Lenov Open Cloud Platform Network Reference Architecture

IBB Platform Team

Abstract

Lenovo Open Cloud consists of a list of physical servers (aka. nodes) and virtual machines (VMs). This reference environment provides a comprehensive example demonstrating how to set up networks to connect these servers and VMs together.

Contents

List of Tables

List of Figures

Title: Setting Up Lenovo Open Cloud Networks Date: 2018-09-14 09:35 Tags: reference architecture Slug: ibb network design Author: Feng Xia Status: Draft

1 Introduction

The target audience for this Reference Architecture (RA) is system administrators or system architects. Some experience with Red Hat implementation of virtualization, shared storage, and OpenStack is helpful, but it is not required.

Technology evolves fast. We have seen the wave of hardware virtualization in both server space and in personal computing. Then came cloud computing, in which infrastructure becomes even more abstract and remote to end user than ever before. Instead of being viewed as brick and mortar, server, storage, networking are **resources** that can be requested, leased for a period of time, paid per use, and releases when done — all through nothing but an online account and a credit card. The flexibility of this model and the feeling that resource pool can be extended boundlessly has both lowered barrier of entry of new application growing from zero to infinity with little sweat, and elevated requirement on the design, implementation, and operation of such infrastructure

Further more, along the trail of technology evolution, business has been left with an army of legacy systems which were designed and built on a technology stack that was adequate then, but not in trend now. Millions have been invested, millions of users are probably depending on the continuity of service, and many developers and operators were trained and are given the responsibility to maintain such stack. It is neither feasible to cut the cord just because a new technology becomes the talk of the day, nor advisable to continue as before without taking advantage of what new tools can bring. Therefore, it is not only desirable, but in our opinion essential to have an infrastruture that is both flexible and balanced — it must support a broad range of user and application by providing a platform that has a rich mix of building blocks which, first of all, covers common needs out of box, such as keeping an operating system up to date via patch, update and hotfix, while maintaining an open architecture to extend both horizontally in term of resource (compute, storage, networking), and vertically (application stack).

It is with this in mind that Leonov Open Cloud is designed to combine the best of technologies in the market today into a coherent user experience while all the following users will feel at home:

1. **VM users**: Open Cloud supports hardware virtualization in its core. Traditional virtual machines users and applications can be migrated onto the platform while minimizing dependency on underline hardware environment.

2. **Cloud users**: Open Cloud provides on-premise cloud computing environment based on OpenStack, the leading cloud operating system.

3. **Container users**: Devops have continuously pushed the boundary to merge development and production into a single, consistent experience that what developer uses as a `sandbox` should be identical as what can be used in production. By doing so not only we will eliminate the necessity to maintain multiple stacks catering for different environments — a typical setup will be one for development, one for testing, and one for production, but minimize chance of incompatibility and bugs due to difference between two environments.

This RA describes the system architecture for the Lenovo Open Cloud Platform based on Lenovo ThinkSystem servers and Lenovo network switches. It provides detail of the hardware requirements to support various node roles and the corresponding configuration of the systems. It also describes the network architecture and details for the switch configurations. The hardware bill of materials is provided for all required components to build the Open Cloud cluster. An example deployment is used to show how to prepare, provision, deploy, and manage the Open Cloud on Lenovo ThinkSystem servers and Lenovo network switches.

2

Business problem and business value

2.1 Business problem

2.2 Business value

3
Architecture Overview

Lenovo Open Cloud has two sets of clusters: `management cluster` and `workload cluster`. Workload cluster refers to applications directly interfacing with end user. Management cluster refers to applications that manage and provide Open Cloud services.

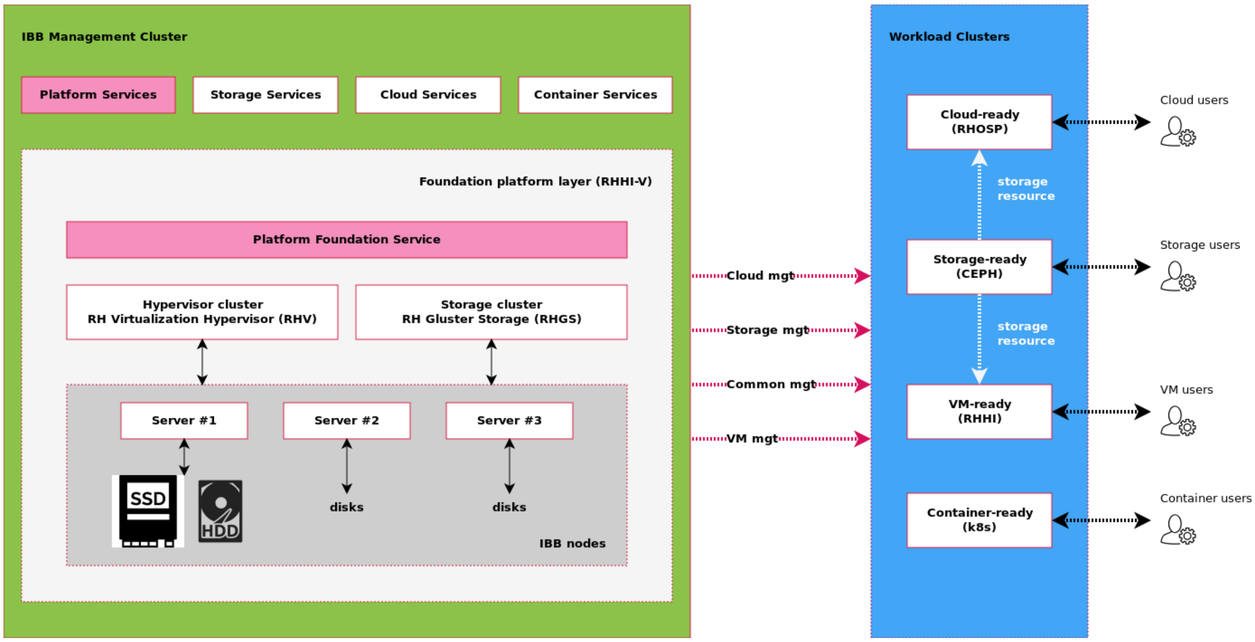

Figure 3.1: Lenovo Open Cloud Architecture

Management cluster includes a `platform foundation layer` and four groups of service.

Platform foundation Foundation layer includes 3-, 6-, or 9- servers depending on configuration, storage disks inside server, two 1Gb switches, and two 10Gb switches. It supports virtual machines on top of Red Hat Virtualization. Storage uses a Gluster FS cluster that spans across all servers.

4

Hardware

Lenovo Open Cloud is a highly configurable system. From the point of view of physical servers, LOC can be deployed on a 6-server or 9-server configuration. In this document we will use a 9-server configuration as example.

4.1 Platform servers

The Lenovo ThinkSystem SR650 server is an enterprise class 2U two-socket versatile server that incorporates outstanding reliability, availability, and serviceability (RAS), security, and high efficiency for business-critical applications and cloud deployments. Unique Lenovo AnyBay technology provides the flexibility to mix-and-match SAS/SATA HDDs/SSDs and NVMe SSDs in the same drive bays. Four direct-connect NVMe ports on the motherboard provide ultra-fast read/writes with NVMe drives and reduce costs by eliminating PCIe switch adapters. Plus, storage can be tiered for greater application performance, to provide the most cost-effective solution.

Combined with the Intel® Xeon® Scalable processors product family, the Lenovo ThinkSystem SR650 server offers a high density combination of workloads and performance. Its flexible, pay-as-you-grow design and great expansion capabilities solidify dependability for any kind of virtualized workload, with minimal downtime. Additionally, it supports two 300W high-performance GPUs and ML2 NIC adapters with shared management.

The Lenovo ThinkSystem SR650 server provides internal storage density of up to 100 TB (with

Figure 4.1: Lenovo ThinkSystem SR650

up to 26 x 2.5-inch drives) in a 2U form factor with its impressive array of workload-optimized storage configurations. The ThinkSystem SR650 offers easy management and saves floor space and power consumption for the most demanding storage virtualization use cases by consolidating the storage and server into one system. The Lenovo ThinkSystem SR650 server supports up to twenty-four 2.5-inch or fourteen 3.5-inch hot-swappable SAS/SATA HDDs or SSDs together with up to eight on-board NVMe PCIe ports that allow direct connections to the U.2 NVMe PCIe SSDs. The ThinkSystem SR650 server also supports up to two NVIDIA GRID cards for AI or media processing acceleration.

The SR650 server supports up to two processors, each with up to 28-core or 56 threads with hyper-threading enabled, up to 38.5 MB of last level cache (LLC), up to 2666 MHz memory speeds and up to 3 TB of memory capacity. The SR650 also support up to 6 x PCIe slots. Its on-board Ethernet solution provides 2/4 standard embedded Gigabit Ethernet ports and 2/4 optional embedded 10 Gigabit Ethernet ports without occupying PCIe slots. All these advanced features make the server ideal to run data and bandwidth intensive VNF workload and storage functions of NFVI platform.

For more information, see product guide.

4.1.1 Memory

There are 24 slots in total for memory in the SR650 server. The maximum memory for each slot is 128 GB. So, the maximum memory of one SR650 server can reach `128GB * 24 slots = 3TB`.

For each SR650 server, it is recommended to use 64GB memory for a small deployment, 128G memory for a medium deployment, and 256GB for a large deployment.[1]

4.1.2 RAID controller

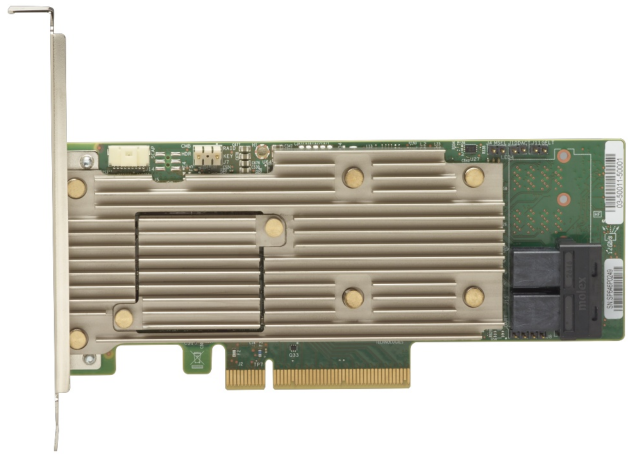

Figure 4.2: ThinkSystem RAID 930-8i RAID Controller

The ThinkSystem RAID 930 family of internal 12 Gbps SAS RAID controllers are high-performance RAID-on-chip (ROC) adapters. These adapters support RAID levels 0/1/10/5/50/6/60 as well as JBOD, and include an extensive list of RAS and management features.

