: Only applicable for the storage instance.
- Shared platform networks: Extensions of the LOC management cluster networks to support storage infrastructure components that are deployed on the Management cluster and/or using the platform management services.

Figure 8 shows the different networks and their relationship to the LOC Management cluster networks.

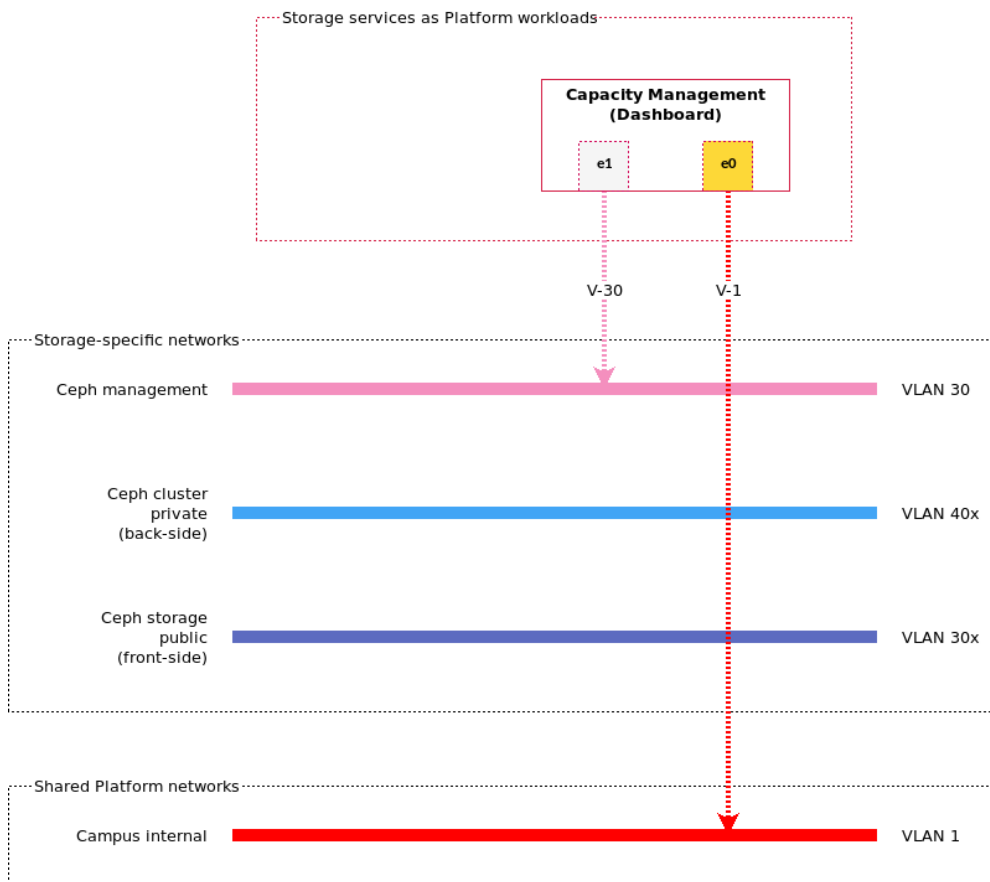


Figure 8: LOC storage networks

Ceph specific logical networks

Ceph adds three logical networks:

- Ceph management For communication between the Ceph dashboard and Ceph nodes, e.g. RPC and transferring *zabbix* monitoring data.

- Ceph storage public For storage data traffic by workloads using Ceph for storage. This is also known as front-side.
- Ceph cluster private For Ceph private data traffic used in data rebalancing, for example. This is also known as back-side.

Table 15 lists the Ceph specific VLANs corresponding to the Ceph specific logical networks described above where “x” represents the workload cluster index.

Table 15. Ceph specific VLANs

Logical Network	VLAN
Ceph management	30
Ceph storage public	30x
Ceph cluster private	40x

Shared platform networks

Table 16 lists the management cluster logical networks that are used by a Ceph storage workload cluster.

Table 16. Usage of shared logical networks by Ceph storage

Platform Logical Networks	VLAN	Ceph Dashboard (VM)	Ceph storage nodes (bare-metal)
Campus internal	1	Assess UI	N/A
OOB	2	N/A	OOD management and discovery
Physical server management	3	N/A	Admin tasks and SSH access
Provisioning	10	N/A	Storage node OS provisioning

Red Hat Virtualization networks for Ceph storage workload clusters

Two Red Hat Virtualization networks are created to support Ceph specific functions including compute node provisioning, and cloud management.

