Lenov Open Cloud 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.

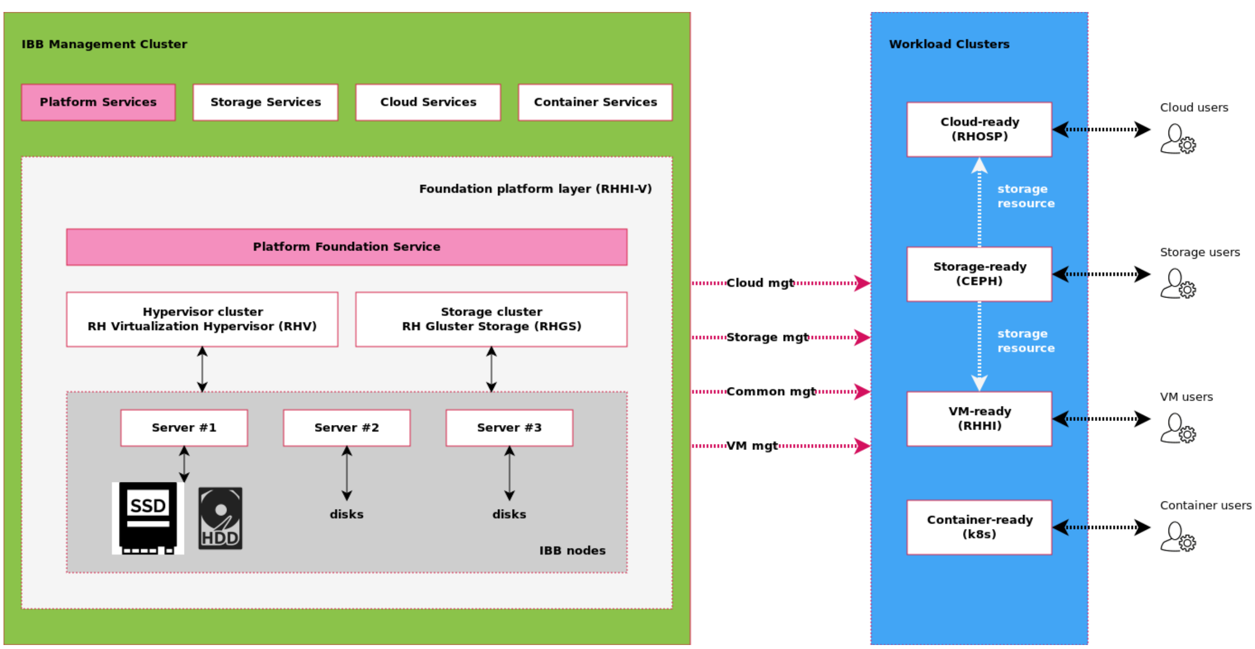

Figure 3.1: Lenovo Open Cloud Architecture

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

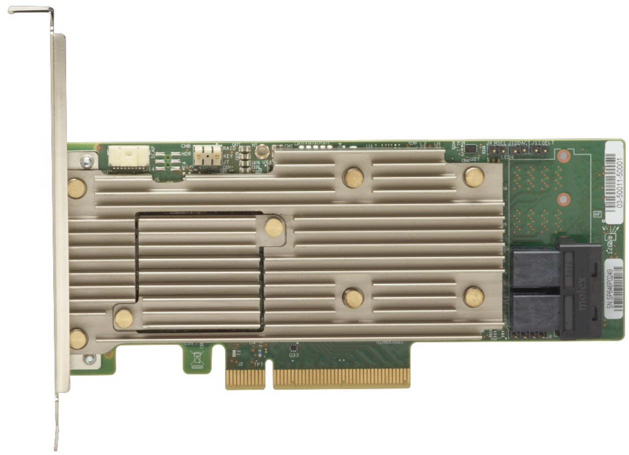

Figure 4.2: ThinkSystem RAID 930-8i RAID Controller

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 13) 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.

For more information, see product guide.

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

Figure 4.4: Lenovo RackSwitch G8052

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).

> 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. All Open Cloud servers are registered to this service, who then is responsible to manage life cycle of:

1. RHEL and Red Hat software products that are deployed in the Open Cloud.
2. Release, update, patch of Lenovo software products.
3. `.iso` and `qcow2` images, which are used by VM creation and server provisioning.

> 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 product guide for details.