The family is comprised of four adapters:

1. The ThinkSystem RAID 930-4i supports up to four internal SAS and SATA drives
2. The ThinkSystem RAID 930-8i supports up to eight internal SAS and SATA drives
3. The ThinkSystem RAID 930-16i supports up to 16 internal SAS and SATA drives
4. The ThinkSystem RAID 930-24i supports up to 24 internal SAS and SATA drives

For more information, see product guide.

4.1.3 Disk configurations

Disk configuration is important to achieve high performance. We recommend the following in each SR650 server:

Type	Position	Number	Size	RAID	Purpose
SSD	Front backplane	2	800GB	RAID1	Operating system
SSD	Front backplane	2	128GB	RAID1	LVM cache

[1]See Red Hat DEPLOYING RED HAT HYPERCONVERGED INFRASTRUCTURE for details.

Type	Position	Number	Size	RAID	Purpose
SAS HDD	Rear backplane	8	2TB	RAID6	Glusterfs

Total Size for Gluster: `3 * 8 * 2TB = 48TB`.

4.2 Network Switches

The following sections describe the Top-of-Rack (ToR) switches used in this reference architecture. The Networking Operating System software features of these Lenovo switches deliver seamless, standards-based integration into upstream switches. Two 10 Gb switches and two 1Gb switches are used in this architecture.

4.2.1 Lenovo RackSwitch G8272 (10Gb)

Figure 4.3: Lenovo RackSwitch G8272

The Lenovo RackSwitch G8272 uses 10Gb SFP+ and 40Gb QSFP+ Ethernet technology and is specifically designed for the data center. It is an enterprise class Layer 2 and Layer 3 full featured switch that delivers line-rate, high-bandwidth, low latency switching, filtering, and traffic queuing without delaying data. Large data center-grade buffers help keep traffic moving, while the hot-swap redundant power supplies and fans (along with numerous high-availability features) help provide high availability for business sensitive traffic.

The RackSwitch G8272 (as shown in Figure 4.3) is ideal for latency sensitive applications, such as high-performance computing clusters, financial applications and NFV deployments. In addition to 10 Gb Ethernet (GbE) and 40 GbE connections, the G8272 can use 1 GbE connections (need the Photoelectric conversion module).

For more information, see product guide.

Figure 4.4: Lenovo RackSwitch G8052

4.2.2 Lenovo RackSwitch G8052 (1Gb)

The Lenovo RackSwitch™ G8052 (as shown in the following figure) is a top-of-rack data center switch that delivers unmatched line-rate Layer 2/3 performance at an attractive price. It has 48x 10/100/1000BASE-T RJ-45 ports and four 10 Gigabit Ethernet SFP+ ports (it also supports 1 GbE SFP transceivers), and includes hot-swap redundant power supplies and fans as standard, which minimizes your configuration requirements. Unlike most rack equipment that cools from side-to-side, the G8052 has rear-to-front or front-to-rear airflow that matches server airflow.

For more information, see product guide.

5

Software Services

Lenovo Open Cloud software can be viewed in three categories based on the services they provide:

1. **Platform services**: Platform services are built upon Red Hat Hyperconverged Infrastructure (RHHI). It provides LOC core services each deployed in one or more virtual machines.
2. **Storage services**: Storage services are built upon Ceph. It provides capability to manage Ceph cluster up to xx.
3. **Cloud services**: Cloud is built upon Red Hat Openstack.

5.1 Platform services

Platform services provide administrative functions to support operation of the Open Cloud. This includes management of software life cycle, automation, list of artifacts such as ISO images and qcow images, and new server discovery.

5.1.1 Runtime service

Built upon Red Hat Hyperconverged Infrastructure (RHHI-V). It supports virtual machine users out of box, and is the foundation of other Lenovo Open Cloud services.

RHHI integrates Red Hat Virtualization (RHV) and Red Hat Gluster Storage (RHGS).

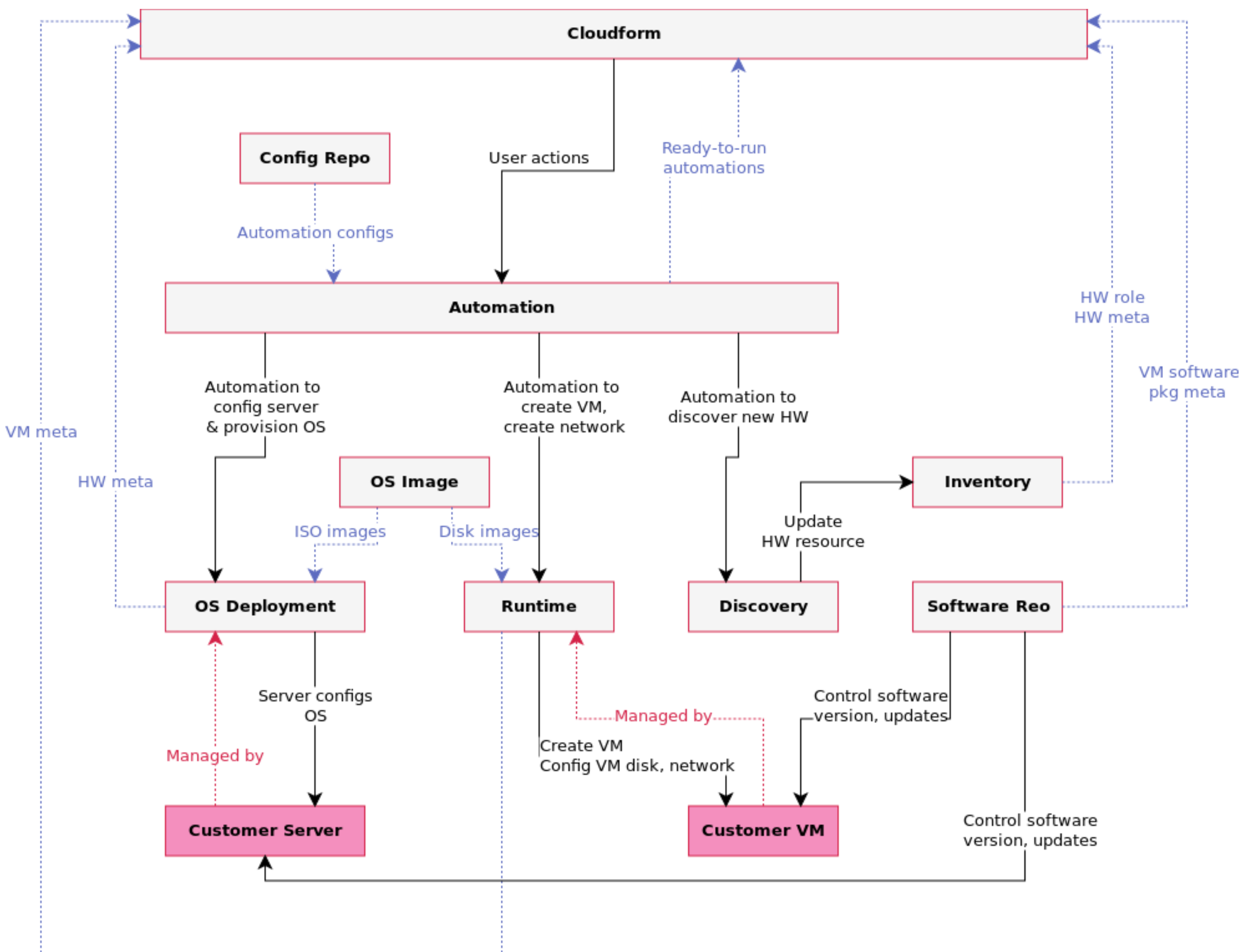

Figure 5.1: Platform services overview

> RHHI for Virtualization provides open-source, centrally administered, and cost-effective integrated compute and storage in a compact footprint for remote sites.

See product guide for details.

5.1.2 Software repository & life cycle management service

Built upon 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 side, 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 subscription, registration, and content from the central Satellite Server.

See Red Hat Product Guide for details.

All Open Cloud physical servers and Lenovo installed virtual machines are registered to this service. If you are using the Open Cloud Automation service to create new virtual machine, it will register itself to this repo service in order to receive Red Hat product updates.

By default, the repo services offers three environments:

1. **development**:
2. **QA**:
3. **production**:

By default, new virtual machine will register to `development` environment, thus having access to the latest packages and fixes.

5.1.3 Automation service

Built upon Red Hat Ansible Tower. It is the single point of contact to manage servers and VMs using ansible playbooks.

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

See product guide for details.

5.1.4 Discovery service

Built upon Lenovo Confluent. It continuously monitors network for new Lenovo server and switch. Once identified, the new hardware can be enlisted by other Open Cloud services, such as extending Ceph cluster or adding an Openstack compute node.

See product guide for details.

5.1.5 Inventory planning service

5.1.6 Server config & OS deployment service

5.1.7 OS image service

5.1.8 Configure & Automation repository service

5.1.9 Launchpad

5.2 Storage services

5.2.1 Ceph capacity management

5.3 Cloud services

5.3.1 Cloud management

5.3.2 Brain controllers

6

Platform Network Design

6.1 Design Conventions

There is endless possibility to design a network. In this architecture we recommend the following conventions as best practice:

1. Data center switches follow a `spine-leaf` topology.
2. Switch to switch connections are always paired for redundancy.
3. Except BMC connection, server to switch connections are always paired. In addition, each pair should connect to separate NICs on the server at north bound, and separate switch at south bound.

