	Password-protected Kibana web interface for exploring and visualizing log data		
	Audit logging (CADF-compliant) of user operations		
	Automated log file rotation and alarms for monitoring log filesystem consumption		

ck® 5.0: Planning

® 5.0: Registering SUSE Linux

x for Getting Online Updates

d product updates, you need to register and activate your SUSE product with the SUSE Customer Center. It is recommended to register during the installation, since this will enable you to install the system with
are offline or want to skip the registration step, you can register at any time later from the installed system.

organization does not provide a local registration server, registering SUSE Linux requires a SUSE account. In case you do not have a SUSE account yet, go to the SUSE Customer Center home page (*https://scc*

® 5.0: Registering SUSE Linux during the Installation

Installation

ovide the E-mail address associated with the SUSE account you or your organization uses to manage subscriptions. In case you do not have a SUSE account yet, go to the SUSE Customer Center home page (*http*

e you received with your copy of SUSE Linux Enterprise Server. Proceed with Next to start the registration process.

gistered with the SUSE Customer Center. However, if your organization provides local regsitration servers you can either choose one form the list of auto-detected servers or provide the URl at "Register System

online update repositories will be added to your installation setup. When finished, you can choose whether to install the latest available package versions from the update repositories. This ensures that SUSE Lin
ates available. If you choose No, all packages will be installed from the installation media. Proceed with Next.

lly registered during installation, YaST will disable repositories from local installation media such as CD/DVD or flash disks when the installation has been completed. This prevents problems if the installation s
the latest updates from the online repositories.

® 5.0: Registering SUSE Linux from the Installed System

stalled System

istration during the installation or want to re-register your system, you can register the system at any time using the YaST module "Product Registration" or the command line tool "SUSEConnect".

"YaST -> Software -> Product Registation". Provide the E-mail address associated with the SUSE account you or your organization uses to manage subscriptions. In case you do not have a SUSE account yet, g
.com/) to create one.

e you received with your copy of SUSE Linux Enterprise Server. Proceed with Next to start the registration process.

gistered with the SUSE Customer Center. However, if your organization provides local regsitration servers you can either choose one form the list of auto-detected servers or provide the URl at "Register System

nnect

N_CODE> with the Registration Code you received with your copy of SUSE Linux Enterprise Server. Replace <EMAIL_ADDRESS> with the E-mail address associated with the SUSE account you or your orga
vith a local registration server, also provide the URL to the server:

```
-r <REGISTRATION_CODE> -e <EMAIL_ADDRESS> --url "https://suse_register.example.com/"
```

® 5.0: Registering SUSE Linux during Automated Deployment

omated Deployment

s automatically using AutoYaST, you can regsiter the system during the installation by providing the respective information in the AutoYaST control file. Refer to *https://www.suse.com/documentation/sles-12/b*
for details.

® 5.0: Hardware and Software Support Matrix

hardware and software for HPE Helion OpenStack 5.0.

ails about the supported hardware and software for HPE Helion OpenStack 5.0

s

allation or upgrade of a HPE Helion OpenStack release on HPE (ProLiant) servers, the Service Pack for ProLiant (SPP) should be applied to be compatible with latest releases in firmware. The Service Pack for I
w.hpe.com/info/spp

® 5.0: OpenStack Version Information

services have been updated to the *OpenStack Newton* release. See *OpenStack Newton Features* for more details.

® 5.0: Supported Hardware

ware supported in HPE Helion OpenStack 5.0, see *HPE Helion Ready Solution Catalog*.

® 5.0: Supported Hardware Configurations

supports the following hardware configurations for a deployment.

otocols

supports Fibre Channel and FCoE boot from SAN in multipath environments. The following list outlines the current limitations based on testing:

)5 Native Fibre Channel - Up to 1024 paths during boot
Native Fibre Channel - Up to 1024 paths during boot
i0 series - Up to 1024 paths during boot
i4FLB - Up to 1024 paths during boot
i and 630 series - Up to 1024 paths during boot

® 5.0: Cloud Scaling

5.0 a total of 200 total compute nodes in a single region across any of the following hypervisors is supported:

supports a total of 8000 virtual machines across a total of 200 compute nodes.

supports 100 baremetal Ironic nodes in a single region.