5.1.3 Automation service

Build 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

Build 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

5.2 Storage services

5.2.1 Ceph capacity management

5.3 Cloud services

5.3.1 Openstack

6

Network Design

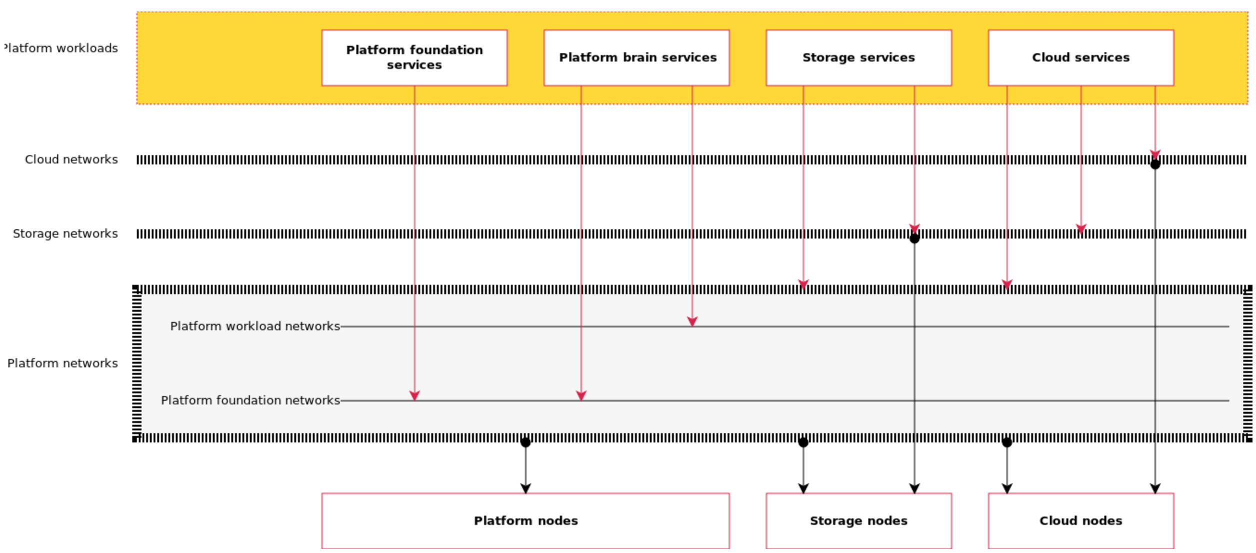

Figure 6.1: Lenovo Open Cloud Network Overview

LOC networks can be viewed in three groups whereas:

1. **platform network**: to support platform services.
2. **storage network**: built on top of platform network with added networks to handle Ceph data storage traffic and Ceph management functions.
3. **cloud network**:

6.1 Conventions

Hardware can break. It is important to keep this in mind when designing a network connection. In this architecture we have followed these conventions:

1. Inter switch connections are paired.
2. Except BMC connection, server to switch connections are paired.
    1. Each pair connect to separate NICs on the server at north bound, and separate switch at south bound.

This then requires matching configuration on the switch using LACP, and on the server using **active-active** `bonding`.

6.2 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.

6.3 Platform networks

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. For this use section "VLAN Mapping Worksheet".

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.

OS 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 can have negative impact on shared traffics.