   This requires matching configuration on the switch using LACP, and on the server using **active-active** `bonding`. Both we will demonstrate how to create in the following sections.
4. Separate management traffic from data traffic whenever possible.
5. Improve network isolation by assigning private IP space to internal only traffic whenever possible.

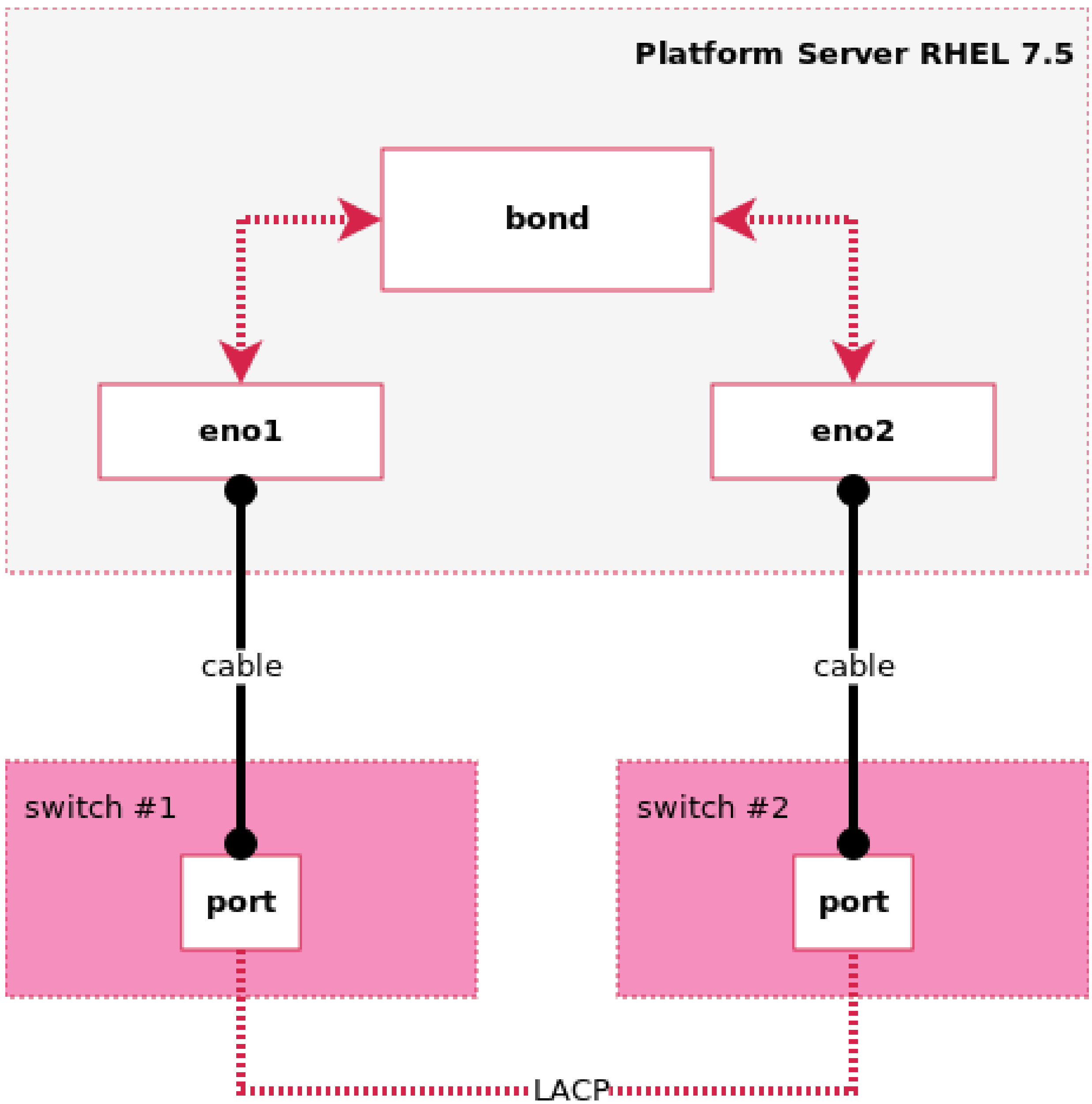

Figure 6.1: Server-Switch network convention

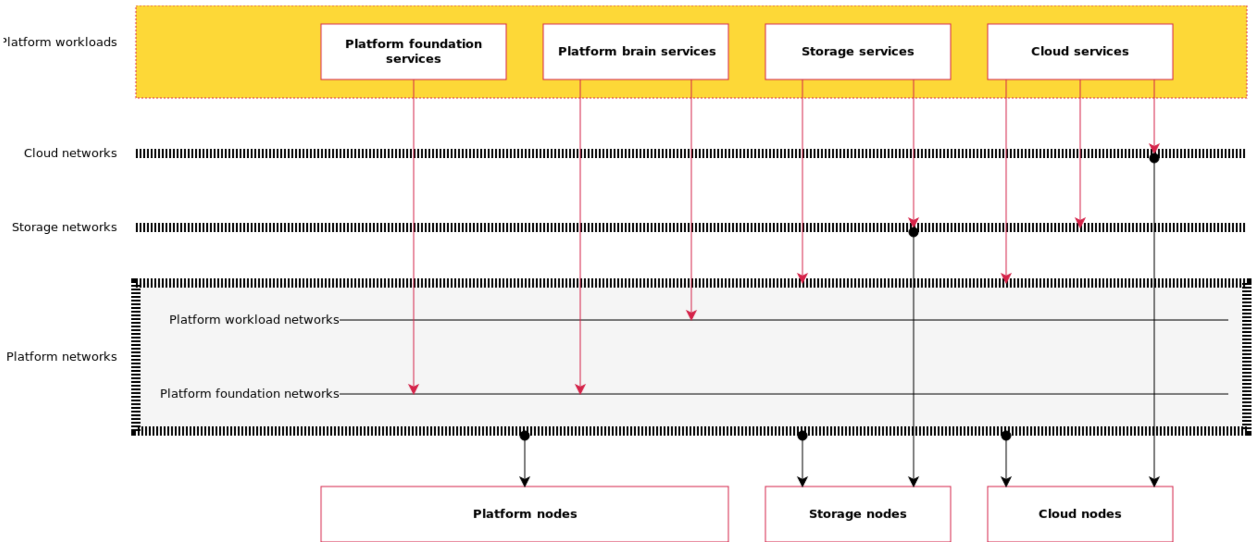

Figure 6.2: Lenovo Open Cloud Network Overview

6.2 Network Overview

On the high level, the Open Cloud networks can be viewed in three groups — platform network, storage network, and cloud network. In the following sections, we will discuss the network design to support **Platform** hardware and services. We will follow these steps to help user understand the design and highlight considerations we recommend to take in implementation:

1. Define logical networks by its function: this defines the purpose of this network, thus clarifying its characteristics such as load, latency, space, etc..
2. Assign VLAN to logical networks: in theory, each logical network is assigned a unique VLAN schema.
3. Map network/VLANs to server's network interface: this defines how server side interfaces will be configured to support these networks.
4. Map server's networking interface to switch: this defines the connection between server and switch, thus the switch side configurations including port mode, vLAG, native vlan, and untagged vlans.
5. Cabling schema: this defines switch port assignments, which server port is to be connected to which switch and which port.
6. Configure switches: this shows how to apply switch side configurations to Lenovo G8052 and Lenovo G8072 switches.
7. Configure server network interfaces: this shows configuration files used to create network interfaces on the server with RHEL 7.5
8. Map services to network/VLAN: this defines networks that are needed to each Platform service.
9. Configure service VM: this defines configuration steps used to prepare each Platform service

instance to be in compliance with this network design.

6.3 Connection To Upstream

Showing switch topology within IBB as well as how it is connected to upstream → what is required from upstream, eg. dhcp, dns, gateway, access to RH CDN.