® 5.0: Supported Software

currently supports the following ESXi versions:

te 3)

te 1b)

rements for your vCenter server:

3 and above (It is recommended to run the same server version as the ESXi hosts)

Plus license

® 5.0: Notes about Performance

ommendations to ensure good performance of your cloud environment:

des, you will want good I/O performance. Your array controllers must have cache controllers and we advise against the use of RAID-5.
I/O performance will influence the virtual machine start-up performance. We also recommend the use of cache controllers in your storage arrays.
ed object storage (Swift) nodes, in particular the account, container, and object servers, we recommend that your storage arrays have cache controllers.
et the the servers power management setting in the iLO to OS Control Mode. This power mode setting is only available on servers that include the HP Power Regulator.

® 5.0: Disk Calculator

npute-Centric Deployments

ce on how to estimate the amount of disk space required for a compute-centric HPE Helion OpenStack deployment. To accurately estimate the disk space needed, it is important to understand how Helion utilizes
the number of compute nodes, a large portion of the utilization is driven by operational tools, such as monitoring, metering, and logging.

k calculator does not accurately estimate a Swift-centric deployment at this time. For more information on Swift, see the *Recommended minimum hardware requirements for an entry-scale Swift model* topic.

operational tools can be estimated from the following parameters:

odes + Number of VM's running on each compute node
ing monitored or metered + Amount of logs created
operational data (for Elastic Search, Vertica/InfluxDB, and Kafka)

also enable auditing, follow the steps in the *Audit Logging Adjustment* section to enter additional input parameters.

ovides entry scale and scale-out models for deployment. This disk estimation tool, currently in a spreadsheet form, helps you decide which disk model to start from as well as what customizations you need to mee
s also provides default settings and minimum values for the parameters that drive disk size.

odel you want to support based on the calculations.

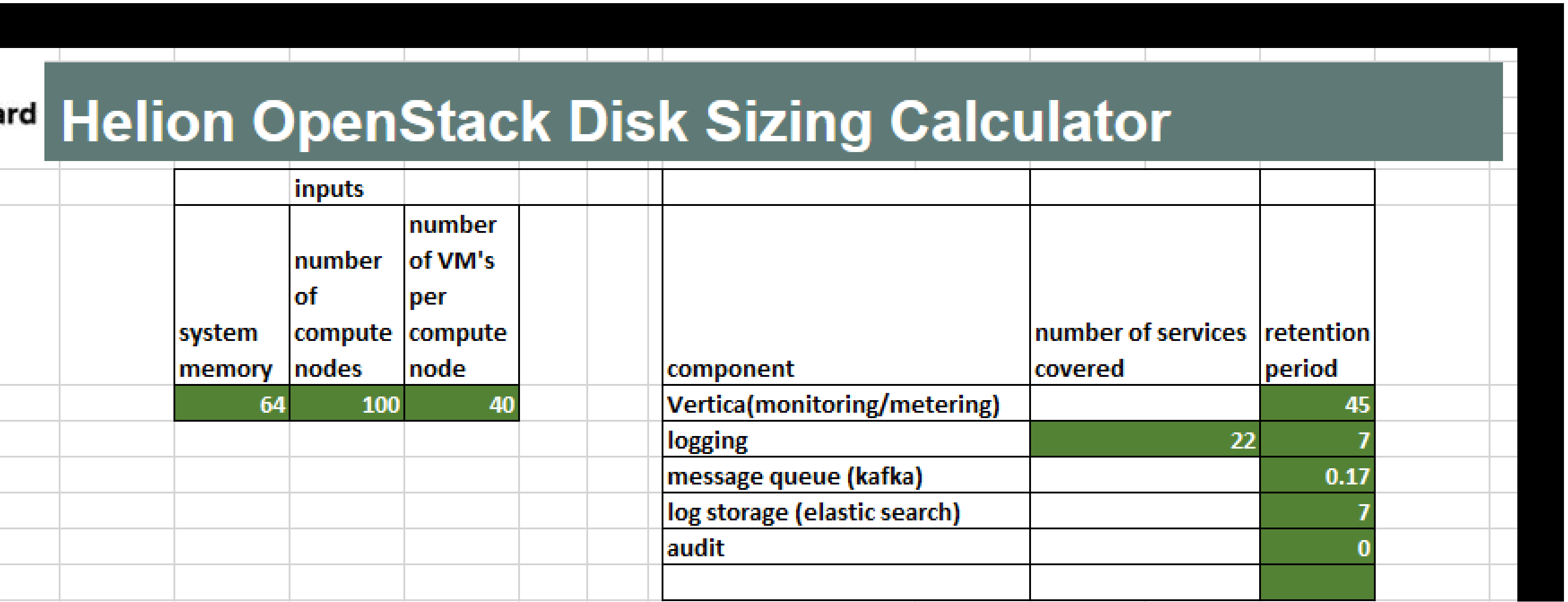
oyment to a disk model example.

sheet automatically displays the minimum requirements for the components that define disk size. You can replace the default values with either the number you have to work with or the number that you want to

want to enable audit logging, follow the steps in the *Audit Logging Adjustment* section to enter additional input parameters.