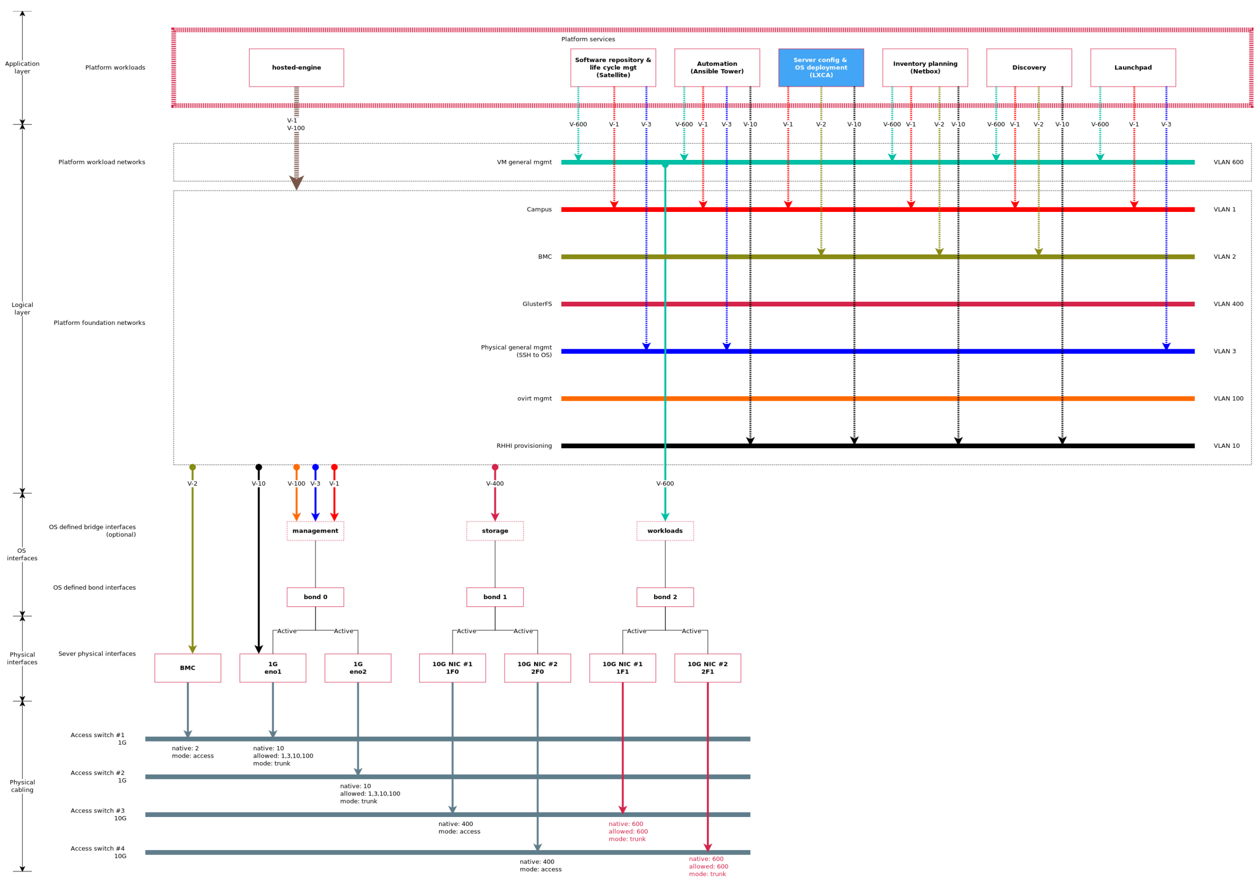

Figure 6.2: Lenovo Open Cloud Platform Network Overview

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.

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
OS 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

6.3.1 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			x
Server config & OS deployment	x	x		x			
Inventory planning	x	x		x			x
Discovery	x	x		x			x
OS image							
Configure & Automation repository service							

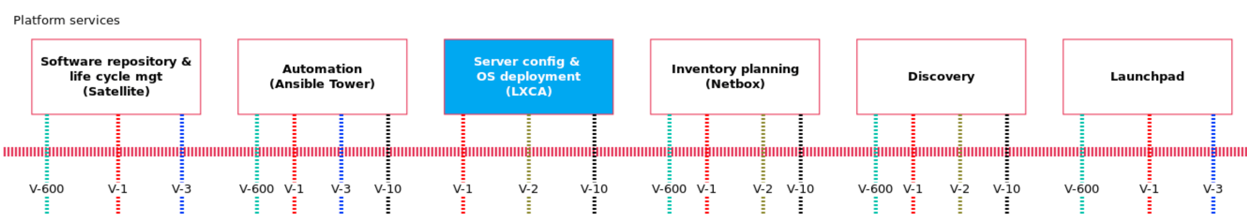

Figure 6.3: Lenovo Open Cloud Plaform Service Network Interfaces

[1]This is an example subnet.

6.3.2 Platform VLAN to server's NIC mapping

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. Optionally, we can also create a network bridge on top of a bonding interface. A sample configuration is shown below:

Network	VLAN	BMC	2 x 1G	2 x 10G	2 x 10G	Bond	Bridge
Campus	1		x			0	management
BMC	2	x				n/a	n/a
Physical server management	3		x			0	management
OS provisioning	10		x[2]			n/a	
OVIRT management	100		x			0	management
glusterFS	400			x		1	storage
VM management	600				x	2	workloads