- Ceph management As defined in “Ceph specific logical networks”, this network is mapped onto the LOC Platform so that the Ceph Dashboard can monitor the Ceph cluster.
- Cloud all A general bucket to support all cloud management logical networks. This includes making the Ceph storage public network available for the management applications.

6.6.3 OpenStack cloud networking

Lenovo Open Cloud supports Red Hat OpenStack Platform workload clusters. Cloud logical networks are categorized into three groups:

- Cloud specific networks: Only applicable to the OpenStack cloud instance.

- Shared storage networks: To utilize a storage backend such as Ceph.
- Shared platform networks: Extensions of the LOC management cluster networks to support storage infrastructure components that are deployed on the Management cluster and/or using the platform management services.

Figure 8 shows the different networks and their relationship to the LOC Management cluster networks.

Cloud services deployed as Platform workloads

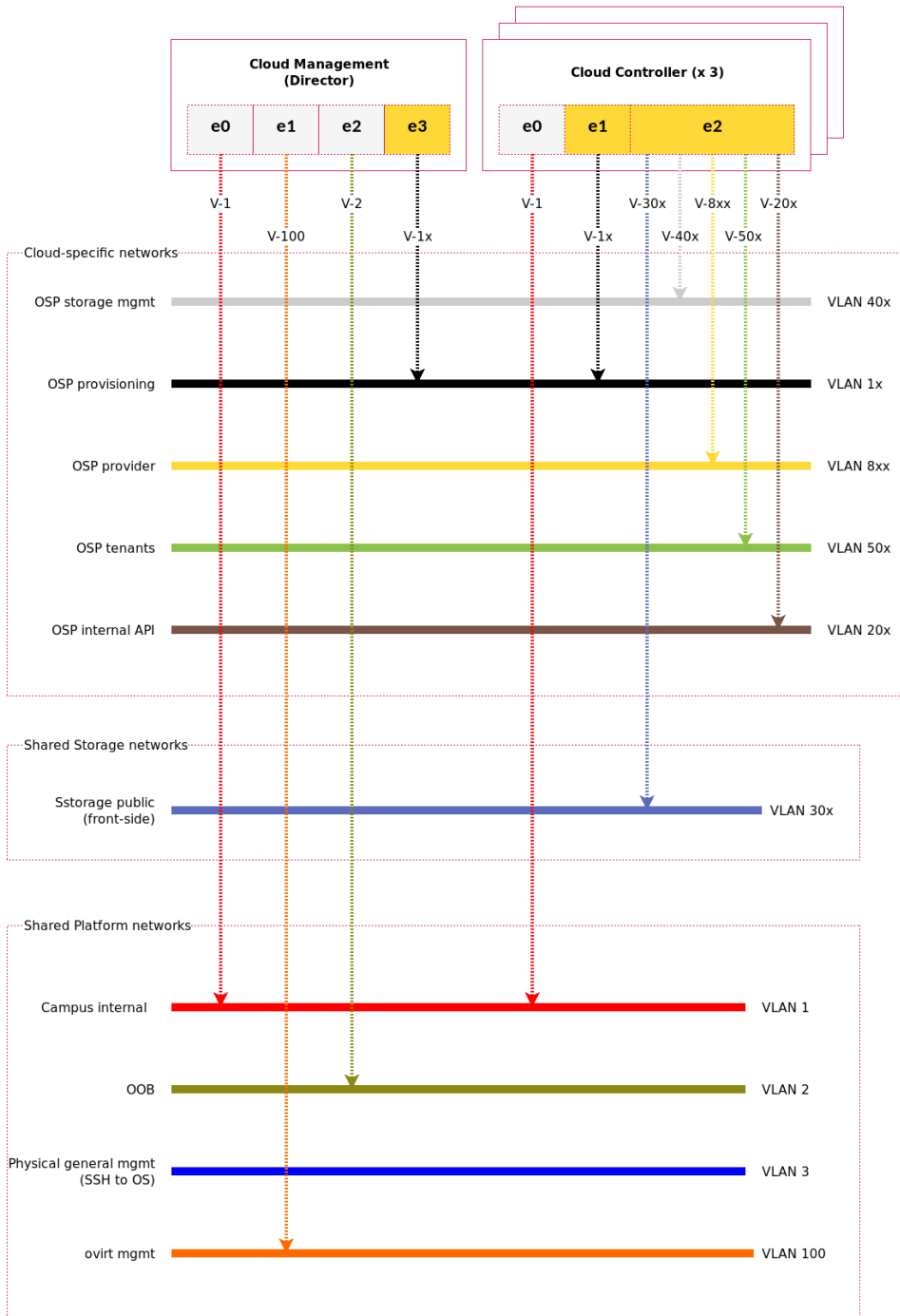


Figure 9: LOC OpenStack cloud networks

Cloud specific logical networks

OpenStack cloud adds six logical networks:

- **OSP provisioning** Provides DHCP and PXE boot functions to help discover bare-metal and provision servers for use in the OpenStack cloud. All cloud nodes use this network.
- **OSP tenants** Provides a private network for the VMs to talk to each other using VM private IP addresses. Cloud controller and compute nodes use this network.
- **OSP storage mgmt** OpenStack object storage (Swift) uses this network to synchronize data objects between participating replica nodes.
- **OSP internal API** The internal API network is used for communication between the OpenStack services using API communication, RPC messages, and database communication. Used by cloud controller and compute nodes.
- **OSP provider** This is the provider network used for VM traffic on a dedicated network. Used by controller and compute nodes.
- **Ceph storage** This is the front-side storage network where Ceph clients (through Glance API, Cinder API, or Ceph CLI) access the Ceph cluster. Ceph Monitors operate on this network. Used by cloud controller and compute nodes.

Table 17 lists the OpenStack cloud specific VLANs corresponding to the cloud specific logical networks described above where “x” represents the workload cluster index and “b” is the VLAN index assigned for the provider inside a cloud instance.

Table 17. OpenStack cloud specific VLANs

Logical Network	VLAN
OSP provisioning	1x
OSP internal API	20x
Ceph storage	30x
OSP storage mgmt	40x
OSP tenants	50x
OSP provider	8xb

Shared storage logical networks

OpenStack cloud infrastructure can take advantage of external storage solutions such as Ceph for its storage backend. This allows storage to be centralized and shared by different OpenStack workload clusters. Table 18 lists the shared storage logical networks.

Table 18. OpenStack cloud spec

Storage Logical Networks	VLAN	Director (VM)	Controller (VM)	Compute (bare-metal)
Storage public	30x	N/A	Yes	Yes

Storage public is the Ceph front-side logical network that serves storage data traffic. Note that the storage public network should be connected to 10Gb or faster ports on the cloud compute nodes.

Shared platform logical networks

All LOC based cloud deployments have the Director and cloud controllers in VMs. This means that these VMs must be connected to the Management cluster platform networks in order to use “Platform management services” for both initial setup and for life cycle management such as software upgrade and fixes.

Table 16 lists the management cluster logical networks that are used by an OpenStack cloud workload cluster.

Table 19. Usage of shared logical networks by OpenStack cloud

Platform Logical Networks	VLAN	Director (VM)	Controller (VM)	Compute (bare-metal)
Campus internal	1	Yes	Yes	N/A
OOB	2	Yes	N/A	Yes
ovirt management	100	Yes	N/A	N/A

Red Hat Virtualization networks for OpenStack cloud workload clusters

Two Red Hat Virtualization networks are created to support OpenStack cloud specific functions including compute node provisioning, and cloud management.

- **OSP Provisioning** As defined in “Cloud specific logical networks”, this network is mapped onto the LOC Platform so that the Director and cloud controllers can initiate and control compute node provisioning.
- **Cloud all** A general bucket to support all cloud management and some service data traffic that need to be through the Management cluster. This includes:
 - OSP internal API
 - OSP storage management
 - OSP tenants
 - OSP providers
 - Ceph storage

Figure 10 shows how these networks are used on the physical NICs of a LOC management cluster node.