6.4 Define Platform Logical Networks

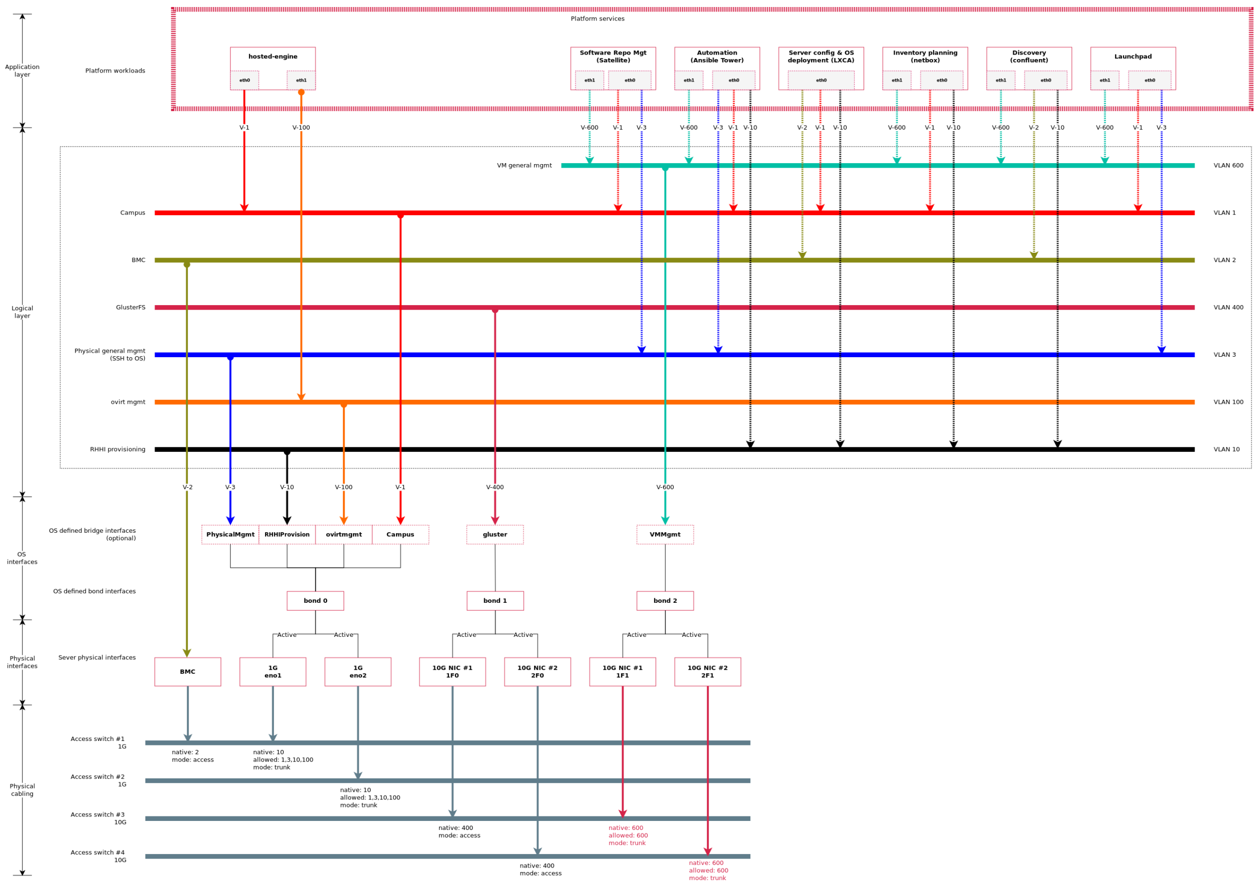

Figure 6.3: Lenovo Open Cloud Platform Network Overview

Platform networks are designed to offer performance and high availability. For illustration purpose we are to separate networks by their function so to highlight some design considerations. It is possible to merge these to fewer networks, or to reuse existing ones for the purpose. See Implementation Worksheet for details.

Campus `campus` network is a name for public access. This is the network that Open Cloud user uses to access its services, eg. Ansible Tower's web UI.

BMC also called `Out-of-Band` (OOB) management network. It connects to a dedicated management port on physical server that is separated from data ports.

ovirt management is a private network linking RHHI management console to RHHI clusters. Except Platform admin, other users should not have direct access to it.

glusterFS is a private network used by `gluster` clusters. Gluster cluster is the storage backend of Open Cloud Platform. For example, in a 6-server configuration, three platform servers will form a 3-node gluster cluster.

Physical server management is to support traffic of `In-Band` managerial tasks, eg. `ssh` to a server.

RHHI provisioning is to support data traffic of installing OS on a physical server. Separating this to its own network is a best practice because operating system image can be large, thus its loading to server will affect other tenants.

VM management is to access RHHI virtual machines. This supports both the Open Cloud services and VM workloads. Later we will see that it's also advised to dedicate a NIC for this same purpose.

As a convention, we assign networks that only support *Open Cloud private traffics* with private spaces, eg. `192.168.x.x`, defined in RFC 1918 whenever possible. This makes this environment self-contained, and you can gradually open it up by routing or other network methods to expose service and access.

Table 6.1: Platform Logical Network Defintions

Logical Network	VLAN	Subnet	Addresses	Mask	Static / DHCP	Gateway
Campus	1	10.240.x.x[1]	10		static	10.240.x.1
BMC	2	192.168.2.x	254	/24	static	192.168.2.1
Physical server management	3	192.168.3.x	254	/24	static	192.168.3.1
RHHI provisioning	10	192.168.10.x	3/6/9	/24	static	192.168.10.1
OVIRT management	100	192.168.100.x	3/6/9	/29	static	192.168.100.1
glusterFS	400	192.168.40.x	3/6/9	/29	static	192.168.40.1
VM management	600	192.168.60.x	11	/28	static	192.168.60.1

6.5 Map VLAN to Server NIC

First, we are to map VLAN to server network interfaces. Besides BMC port, each server has minimal two 1Gb ports and four 10Gb ports. Interfaces are paired to form an `active-active` bonding interface on Platform server. A RHHI network profile is created for each logical network, which in

[1]This is an example subnet.

turn creates a bridge on RHHI server with the same name.

Table 6.2: Platform VLAN to Server's NIC Mapping

Logical Network	VLAN	RHHI Network	Bond	BMC	2 x 1G	2 x 10G	2 x 10G
Campus	1	Campus	0		x		
BMC	2	n/a	n/a	x			
Physical server management	3	PhysicalMgmt	0		x		
RHHI provisioning	10	RHHIProvision	0		x		
OVIRT management	100	ovirtmgmt	0		x		
glusterFS	400	gluster	1			x	
VM management	600	VMMgmt	2				x

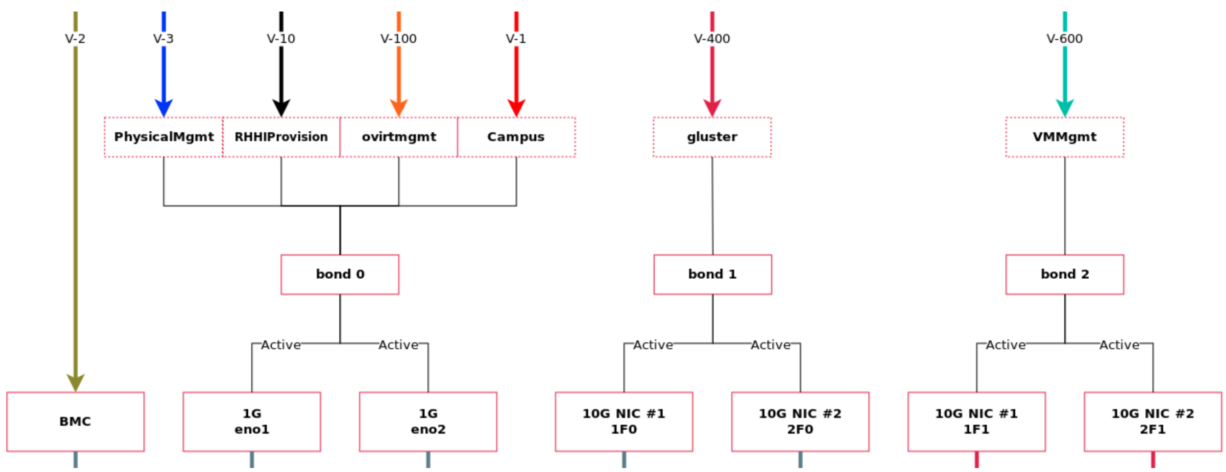

Figure 6.4: Lenovo Open Cloud Plaform Server Network Interfaces

6.6 Map Server NIC to Switch

Next, we are to map connection between server's network interface to switch connections.

By convention, in order to achieve high availability of connections, Open Cloud uses two G8052 (1Gb) switches and two G8272 (10Gb) switches to gain redundancy. Table below shows switch port configurations including `mode`, `native VLAN` (aka. untagged VLAN), and `tagged VLAN` (aka. allowed VLANs):

On the server side, each server has minimal two 1Gb ports (for example, `eno1` and `eno2`) and four 10Gb ports (for example, `1F0`, `1F1`, `2F0`, `2F1`). Once provisioned three bond interfaces and three bridge intefaces will be created per server:

Table 6.3: Platform Server NIC to Switch Mapping

Server NIC	Server Bond	Switch Mode	Native VLAN	Tagged VLAN
BMC	n/a	access	2	n/a
eno1, eno2	bond 0	trunk	10	1,3,10,100
1F0, 2F0	bond 1	access	400	n/a
1F1, 2F1	bond 2	trunk	600	600

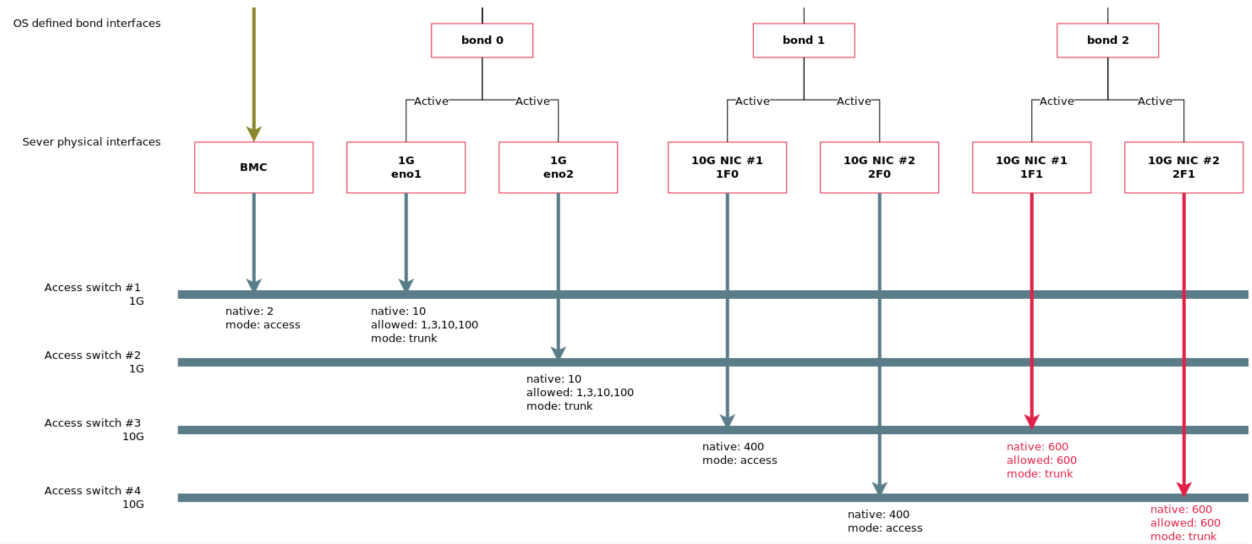

Figure 6.5: Lenovo Open Cloud Plaform Server to Switch

6.7 Define Cable Schema

Each environment is different. Here we present an example cable schema following the network designs laid out in previous sections. In the following sections we will use this schema[2] to demonstrate switch port configurations.