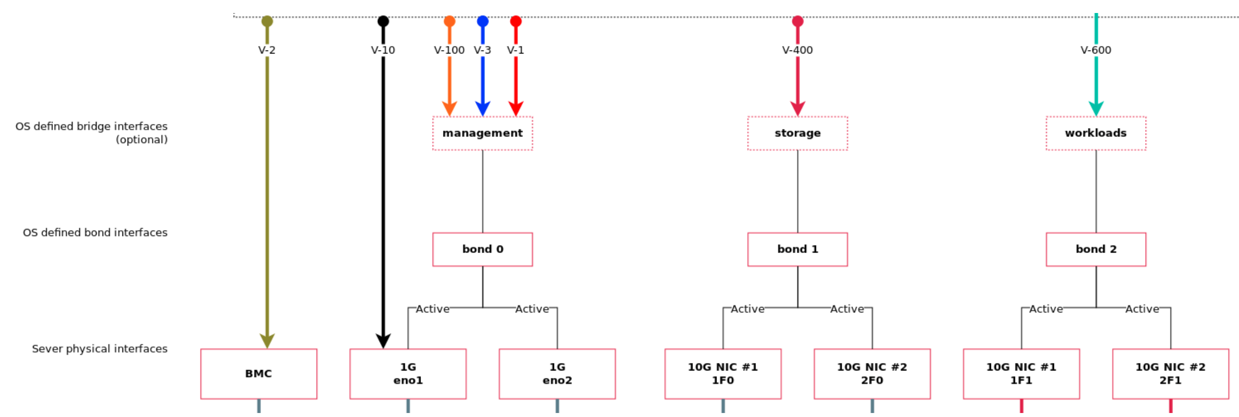

Figure 6.4: Lenovo Open Cloud Plaform Server Network Interfaces

6.3.3 Platform server's NIC to switch mapping

To comply with convention, Open Cloud uses two G8052 (1Gb) switches and two G8272 (10Gb) switches to support high throughput and fault tolerance. Table below shows switch port configurations including `mode`, `native VLAN` (aka. untagged VLAN), `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 will be created per server:

- `bond 0`: `eno1` and `eno2`

[2] Bonding provisioning network is optional because loading an operating system is not a frequent event.

- `bond 1`: `1F0` and `2F0`
- `bond 2`: `1F1` and `2F1`

Server NIC	Server Bond	Server Bridge	Switch Mode	Native VLAN	Tagged VLAN
BMC	n/a	n/a	access	2	n/a
2 x 1G	bond 0	management	trunk	10	1,3,10,100
2 x 10G	bond 1	storage	access	400	n/a
2 x 10G	bond 2	workloads	trunk	600	600

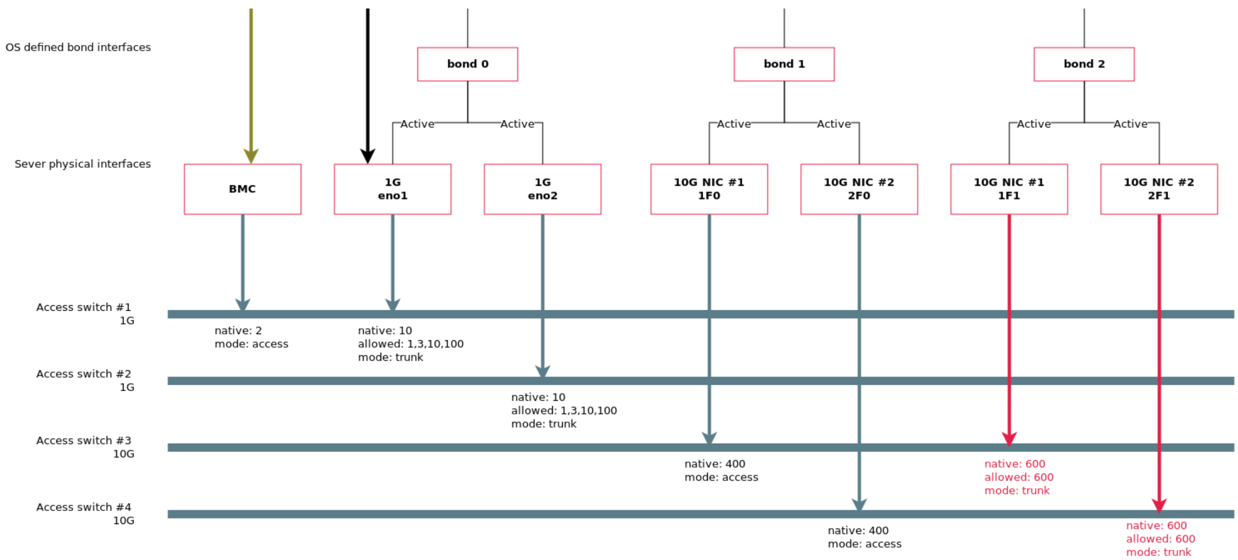

Figure 6.5: Lenovo Open Cloud Plaform Server to Switch

To implement this design takes three steps:

1. Cable server to switch: see Platform Server Cabling Schema
2. Switch port configuration: see Platform Server's Switch Configuration for an example of platform to switch cabling and port configurations.
3. Server network interface configuration: see Platform Server Network Interface Configuration.