Input Parameter	Default	Minimum
	64 GB	64 GB
	100	100
	40	40
	45 days retention period	30 days
	22 services covered 7 days retention period	7 days retention period
ge queue)	0.17 of an hour retention period	0.042 of an hour retention period
n (log storage)	7 days retention period	7 days retention period
	0 days retention period	0 days retention period

ws the input parameters in the spreadsheet.



size, replace the default value in the **System Memory** field.
mpute nodes, replace the default in the **Compute Nodes** field.
per of virtual machines per compute node, replace the default in the **VM's per Compute Node** field.
ys you want the metering and logging files retained, replace the default in the **Vertica Retention Period** field.
eplace the default in **Number of Services Covered** and **Retention Period**.

ou enable additional logging of services than those set by default, then you must increase the number in the **Logging Number of Services** Field.

 messages to be retained, replace the default in the **Kafka Retention Period** field.
c Search log file retention, replace the default in the **Elastic Search Retention Period** field.
 logging file retention, replace the default in the **Audit Retention Period** field.

ent

logging, you must enter additional input parameters to ensure there is enough room to retain the audit logs. The following diagram shows the parameters you need to specify in the Disk Calculator spreadsheet.

		API/Core Services	Networking	Swift - Images	MMLB	MySQL/RabbitMQ
number of services on cluster		13	10	5	9	6
r Audit Enabled services on cluster		9	1	1	2	
	subcomponents					
		60				
		64				
		175	134	67	121	81
ing, core		0	0	0	0	60
core services		0	0	0	0	26
		0	0	0	362	0
		0	0	0	141	0
		0	0	0	246	0
	logging					
	BURA					
ing, logging, g		0	0	0	1	0
		7	0	3	1	
		0	0	0	0	0

k size calculations:

es you have enabled to collect audit logging information. This is part of HLM configuration.
dit Enabled services on cluster. Auditing is disabled by default, so these values will initially be 0. If audit logging is enabled, initial suggested values would be 9 for API/Core Services, 1 for Networking, 1 for S

ou enable logging for services beyond the defaults, you must change the **Number of Services on a Cluster** field in the spreadsheet. It is recommended that you increase the total services covered as well as incre
ter. For example, if you enable Apache logs on the core services, then the total would increase to 23 and the api/core services entry would change from 13 to 14.

e space in your estimation, determine the size of the images that will be cached.
ed to store Glance images in the **/var/lib/glance/work_dir** field.

Model

API/Core Services	Networking	Swift	MMLB	MySQL/RabbitMQ
216			573	252
216	195	195	573	252

diagram, if you wanted to choose an Entry Scale MML deployment, the calculator recommends the following disk sizes:

rvice
working)
ge)

bbitMQ

and scale-out cloud models, there is a set of associated disk models that can be used as the basis for your deployment. These models provide examples of pontetial parameters for operational tools and are expect
s. Since each deployment can vary greatly, the disk calculator spreadsheet provides a way to create the basic disk model and customize it to fit the specific parameters your deployment. Once you have estimated
ou can choose which example disk partitioning file to use from the tables below. Keep in mind if you are enabling more options than are listed in the Disk Calculator, or if you want to plan for growth, you will n

or each deployment option based on the expected size of the disk available to the control plane nodes. The available space is then partitioned by percentage to be allocated to each of the required volumes on the
ific set of parameters which can be found in the following tables:

TB
MML Servers: 600 GB, 2 TB, 4.5 TB

s

gle cluster of control plane nodes and all services.

Component	Parameters
	100 This model provides lower than recommended retention and should only be used for POC deployments.

Component	Parameters
	100
	7 day retention
	45 day retention
	7 day retention

scale MML models include seperate control plane nodes for core services, metering/monitoring/logging, and MySQL/RabbitMQ. Optionally you can also seperate out Swift (storage) and Neutron (networking). I
ed on the scale and operational parameters.