We have opted to reserve port 1-16 on two G8052 switches for BMC connections of all servers within this rack. From experience we found this setup easier for troubleshooting BMC connections on the switch. However, your management style may call a different approach.

Table 6.4: Platform server to switch schema in a 3-server configuration

Port	G8052 (1Gb)	G8052 (1Gb)	G8272 (10Gb)	G8272 (10Gb)
1	server 1 BMC		server 1 1F0	server 1 2F0

[2] In this layout, we follow a practice of reserving port 1-17 for BMC connections on 1Gb switches.

Port	G8052 (1Gb)	G8052 (1Gb)	G8272 (10Gb)	G8272 (10Gb)
2	server 2 BMC		server 2 1F0	server 2 2F0
3	server 3 BMC		server 3 1F0	server 3 2F0
4			server 1 1F1	server 1 2F1
5			server 2 1F1	server 2 2F1
6			server 3 1F1	server 3 2F1
17	server 1 eno1	server 1 eno2		
18	server 2 eno1	server 2 eno2		
19	server 3 eno1	server 3 eno2		

6.8 Configure Switch Ports

There are multiple methods to apply switch port configurations, see Switch Port Configuration Methods for details. Here we take the two connections to port `17` of G8052 switches for example and walk through switch side configurations using switch CLI directly.

In general, the workflow will be:

1. Login in switch as `admin` user.
2. Enter `config` mode.
3. Select the port to configure.
4. Apply settings.
5. `exit` config mode. This then allows you to select another port without leaving the `config` mode.

6.8.1 Login to switch

Lenovo G8052 switch allows telnet access by default. Use the switch IP, for example, `10.240.43.51`, default admin user `admin` and default admin password `admin`:

```
$ telnet 10.240.43.51
Trying 10.240.43.51...
Connected to 10.240.43.51.
Escape character is '^]'.

Lenovo RackSwitch G8052.


Enter login username: admin <-- admin user name
```

```
Enter login password:       <-- admin user password

G8052>
```

Note that prompt `G8052>` shown above is depending on the switch name, in this example, `G8052`. This value is configurable. Thus your terminal prompt may look different.

6.8.2 Enter config mode

The rest of port configurations can only be applied when in `config mode`.

```
G8052> en <-- enable admin mode
G8052> configure <-- to enter config mode
```

6.8.3 Configure BMC connections

Table 6.5: Platform server BMC connection switch port config

Server NIC	Server Bond	Switch Mode	Native VLAN	Tagged VLAN
BMC	n/a	access	2	n/a

To configure BMC connections, using G8052 port 1 for example:

1. select the port to config: replace `1/1` with `1/2` for port 2, and `1/3` for port 3.
2. set port mode: `access`
3. set native vlan: `2`

Example:

```
G8052> interface ethernet 1/1 <-- select port
G8052> bridge-port mode access <-- set mode
G8052> bridge-port access vlan 2 <-- set native vlan
```

6.8.4 Configure `bond 0` connections

In general, `bond 0` is the connection for management traffic. These type of traffics are typically low frequency and requiring low bandwidth. To support Platform services, this interface will include `Campus` network (VLAN 1), `Physical server management` network (VLAN 3), `RHHI provisioning` network (VLAN 10), and `ovirt management` network (VLAN 100).

Table 6.6: Platform server `management` connection switch port config

Server NIC	Server Bond	Switch Mode	Native VLAN	Tagged VLAN
2 x 1G	bond 0	trunk	10	1,3,10,100

To configure connections used for `bond 0`, using G8052 port 17 for example. Apply the same configuration to port 17, 18 and 19 on both G8052 switches according to the cable schema.

Example:

```
G8052> interface ethernet 1/17
G8052> bridge-port mode trunk
G8052> bridge-port trunk allowed vlan 1,3,10,100 <-- set tagged vlans
G8052> bridge-port trunk native vlan 10 <-- set native vlan
```

Note that you must set `native vlan` after `allowed vlan`, because the native vlan must also be included as an *allowed* vlan. Otherwise, CLI will fail with error.

6.8.5 Configure `bond 1` connections

Platform server's `bond 1` interface is designed to handle data traffic to the Gluster file system of RHHI.

Table 6.7: Platform server `bond 1` connection switch port config

Server NIC	Server Bond	Switch Mode	Native VLAN	Tagged VLAN
2 x 10G	bond 1	access	400	n/a

To configure connections used for `bond 1`, using G8272 port 1 for example. Apply the same configuration to port 1,2,3 on both G8272 switches according to the cable schema.

Example:

```
G8272> interface ethernet 1/1
G8272> bridge-port mode access
G8272> bridge-port access vlan 400
```

6.8.6 Configure `bond 2` connections

Platform server's `bond 2` is designed to support traffic generated by additional storage deployment such as Ceph, and cloud deployment such as OpenStack.

Table 6.8: Platform server `workloads` connection switch port config

Server NIC	Server Bond	Switch Mode	Native VLAN	Tagged VLAN
2 x 10G	bond 2	trunk	600	600

To configure connections used for `bond 2`, using G8272 port 4 for example. Apply the same configuration to port 4,5,6 on both G8272 switches according to the cable schema.

Example:

```
G8272> interface ethernet 1/4
G8272> bridge-port mode trunk
G8272> bridge-port trunk allowed vlan 600
G8272> bridge-port trunk native vlan 600
```

Note that this may appear strange that we are using `trunk` mode even though there is only one vlan — VLAN 600. The mode is necessary when we add Open Cloud's storage services and cloud services to the stack.

6.9 Configure Server Interfaces

Platform servers have at least two 1Gb ports and four 10Gb ports. As defined in Platform Networks, Platform servers will be configured with three bonding interfaces and a list of bridges:

Server Physical Interface	Bond Interface	Bridges
eno1, eno2	bond 0	Campus,ovirtmgmt,PhysicalMgmt,RHHIProvisioning
1F0, 2F0	bond 1	gluster
1F1, 2F1	bond 2	VMMgmt

Note that the name of these interfaces, eg. `eno1` for the *first* 1Gb port, and `1F0` for the *first* 10Gb port, are assigned by operating system. These names, however, are depending on various factors such as the slot the NIC card, and naming convention the operating system follows. Therefore, they are presented here for illustration purpose. Please contact Lenovo Sales for assistance of your hardware configuration.

Further, the ports you choose to cable to switch can also be different from the example cable schema. Use the Implementation Worksheet to create a mapping between your environment and this example.

6.9.1 Configure Bonding Interface

We will use `bond 0` for example to show steps to create a bonding interface on a Platform server running RHEL 7.5. We will highlight options and values that are important for Lenovo Open Cloud. For general information of bonding interface, please refer to Red Hat Enterprise Linux 7 Networking Guide.

On each Platform servers:

1. Go to `/etc/sysconfig/network-scripts/`.
2. Create networking config file `ifcfg-bond0`. Replace `IPADDR`, `NETMASK`, `GATEWAY`, and `DNS1` values with your values:

   ```shell
   DEVICE=bond0
   BONDING_OPTS='mode=4'
   ONBOOT=yes
   BOOTPROTO=none
   DEFROUTE=no
   IPADDR=10.240.41.231
   NETMASK=255.255.252.0
   GATEWAY=10.240.40.1
   DNS1=10.240.0.10
   ```

   Note:

   1. `DEVICE`: name of the bonding interface, in this example, `bond0`.
   2. `BONDING_OPTS`:
      1. `mode=4`: set to **active-active**. This is consistent with switch ports who are bonded in vLAG. Refer to Red Hat Enterprise Linux 7 USING CHANNEL BONDING for more information of bonding modes and their implications.
      2. `miimon=100`: enable MII monitoring.
   3. `DEFROUTE`: must be `no`.
3. Create `ifcfg-eno1`:

   ```shell
   DEVICE=eno1
   MASTER=bond0
   SLAVE=yes
   ONBOOT=yes
   DEFROUTE=no
   ```