6.4 Storage networks

Lenovo Open Cloud supports Ceph storage backend. A storage backend can be shared among multiple workloads and platforms, such as OpenStack. Three new networks are added to Platform Network for Ceph function while leveraging platform services to support Storage hardware and software workloads.

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

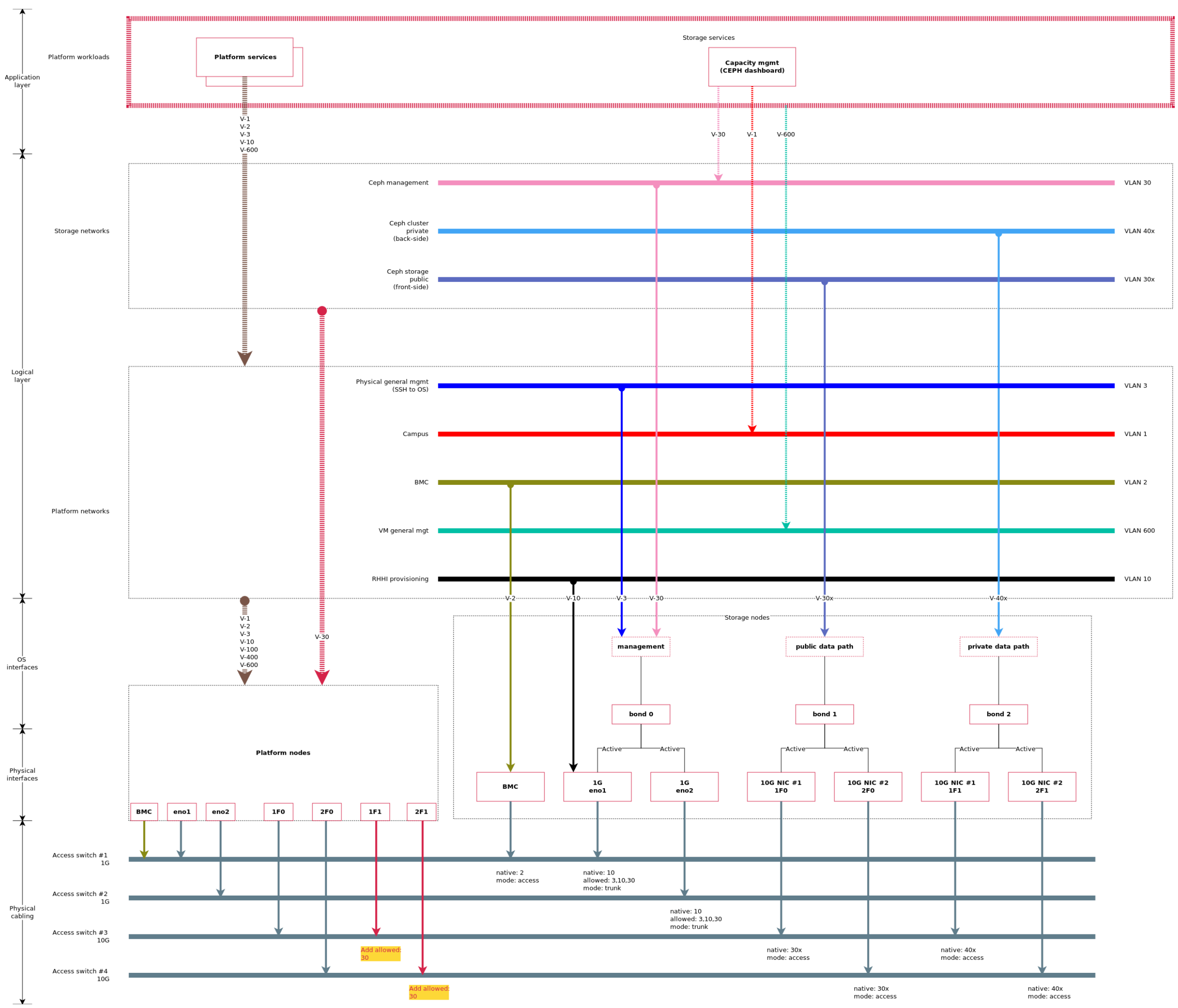

Figure 6.6: Lenovo Open Cloud Storage Networks

Ceph storage public Data transferring between OpenStack nodes (both controller nodes and compute nodes) and Ceph nodes.