Component	Parameters
	100
	7 day retention
	30 day retention **Caution:** 45 days is the default minimum.
	7 day retention
	4 hour retention

Component	Parameters
	200
	7 day retention
	45 day retention
	7 day retention
	12 hour retention

Component	Parameters
	200

	45 day retention
	45 day retention
	12 hour retention

g for Cinder bootable volumes

odel for nodes that will have the cinder volume role make sure that there is sufficient disk space allocated for a temporary space for image conversion if you will be creating bootable volumes.

`ar/lib/cinder` for image conversion and this will be on the root filesystem unless it is explicitly separated. You can ensure there is enough space by ensuring that the root file system is sufficiently large, or l `inder` in the disk model when installing the system.

n issues with creating bootable volumes, see the *Block Storage Troubleshooting* documentation for steps to resolve these issues.

® 5.0: KVM Guest OS Support

een tested by HPE and appears to function properly as a Nova compute virtual machine on HPE Helion OpenStack 5.0.

been officially tested by the operating system vendor, or by HPE under the vendor's authorized program, and will be supported by the operating system vendor as a Nova compute virtual machine on HPE Helion

KVM Guest Operating System	Verified	Certified
		Yes
		Yes
		Yes
		Yes
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	
	Yes	

® 5.0: Ironic Guest OS Support

een tested by HPE and appears to function properly as a bare metal instance on HPE Helion OpenStack 5.0.

been officially tested by the operating system vendor, or by HPE under the vendor's authorized program, and will be supported by the operating system vendor as a bare metal instance on HPE Helion OpenStack

Ironic Guest Operating System	Verified	Certified
	Yes	
	Yes	
	Yes	

® 5.0: Recommended Hardware Minimums for the Example Configurations

for disk, memory (RAM), network interface, and CPU hardware for several of our example configurations.

s

allation or upgrade of a HPE Helion OpenStack release on HPE (ProLiant) servers, the Service Pack for ProLiant (SPP) should be applied to be compatible with latest releases in firmware. The Service Pack for l
w.hpe.com/info/spp

® 5.0: Recommended Hardware Minimums for an Entry-scale KVM with VSA Model

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor
ur hardware.

uirements detailed below can be met with logical drives, logical volumes, or external storage such as a 3PAR array.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	Compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the ' Con
	VSA or OSD (Ceph)	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details.	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

supported network requirements, see *Example Configurations*.

er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
or)	Compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the Com
	ceph-osd	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum)	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	radosgw	2	2 x 600 GB (minimum)	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

® 5.0: Recommended Hardware Minimums for an Entry-scale ESX, KVM with VSA Model

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor ur hardware.

currently supports the following ESXi versions:

te 3)

te 1b)

rements for your vCenter server:

3 and above (It is recommended to run the same server version as the ESXi hosts)

Plus license

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
r)		2	2 X 1 TB (minimum, shared across all nodes)	128 GB (minimum)	2 x 10 Gbit/s +1 NIC (for DC access)	16 C x86_

r)	kvm-compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the V Con
	VSA	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details.	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

® 5.0: Recommended Hardware Minimums for an Entry-scale ESX, KVM with VSA model with Dedicated Cluster for Metering, Monitoring, and Logging

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfo ur hardware.

currently supports the following ESXi versions:

te 3)

te 1b)

rements for your vCenter server:

3 and above (It is recommended to run the same server version as the ESXi hosts)

Plus license

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Core-API Controller	2	• 1 x 600 GB (minimum) - operating system drive • 2 x 300 GB (minimum) - Swift drive	128 GB	2 x 10 Gbit/s with PXE Support	24 C x86_
	DBMQ Cluster	3	• 1 x 600 GB (minimum) - operating system drive • 1 x 300 GB (minimum) - MySQL drive	96 GB	2 x 10 Gbit/s with PXE Support	24 C x86_
	Metering Mon/Log Cluster	3	• 1 x 600 GB (minimum) - operating system drive	128 GB	2 x 10 Gbit/s with one PXE enabled port	24 C x86_
r)		2 (minimum)	2 X 1 TB (minimum, shared across all nodes)	64 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s +1 NIC (for Data Center access)	16 C x86_

or)	kvm-compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 CP (Inte virtu core the V Con
	VSA	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details.	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 CI x86_

® 5.0: Recommended Hardware Minimums for an Ironic Flat Network Model

lo driver, you should ensure that the most recent iLO controller firmware is installed. A recommended minimum for the iLO4 controller is version 2.30.

m hardware requirements are based on the *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacit
out your hardware.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 CI x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 CI x86_
	Compute	1	1 X 600 GB (minimum)	16 GB	2 x 10 Gbit/s with one PXE enabled port	16 C x86_

supported network requirements, see *Example Configurations*.