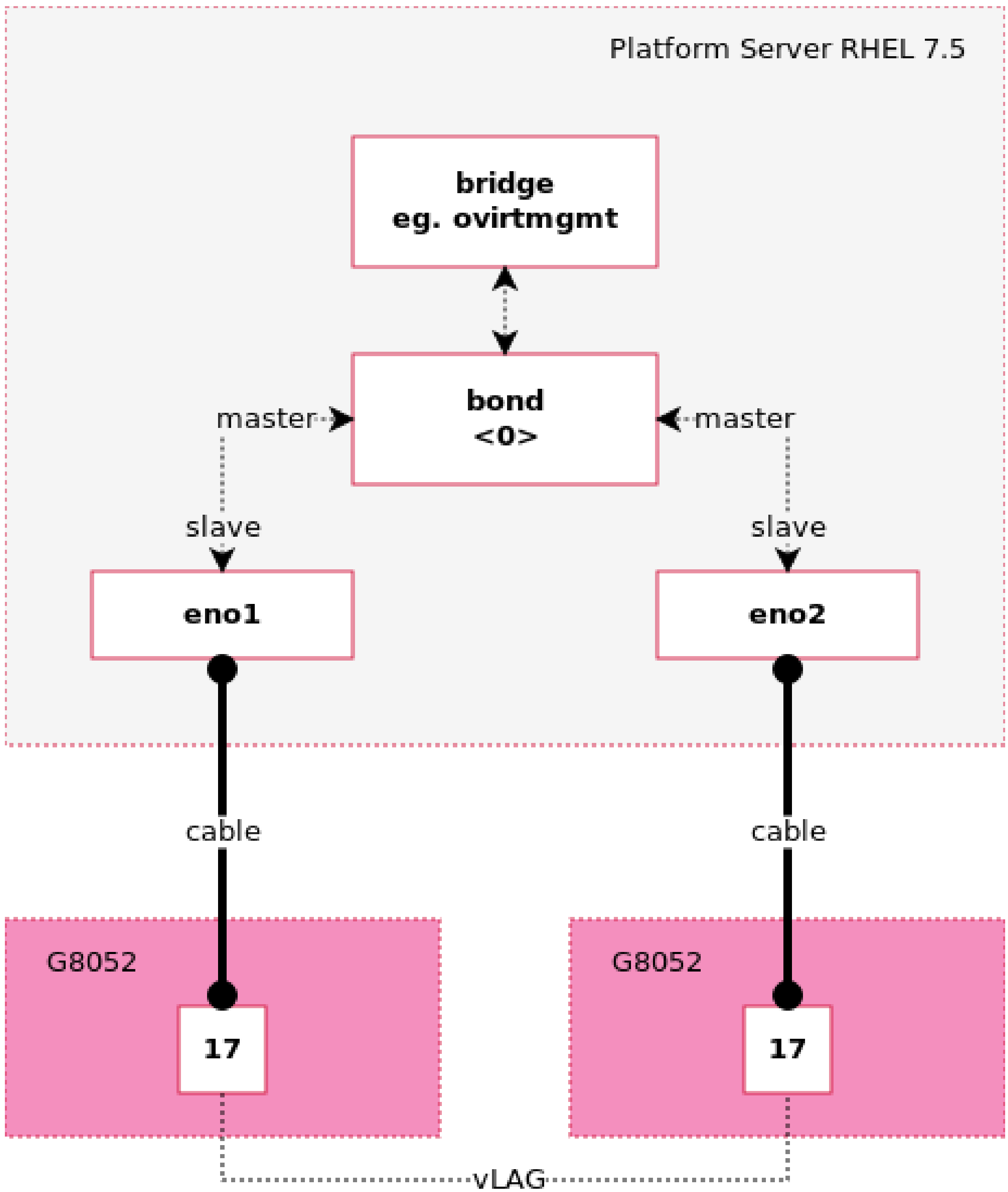

Figure 6.6: Example of Platform server bonding interface and their switch connections

`` ``` ``

Note:

1. Here we setup a master-slave between the physical NIC interface `eno1` (as slave) and bonding interface `bond0` (as master).

4. Similar to `eno1` bonding, create `ifcfg-eno2`:

   ```shell
   DEVICE=eno2
   MASTER=bond0
   SLAVE=yes
   ONBOOT=yes
   DEFROUTE=no
   ```

5. Restart network service to take effect: `systemctl network restart`.

6. Use `ip a` to verify that three interfaces are up and running &mdash: `eno1`, `eno2`, and `bond0`.

6.9.2 Create Bridge Interface

Virtual bridge is a flexible way to create multiple interfaces that can be tied to a physical or another virtual interface. On Open Cloud Platform, we build virtual bridges on top of bonding interface to serve two purposes:

1. Further isolate networks by their function.
2. Ease of maintenance since user can add and remove user-defined networks as virtual bridges freely through Open Cloud Automation service, or through the RHV Administrator Portal.

Table 6.10: Platform server bridge interfaces

Logical Network	Bridge Name	Bond Interface	Supporting VLAN
Campus	Campus	0	1
Physical server management	PhysicalMgmt	0	3
RHHI provisioning	RHHIProvisioning	0	10
OVIRT management	ovirtmgmt	0	100
glusterFS	gluster	1	400
VM management	VMMgmt	2	600

In this example, we will setup bridge interfaces using the RHV Administrator Portal:

1. Login Admin portal and select `System > Network`.

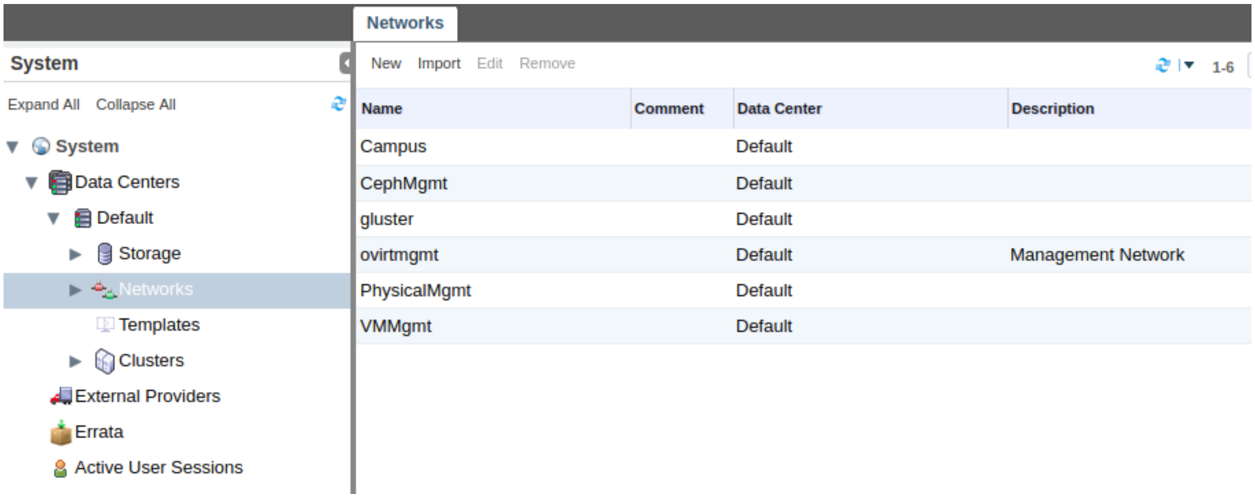

Figure 6.7: View RHV Network List

2. Click `New` to create a new network, or `Edit` to modify an existing one. This will in turn create a virtual bridge on the host. In this example, `PhysicalMgmt`:

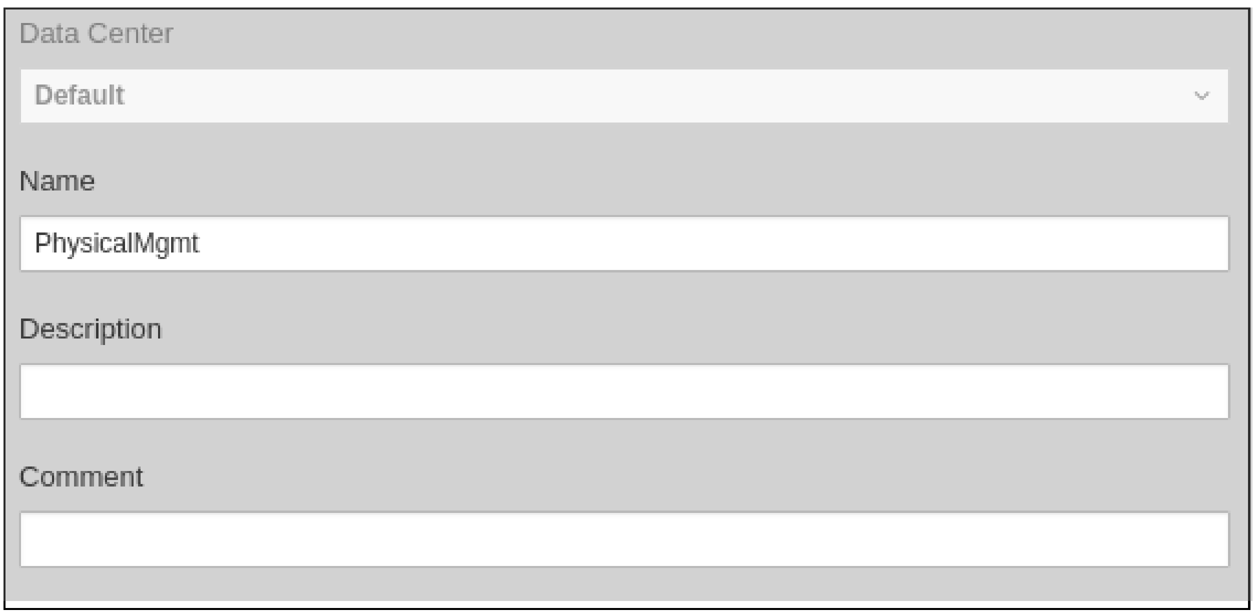

Figure 6.8: Create New RHV Network

3. Set VLAN. Only one VLAN can be associated with one bridge interface.
4. Link the bridge to a bonding interface.

Link RHV Network to a bonding interface

To view the resulting bridge interface configurations created by the Admin Portal:

1. Login into any Platform server.

Figure 6.9: Setup RHV Network VLAN

2. Go to `/etc/sysconfig/network-scripts/`.

3. Check `ifcfg-PhysicalMgmt`:

   ```shell
   DEVICE=PhysicalMgmt
   TYPE=Bridge
   DELAY=0
   STP=off
   ONBOOT=yes
   MTU=1500
   DEFROUTE=no
   NM_CONTROLLED=no
   IPV6INIT=no
   ```

   Note:

   1. `DEVICE`: bridge interface name, in this case, `PhysicalMgmt`.
   2. `TYPE`: must be `Bridge`.

4. Check `ifcfg-bond0.3` that links bridge `PhysicalMgmt` to `bond0`, and supports VLAN `3`:

   ```shell
   DEVICE=bond0.3
   VLAN=yes
   BRIDGE=PhysicalMgmt
   ONBOOT=yes
   MTU=1500
   DEFROUTE=no
   NM_CONTROLLED=no
   IPV6INIT=no
   ```

Note:

1. `DEVICE`: to define a VLAN on an interface, both the filename and `DEVICE` should be set in format of `<interface name>.<vlan number>`.
2. `BRIDGE`: name of the bridge, in this case `PhysicalMgmt`, that this bonding interface is connected to.

6.10 Map Services to VLAN

Platform services are VMs running on RHV. Based on the VLANs we have defined in previous section, we are to map these services to VLANs based on the functions the service will provide.