Ceph cluster private Ceph private data transferring, e.g. rebalancing.

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
Ceph cluster private	40x	192.168.4.x	254	/24	static	192.168.4.1

6.4.1 Storage services to VLAN mapping

Storage Services	1	30	600
Capacity management	x	x	x

6.4.2 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			public data path
Ceph cluster private	40x				x		private data path

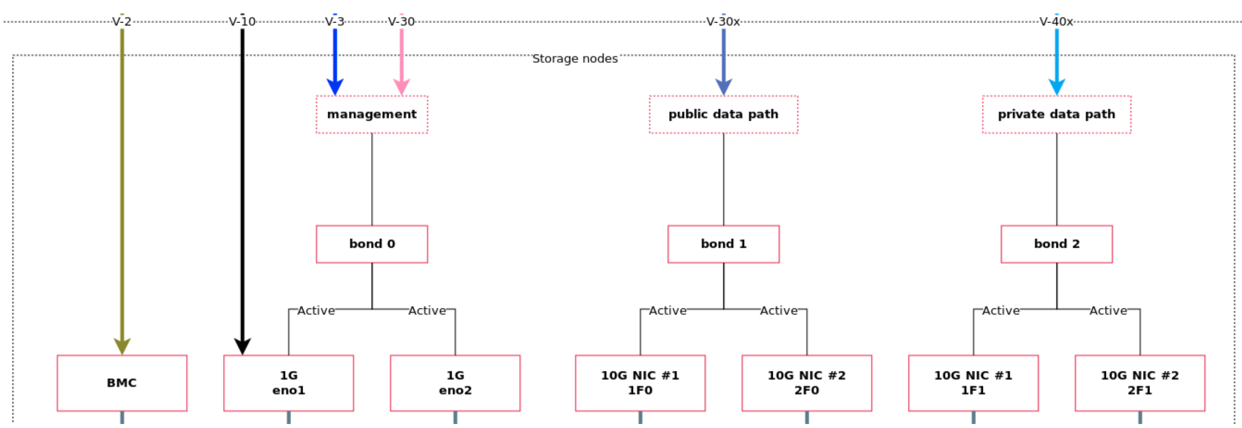

Figure 6.7: Lenovo Open Cloud Storage Server Network Interfaces

6.4.3 Storage server's NIC to switch mapping

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

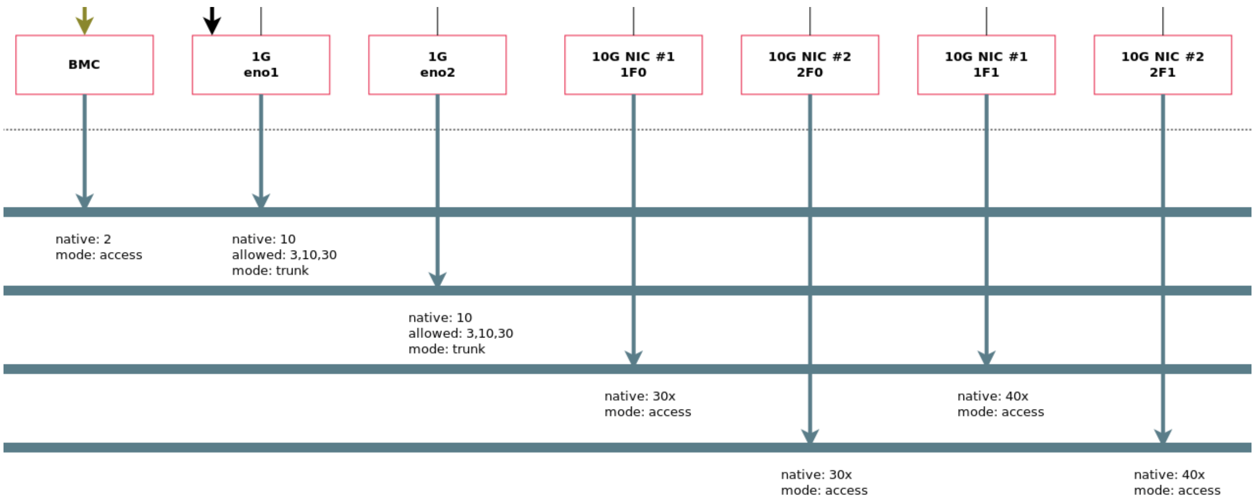

Figure 6.8: Lenovo Open Cloud Storage Server to Switch

6.5 Cloud networks

6.5.1 Cloud VLANs