® 5.0: Recommended Hardware Minimums for an Entry-scale Swift Model

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor
ur hardware.

ft example runs the Swift proxy, account and container services on the three controller servers. However, it is possible to extend the model to include the Swift proxy, account and container services on dedicate
you are using this model, we have included the recommended Swift proxy servers specs in the table below.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 CI x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Swift account/container data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 CI x86_

	swobj	3	If using x3 replication only: • 1 x 600 GB (minimum, see considerations at bottom of page for more details) If using Erasure Codes only or a mix of x3 replication and Erasure Codes: • 6 x 600 GB (minimum, see considerations at bottom of page for more details) **Note:** The disk speeds (RPM) chosen should be consistent within the same ring or storage policy. It's best to not use disks with mixed disk speeds within the same Swift ring.	32 GB (see considerations at bottom of page for more details)	2 x 10 Gbit/s with one PXE enabled port	8 CI x86_
	swpac	3	2 x 600 GB (minimum, see considerations at bottom of page for more details)	64 GB (see considerations at bottom of page for more details)	2 x 10 Gbit/s with one PXE enabled port	8 CI x86_

wift object and proxy, account, container servers RAM and disk capacity needs

mber of hardware configurations. For example, a Swift object server may have just a few disks (minimum of 6 for erasure codes) or up to 70 and beyond. The memory requirement needs to be increased as more
ed is 0.5 GB per TB of storage. For example, a system with 24 hard drives at 8TB each, giving a total capacity of 192TB, should use 96GB of RAM. However, this does not work well for a system with a small n
arge drives. So, if after calculating the memory given this guideline, if the answer is less than 32GB then go with 32GB of memory minimum and if the answer is over 256GB then use 256GB maximum, no need

city needs for the Swift proxy, account, and container (PAC) servers, you should calculate 2% of the total raw storage size of your object servers to specify the storage required for the PAC servers. So, for examp
r and you had an object server setup of 24 hard drives with 8TB each for a total of 192TB and you had a total of 6 object servers, that would give a raw total of 1152TB. So you would take 2% of that, which is 2
ble on your Swift proxy, account, and container (PAC) server cluster. If you had a cluster of three Swift PAC servers, that would be ~8TB each.

mb is that if you are expecting to have more than a million objects in a container then you should consider using SSDs on the Swift PAC servers rather than HDDs.

® 5.0: High Availability

ving topics:

epts Overview

oud ensures that a minimum level of cloud resources are always available on request, which results in uninterrupted operations for users.