Table 6.11: Platform Services to VLAN Mapping

Platform Services	1	2	3	10	100	400	600
Runtime	x				x		
Software repository & life cycle management	x		x				x
Automation	x		x				x
Server config & OS deployment	x	x		x			
Inventory planning	x	x					x
Discovery		x		x			x
OS image							
Configure & Automation repository service							
Launchpad	x		x				x

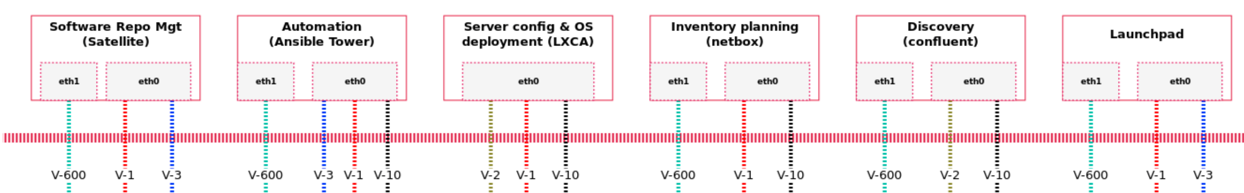

Figure 6.10: Lenovo Open Cloud Plaform Service Network Interfaces

6.11 Configure VM Interfaces

7

Storage networks

Lenovo Open Cloud supports Ceph storage backend. A storage backend can be shared among multiple workloads and platforms, such as an OpenStack on-premise cloud. By deploying Ceph on top of Open Cloud Platform, we can also share common services and networks.

7.1 Ceph Specific Networks

Ceph itself adds three new networks — Ceph management, Ceph storage public, and Ceph cluster private.

Ceph management is communication between Ceph dashboard and Ceph nodes, e.g. RPC, transferring zabbix monitoring data.

Ceph storage public data traffic by workloads who will use Ceph as storage backend, also known as `front-side`.

Ceph cluster private Ceph private data traffic such as used in data rebalancing, also known as `back-side`.

Table 7.1: Storage Logical Network Definition

Logical Network	VLAN	Subnet	Addresses	Mask	Static / DHCP	Gateway
Ceph management	30	192.168.30.x	254	/24	static	192.168.30.1
Ceph storage public	30x	192.168.3.x	254	/24	static	192.168.3.1

Logical Network	VLAN	Subnet	Addresses	Mask	Static / DHCP	Gateway
Ceph cluster private	40x	192.168.4.x	254	/24	static	192.168.4.1

7.2 Shared Platform networks

To manage Storage physical servers, we will connect them to three existing Platform networks — `BMC` (VLAN 2) for OOB management and node discovery, `RHHI Provisioning` (VLAN 10) for server OS provisioning, and `Physical server management` (VLAN 3) for general admin tasks such as SSH access.

In addition, to support `Storage capacity service`, which is the main management tool of Storage deployed on Platform, we also need `Campus` (VLAN 1) for accessing its UI, and `VM general management` (VLAN 600).

Table 7.2: Platform logical networks shared with Storage

Logical Network	VLAN	Subnet	Addresses	Mask	Static / DHCP	Gateway
Campus	1	10.240.x.x[1]	10		static	10.240.x.1
BMC	2	192.168.2.x	254	/24	static	192.168.2.1
Physical server management	3	192.168.3.x	254	/24	static	192.168.3.1
RHHI provisioning	10	192.168.10.x	3/6/9	/24	static	192.168.10.1
VM management	600	192.168.60.x	11	/28	static	192.168.60.1

7.3 Storage VLAN to server's NIC mapping

Network	VLAN	BMC	2 x 1G	2 x 10G	2 x 10G	Bond	Bridge
BMC	2	x				n/a	n/a
Ceph management	30		x			0	management
Ceph storage public	30x			x		1	public data path
Ceph cluster private	40x				x	2	private data path

7.4 Storage server's NIC to switch mapping

[1] This is an example subnet.

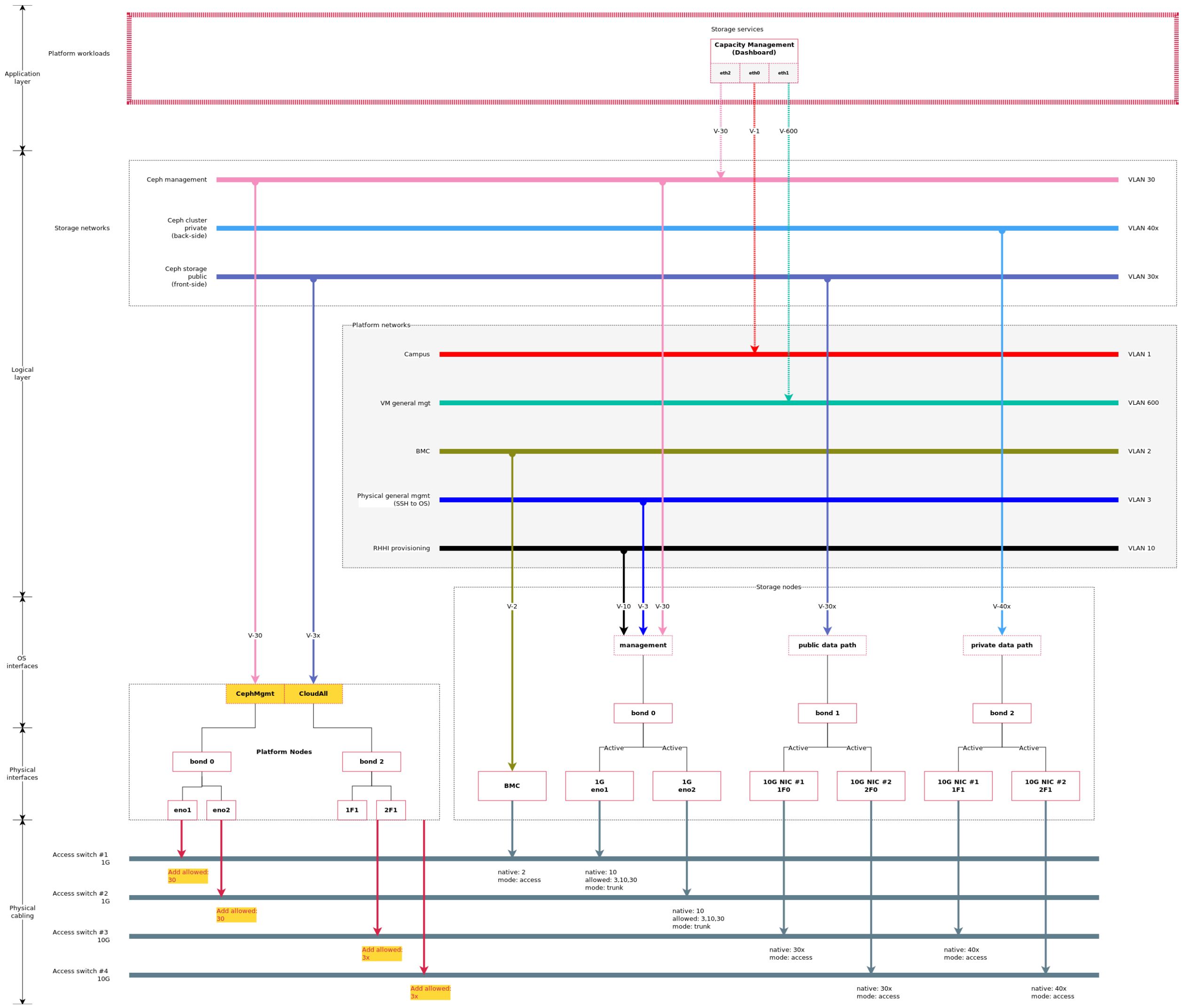

Figure 7.1: Lenovo Open Cloud Storage Networks

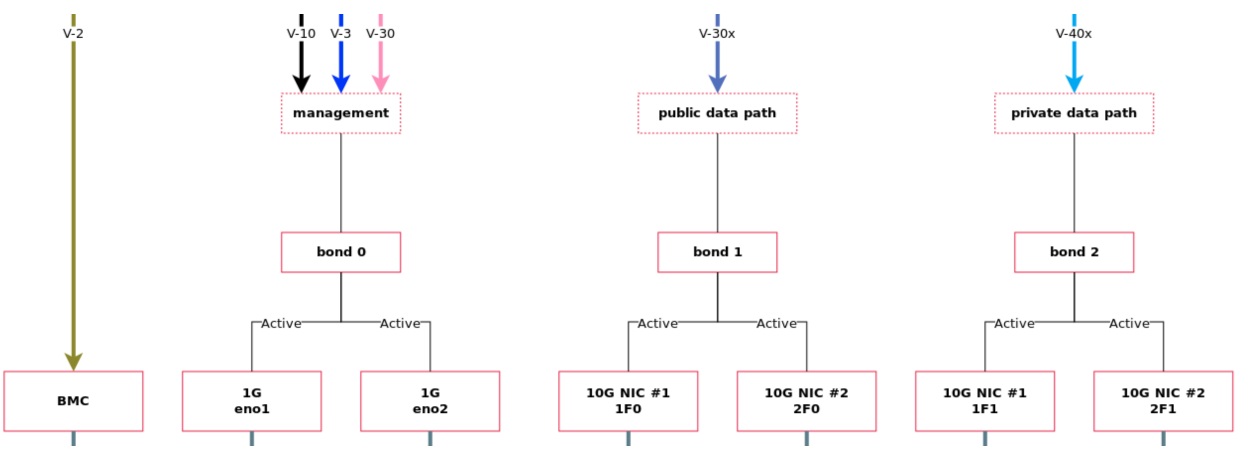

Figure 7.2: Lenovo Open Cloud Storage Server Network Interfaces

Server Side	Switch port mode	Switch port native VLAN	Switch port tagged VLAN
BMC	access	2	n/a
2 x 1G	trunk	10	3,10,30
2 x 10G	access	30x	n/a
2 x 10G	access	40x	n/a