6.5.2 Cloud server's NIC to VLAN mapping

6.5.3 Configure Cloud server network interfaces

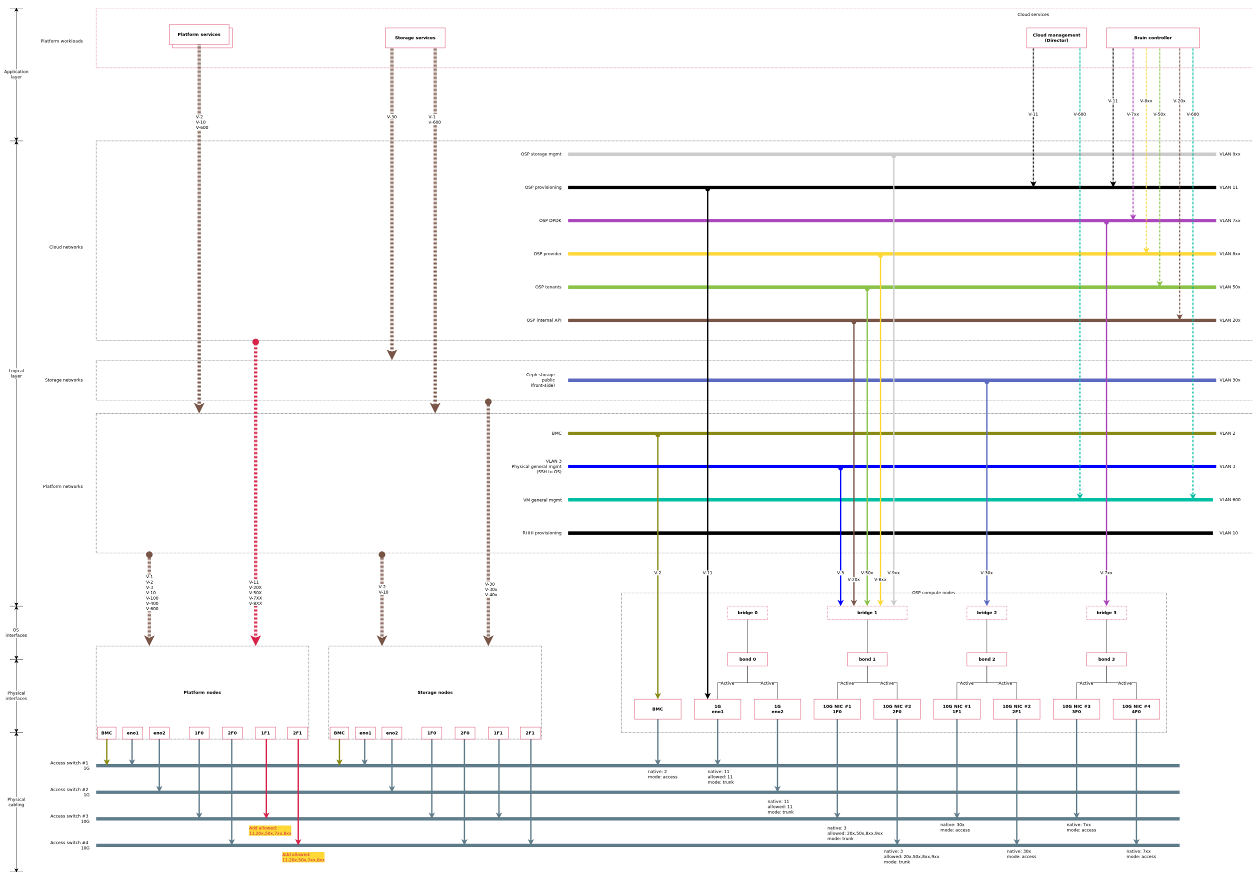

Figure 6.9: Lenovo Open Cloud Cloud Networks

7

Configure Switches

There are two aspects of switch configurations:

1. **inter switch connections**: switch are connected to form a topology allowing data traffic between Lenovo Open Cloud environment and its host environments, and between LOC switches within the LOC itself. All switches are paired for high availability.
2. **server connections**: are connections between server and switch. Except out-of-band connection which has only one connection between a server and a switch, thus does not have redundancy, all other server to switch connections are in pairs.