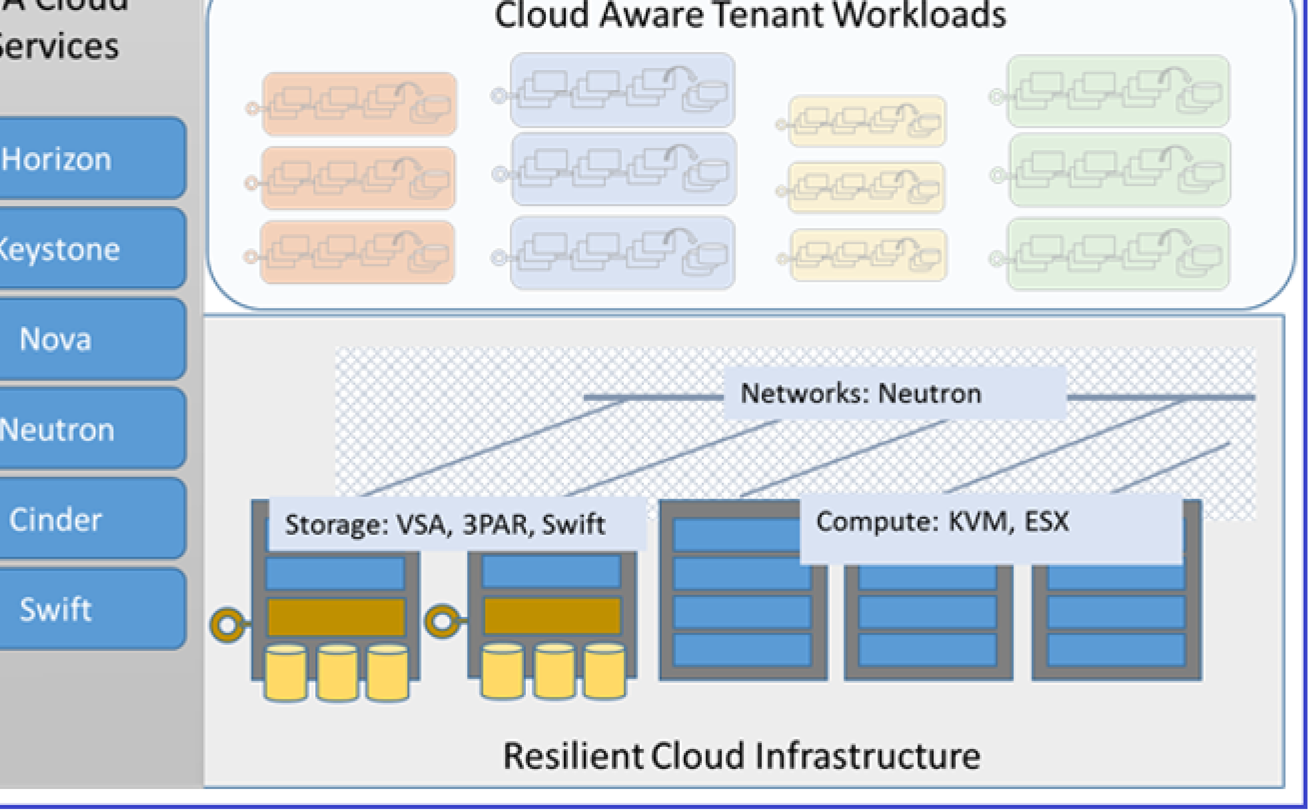
n availability of infrastructure and workloads, we define the scope of HA to be limited to protecting these only against single points of failure (SPOF). Single points of failure include:

rdware failures can take the form of server failures, memory going bad, power failures, hypervisors crashing, hard disks dying, NIC cards breaking, switch ports failing, network cables loosening, and so forth.
er processes can crash due to software defects, out-of-memory conditions, operating system kernel panic, and so forth.

enStack strives to create a system architecture resilient to SPOFs, and does not attempt to automatically protect the system against multiple cascading levels of failures; such cascading failures will result in an un
ed to recover and restore any failed component, as soon as the first level of failure occurs.

nfrastructure

vision and manage the compute, storage, and network infrastructure resources at any given point in time and the Horizon Dashboard and the OpenStack APIs must be reachable and be able to fulfill user requests



, and Network resources are deployed, users expect these resources to be reliable in the following ways:

/M hypervisors/servers hosting a project compute instance (virtual machine) dies and the compute instanceM is lost along with its local ephemeral storage, you will be able to re-launch a fresh compute instance te KVM Hypervisor/server. The following mechanisms exist to ensure that data on compute instances are backed up:

- eate snapshot images of compute instances is available for your root partitions.
- loss is undesirable, the compute instance can be booted from a Cinder volume which can be re-used on new instances.
- orage service volumes can be made highly-available by clustering *(Details below in VSA section below)*
- ct service is always available *(Details in Swift section below)*
- as routers, subnets, and floating IP addresses provisioned by the Networking Operation service are made highly-available via Helion Control Plane redundancy and DVR.

ides these features is called a **Highly Available Cloud Infrastructure**.

ware Tenant Workloads

mpute hypervisors do not support transparent high availability for user applications; as such, the project application provider is responsible for deploying their applications in a redundant and highly available ma bility zones, routed through the load balancers and made highly available through clustering.

Available Cloud-Aware Tenant Workloads.