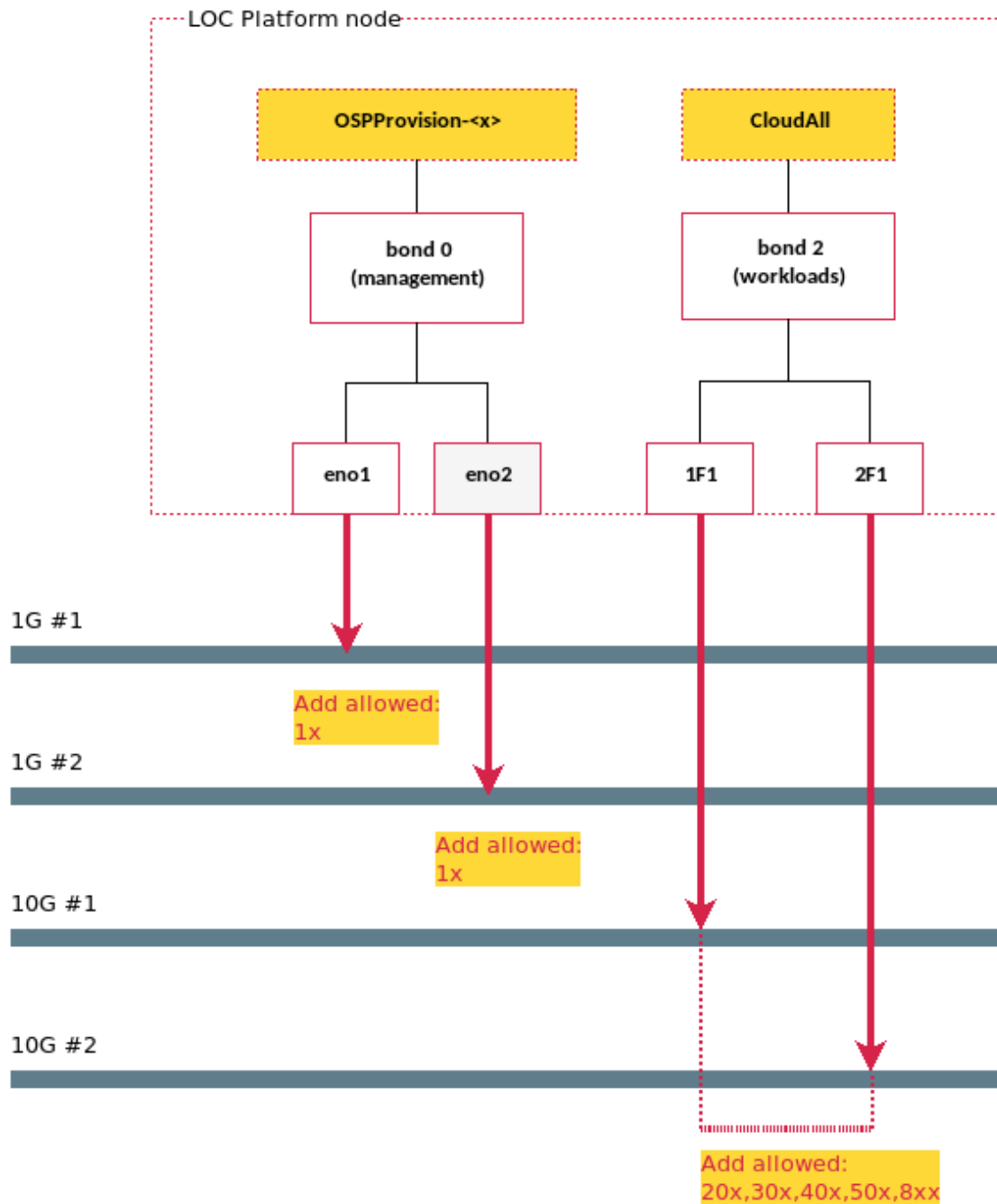


Figure 10: Added Platform networks for OpenStack cloud workload clusters

This design simplifies integration between the LOC Management cluster and the OpenStack cloud workload clusters by keeping these cloud-driven decisions internal to workload designers.

7 Appendix A: Lenovo bill of materials

This appendix contains the Bill of Materials (BOMs) for the different hardware and software configurations supported for the Lenovo Open Cloud deployments.

7.1 Management cluster server BOM

This is the server BOM for nodes in Lenovo Open Cloud management cluster where each table represents a single node. To order N nodes to construct the cluster, just multiply the “Quantity” column by “N”. The tables below are generated using the online Lenovo [Data Center Solution Configurator](#).

See the Lenovo Red Hat Hyperconverged Infrastructure for Virtualization lenovopress.com/lp1148 for more details on the hardware BOMs for the management cluster.