7.1 Switch to Switch

7.1.1 cable schema

7.1.2 port configurations

7.2 Server to Switch

7.2.1 Platform servers

7.2.1.1 Platform server to switch cable schema (#platform-cabling)

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[1] to demonstrate switch port configurations.

Table 7.1: 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 storage	server 1 storage
2	server 2 BMC		server 2 storage	server 2 storage
3	server 3 BMC		server 3 storage	server 3 storage
4			server 1 workloads	server 1 workloads
5			server 2 workloads	server 2 workloads
6			server 3 workloads	server 3 workloads
17	server 1 management	server 1 management		
18	server 2 management	server 2 management		
19	server 3 management	server 3 management		

7.2.1.2 Platform server switch port configurations

There are multiple methods to apply switch port configurations, see Switch Port Configuration Methods for details. Here we show an example using switch CLI directly.

7.2.1.2.1 To enter switch config mode

In terminal, telnet to either G8052 or G8272 Lenovo switch, and enter config mode by enabling `admin mode`:

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

```
# en <-- enable admin mode
# configure <-- to enter config mode
```

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

7.2.1.2.2 Platform server BMC connections

Table 7.2: Platform server BMC connection switch port config

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

To configure BMC connections, using G8052 port 1 for example. Replace `1/1` with `1/2` for port 2, and `1/3` for port 3.

In switch admin terminal:

```
# interface ethernet 1/1
# bridge-port mode access
# bridge-port access vlan 2
```

7.2.1.2.3 Platform server `management` connections

Table 7.3: Platform server `management` connection switch port config

Server NIC	Server Bridge	Switch Mode	Native VLAN	Tagged VLAN
2 x 1G	management	trunk	10	1,3,10,100

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

In switch's admin terminal:

```
# interface ethernet 1/17
# bridge-port mode trunk
# bridge-port trunk allowed vlan 1,3,10,100
# bridge-port trunk native vlan 10
```

7.2.1.2.4 `storage` connections

Table 7.4: Platform server `storage` connection switch port config

Server NIC	Server Bridge	Switch Mode	Native VLAN	Tagged VLAN
2 x 10G	storage	access	400	n/a

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

In switch's admin terminal:

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

7.2.1.2.5 `workloads` connections

Table 7.5: Platform server `workloads` connection switch port config

Server NIC	Server Bridge	Switch Mode	Native VLAN	Tagged VLAN
2 x 10G	workloads	trunk	600	600

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

In switch's admin terminal:

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

7.2.2 Storage servers

7.2.2.1 Storage server to switch cable schema

7.2.2.2 Storage server switch port configurations

7.2.3 Cloud servers

7.2.3.1 Cloud server to switch cable schema

7.2.3.2 Cloud server switch port configurations

8

Configure Server Network Interfaces

8.1 Platform Server Network Interfaces

As defined in Platform Networks, Platform servers will be configured with three bonding interfaces:

- `bond 0`: `eno1` and `eno2`
- `bond 1`: `1F0` and `2F0`
- `bond 2`: `1F1` and `2F1`

Note that the name of these interfaces, eg. `eno1`, `1F0`, can vary depending on the slot the NIC card and the server side ports you choose to cable with switch. You can use the Implementation Worksheet to create a mapping between your environment and this design.

We will use `bond 0` for example to show steps needed 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, you can further 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 yours.

````
```shell
DEVICE=bond0
BONDING_OPTS='mode=4'
ONBOOT=yes
BOOTPROTO=none
IPADDR=10.240.41.231
NETMASK=255.255.252.0
GATEWAY=10.240.40.1
DNS1=10.240.0.10
```
````

1. `DEVICE`: must be named `bond0`.
2. `BONDING_OPTS`: set to mode 4 “active-active”. Refer to Red Hat Enterprise Linux 7 USING CHANNEL BONDING for more information of bonding modes and their implications.

3. Create `ifcfg-eno1`:

````
```shell
DEVICE=eno1
MASTER=bond0
SLAVE=yes
ONBOOT=yes
```
````

4. Create `ifcfg-eno2`:

````
```shell
DEVICE=eno2
MASTER=bond0
SLAVE=yes
ONBOOT=yes
```
````

9

Configure Virtual Machines network interfaces

10
Appendix

10.1 Implementation Worksheet (questioinnaire)

1. map server interface name –> ifcfg- files

10.2 Hardware BOM

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

10.3 Software BOM

10.3.1 6-server, HCI deployment, 3 year premium

Table 10.1: 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

SKU	Product	Qty
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

10.4 Configure Lenovo Switch Port

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

10.4.1 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>
```
````
````