Infrastructure

infrastructure consists of the following:

ntrollers

ntrollers

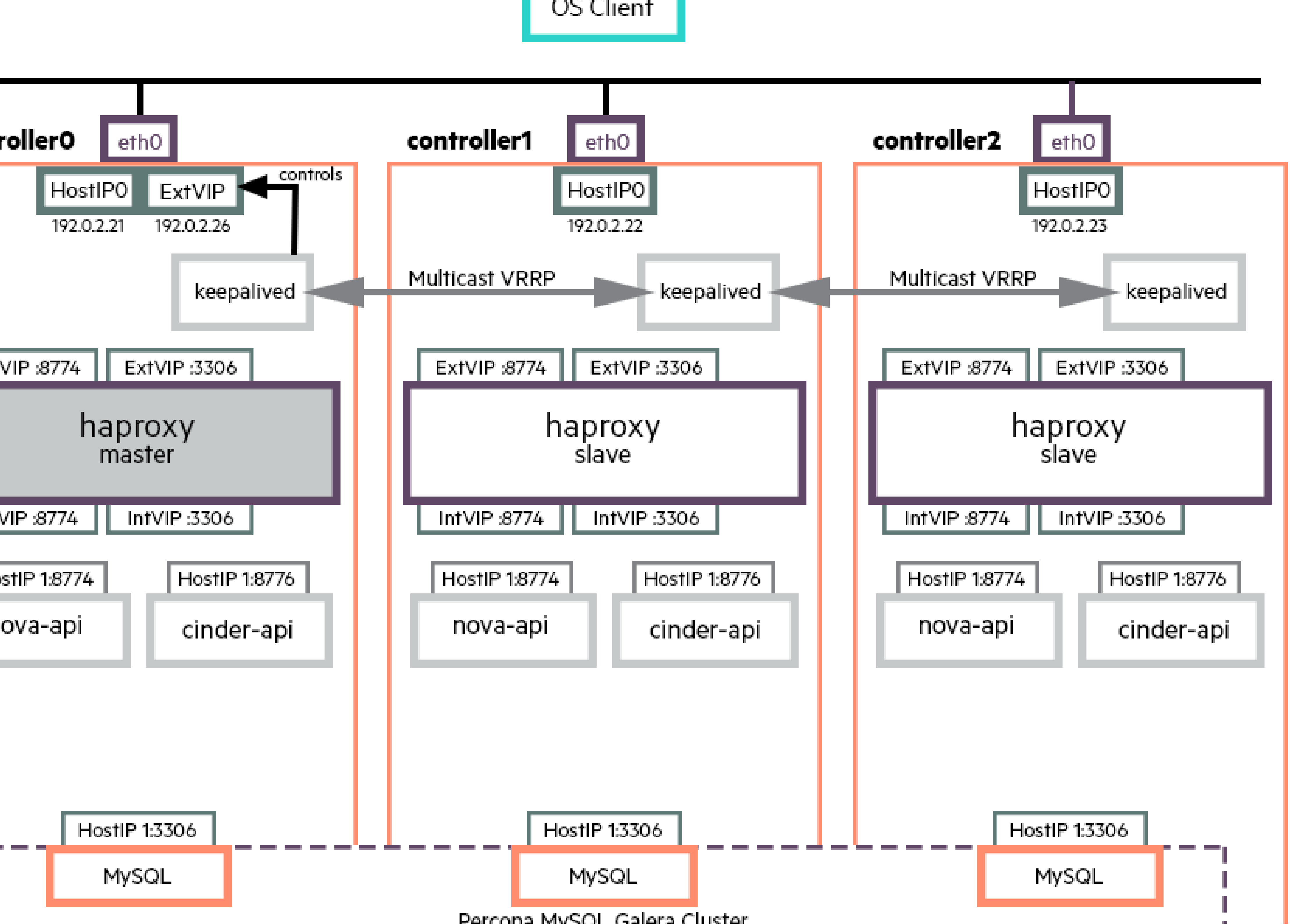
k installer deploys highly available configurations of OpenStack cloud services, resilient against single points of failure.

controller components comes in two main forms.

ess and multiple instances are run across the control plane in active-active mode. The API services (nova-api, cinder-api, etc.) are accessed through the HA proxy load balancer whereas the internal services (nov gh the message broker. These services use the database cluster to persist any data.

proxy load balancer is also run in active-active mode and keepalived (used for Virtual IP (VIP) Management) is run in active-active mode, with only one keepalived instance holding the VIP at any one point in t

the message queue service and the database service is achieved by running these in a clustered mode across the three nodes of the control plane: RabbitMQ cluster with Mirrored Queues and Percona MySQL Ga


OS Client
roller0
eth0
HostIP0
ExtVIP
controls
192.0.2.21
192.0.2.26
keepalived
VIP :8774
ExtVIP :3306
haproxy
master
VIP :8774
IntVIP :3306
stIP 1:8774
HostIP 1:8776
ova-api
cinder-api
HostIP 1:3306
MySQL
controller1
eth0
HostIP0
192.0.2.22
Multicast VRRP
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL
controller2
eth0
HostIP0
192.0.2.23
Multicast VRRP
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster

h is listening for requests on the IP of its host machine, then receives the request and deals with it accordingly. The database service is also accessed through the load balancer. RabbitMQ, on the other hand, is no configured with the set of nodes in the RabbitMQ cluster and failover between cluster nodes is automatically handled by the clients.