Table 20. Configuration for Lenovo Open Cloud management cluster node

Code	Description	Quantity
7X06CTO1WW	Storage-development-mode-1-2.5" : ThinkSystem SR650 - 3yr Warranty	1
AUVV	ThinkSystem SR650 2.5" Chassis with 8, 16 or 24 bays	1
AX6D	Intel Xeon Gold 6130 16C 125W 2.1GHz Processor	2
AUND	ThinkSystem 32GB TruDDR4 2666 MHz (2Rx4 1.2V) RDIMM	24
AUR5	ThinkSystem 2U/Twr 2.5" AnyBay 8-Bay Backplane	1
AURA	ThinkSystem 2U/Twr 2.5" SATA/SAS 8-Bay Backplane	2
5977	Select Storage devices - no configured RAID required	1
B31E	ThinkSystem RAID 930-16i 8GB Flash PCIe 12Gb Adapter	2
B49M	ThinkSystem 2.5" Intel S4610 480GB Mainstream SATA 6Gb Hot Swap SSD	2
B4Y4	ThinkSystem 2.5" SS530 400GB Performance SAS 12Gb Hot Swap SSD	2
B0YS	ThinkSystem 2.5" 2.4TB 10K SAS 12Gb Hot Swap 512e HDD	18
AURC	ThinkSystem SR550/SR590/SR650 (x16/x8)/(x16/x16) PCIe FH Riser 2 Kit	1
AUKH	ThinkSystem 1Gb 4-port RJ45 LOM	1
AUKX	ThinkSystem Intel X710-DA2 PCIe 10Gb 2-Port SFP+ Ethernet Adapter	2
5053	SFP+ SR Transceiver	4
AVWF	ThinkSystem 1100W (230V/115V) Platinum Hot-Swap Power Supply	2
6400	2.8m, 13A/100-250V, C13 to C14 Jumper Cord	2
AUPW	ThinkSystem XClarity Controller Standard to Enterprise Upgrade	1
B0MJ	Feature Enable TPM 1.2	1
AUQB	Lenovo ThinkSystem Mainstream MB - 2U	1
AXCH	ThinkSystem Toolless Slide Rail Kit with 2U CMA	1

7.2 Three-node RHHI-V cluster software BOM

SKU	Product Description	Quantity
RS00139F3	Red Hat Hyperconverged Infrastructure for Virtualization (RHHI-V)	1
RH00031F3	Smart Management	3
MCT3475F3	Red Hat Insights	1
00MT202	Lenovo XClarity Enterprise	3

7.3 Three-node HCI Ceph and OpenStack software BOM

SKU	Product Description	Quantity
RS00032F3	Red Hat Ceph Storage	1
MCT2982F3	Red Hat OpenStack Platform (w/o Guest) with Smart Management	4
MCT2980F3	Red Hat OpenStack Platform with Smart Management & Guests	3
00MT202	Lenovo XClarity Enterprise	3

7.4 Three-node Ceph only software BOM

SKU	Product Description	Quantity
RS00032F3	Red Hat Ceph Storage	1
00MT202	Lenovo XClarity Enterprise	3

7.5 Three-node OpenStack software BOM

SKU	Product Description	Quantity
MCT2982F3	Red Hat OpenStack Platform (w/o Guest) with Smart Management	4
MCT2980F3	Red Hat OpenStack Platform with Smart Management & Guests	3
00MT202	Lenovo XClarity Enterprise	3

Resources

For more information about the topics in this document, see the following resources:

- Lenovo Reference Architecture for Red Hat OpenStack Platform:
lenovopress.com/lp0762
- Lenovo Reference Architecture for Red Hat Hyperconverged Infrastructure for Virtualization:
lenovopress.com/lp1148
- Lenovo Reference Architecture for Red Hat Ceph Storage:
lenovopress.com/lp1147
- Lenovo Reference Architecture for Red Hat OpenShift Container Platform:
lenovopress.com/lp0968
- Lenovo OpenStack Reference Architecture for Service Providers:
lenovopress.com/lp0913
- Lenovo Data Center Solution Configurator:
dcsc.lenovo.com

Acknowledgements

A special thank you to the following Lenovo colleagues for their contributions to this document:

- Wei Chen, Engineer — Data Center Group
- Chengzhen Chu, Engineer — Data Center Group
- Guannan Sun, Engineer — Data Center Group
- Ricky Stambach, Engineer — Data Center Group
- Zelin Wang, Engineer — Data Center Group

Trademarks and special notices

© Copyright Lenovo 2019.

References in this document to Lenovo products or services do not imply that Lenovo intends to make them available in every country.

Lenovo, the Lenovo logo, ThinkSystem, ThinkCentre, ThinkVision, ThinkVantage, ThinkPlus and Rescue and Recovery are trademarks of Lenovo.

Red Hat, Red Hat Enterprise Linux and the Shadowman logo are trademarks of Red Hat, Inc., registered in the U.S. and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. The OpenStack mark is either a registered trademark/service mark or trademark/service mark of the OpenStack Foundation, in the United States and other countries, and is used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used Lenovo products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-Lenovo products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by Lenovo. Sources for non-Lenovo list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. Lenovo has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-Lenovo products. Questions on the capability of non-Lenovo products should be addressed to the supplier of those products.

All statements regarding Lenovo future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local Lenovo office or Lenovo authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in Lenovo product announcements. The information is presented here to communicate Lenovo's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard Lenovo benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-Lenovo websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Lenovo product and use of those websites is at your own risk.