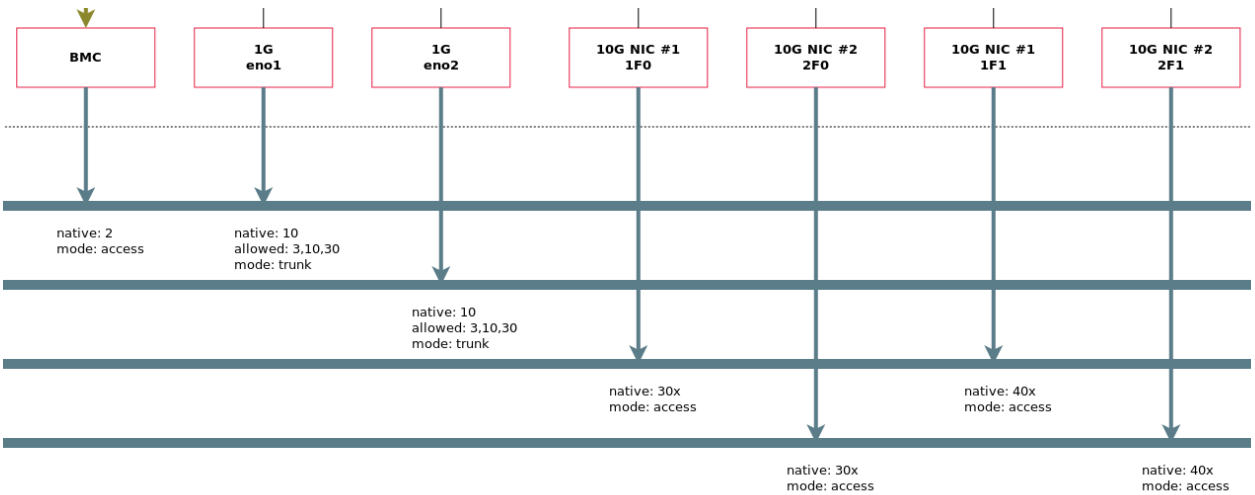

Figure 7.3: Lenovo Open Cloud Storage Server to Switch

7.5 Storage services to VLAN mapping

Storage Services	1	30	600
Capacity management	x	x	x

8

Cloud networks

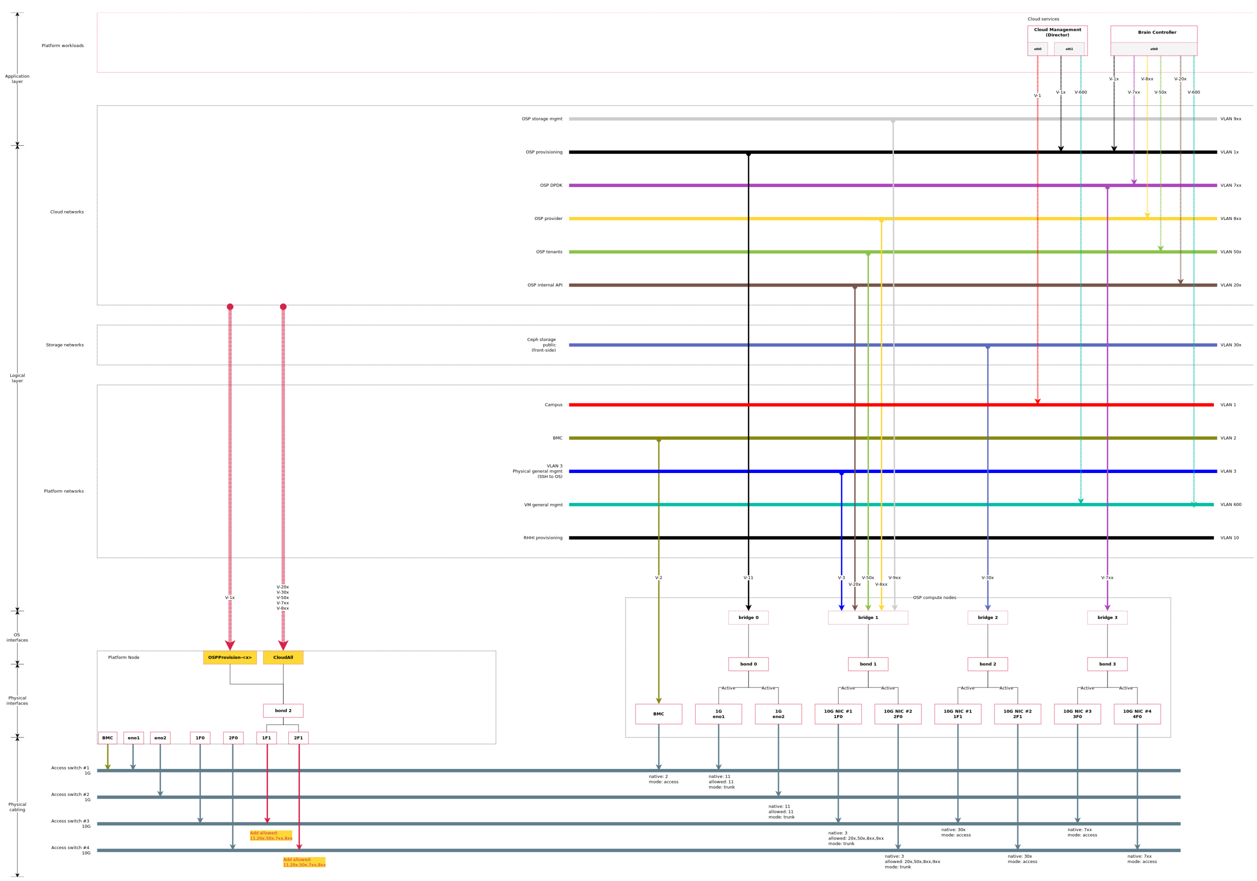

Figure 8.1: Lenovo Open Cloud Cloud Networks

9

Appendix

9.1 Implementation Worksheet (questioinnaire)

Implementation worksheets are tools to map Open Cloud infrastructure to your environment.

9.1.1 Switches

Table 9.1: Implementation Worksheet - Switch

Switches	Type	Qty		Model	Firmware
management switch	1Gb	2	suggested: your env:	G8052	8.3.3 ENOS
____________-	____	___	________	____-	_________
access switch	10Gb	2	suggested your env:	G8272	10.3.3.0 CNOS

9.1.2 Networks

In order for Lenovo Open Cloud functions to work as designed, it is important to map your network environment to meet requirements presented by suggested values.

For example, `Campus` network represents a public network used by users to access Open Cloud web portals, and by admins to manage server and/or IBB services remotely. Its default VLAN tag is `1`. If your environment uses a different VLAN schema, fill in your tag number in the cell below the default value one.

Table 9.2: Implementation Worksheet — Networks

Network		VLAN	Subnet	Mask	Gateway	Static/DHCP
Campus	suggested:	1	n/a	n/a	n/a	Static
	your env:					
IMM	suggested:	2	192.168.2.x	/24	192.168.2.1	Static
	your env:					
OVIRT mgt	suggested:	100	192.168.100.x	/29	192.168.100.1	Static
	your env:					
OS deployment	suggested:	10	192.168.10.x	/27	192.168.10.1	Static
	your env:					
Internal storage	suggested:	400	192.168.40.x	/29	192.168.40.x	Static
	your env:					
Server mgt	suggested:	3	192.168.3.x	/27	192.168.3.1	Static
	your env:					
VM mgt	suggested:	600	192.168.60.x	/28	192.168.60.1	Static
	your env:					

1. map server interface name –> ifcfg- files

9.2 Hardware BOM

Simplified version covers server & switch at high level should be fine.

9.3 Software BOM

9.3.1 6-server, HCI deployment, 3 year premium

Table 9.3: Software BOM, 6-server, HCI deployment, 3 year premium

SKU	Product	Qty
RS00139F3	Red Hat Hyperconverged Infrastructure for Virtualization (RHHI-V)	1
MCT3305F3	Red Hat Ansible Tower	1
MCT2981F3	Red Hat Openstack Platform (w/o Guest) with Smart Management	4
MCT2979F3	Red Hat OpenStack Platform with Smart Management & Guests	3
RS00036F3	Red Hat Ceph Storage	1
RS00031F3	Smart Management	3
MCT2838F3	Cloudforms	1
MCT3474F3	Red Hat Insights	1

9.4 Cable Schema

Port	G8052 (1Gb)	G8052 (1Gb)	G8272 (10Gb)	G8272 (10Gb)
1	server 1 BMC		server 1 1F0	server 1 2F0
2	server 2 BMC		server 2 1F0	server 2 2F0
3	server 3 BMC		server 3 1F0	server 3 2F0
4			server 1 1F1	server 1 2F1
5			server 2 1F1	server 2 2F1
6			server 3 1F1	server 3 2F1
17	server 1 eno1	server 1 eno2		
18	server 2 eno1	server 2 eno2		
19	server 3 eno1	server 3 eno2		

9.5 Configure Lenovo Switch Port

There are two ways to configure port on a Lenovo switch — using switch CLI, or using a Lenovo utility.

9.5.1 method 1: using switch CLI

Using switch CLI has always the choice of network admin. You can find more information of these command in the G8272 application guide.

1. Open a telnet session to the switch. Default login is username `admin` and password `admin`.

   ```shell
   ```

````
$ telnet  10.240.41.51 <-- switch IP
```
````

2. Once in the switch admin terminal

    1. enter config mode: `en` then `configure`
    2. select the port to config: `interface ethernet 1/<port id>`
    3. set mode to `trunk`: `bridge-port mode trunk`
    4. set allowed vlans: `bridge-port trunk allowed vlan <1,2,3..>`. Allowed vlans can be a comma delimited value list.
    5. last, set native vlan: `bridge-port trunk native vlan <vlan id>`. **Note** that native VLAN must also be included in the `allowed vlan` list.

    Example:

````
```shell
# en <-- enable admin mode
# configure <-- to enter config mode
# interface ethernet 1/<port id>  <-- select port to config
# bridge-port mode trunk
# bridge-port trunk allowed vlan <1,2,3...>
# bridge-port trunk native vlan <vlan id>
```
````
````