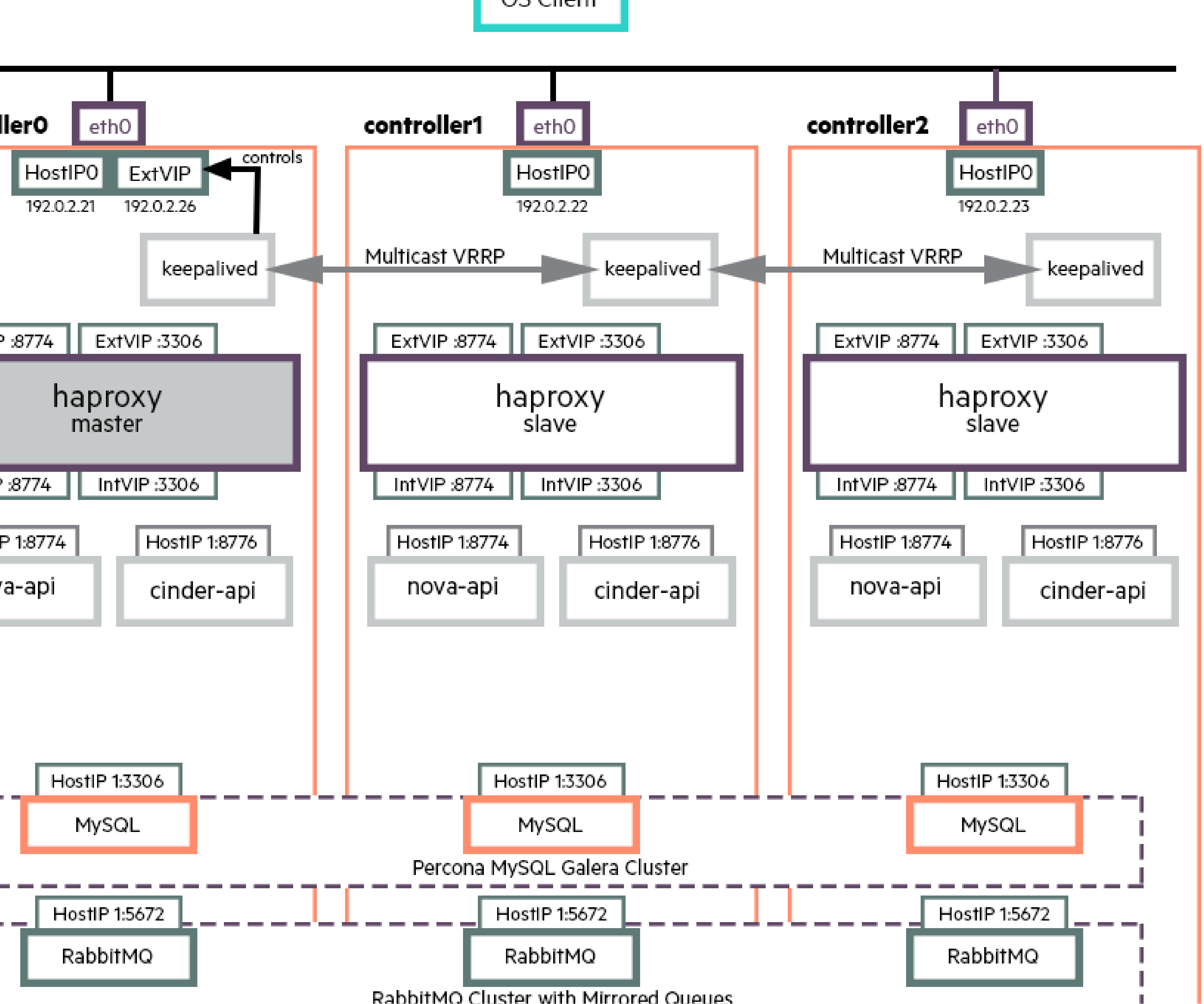
he following topics in detail:

low

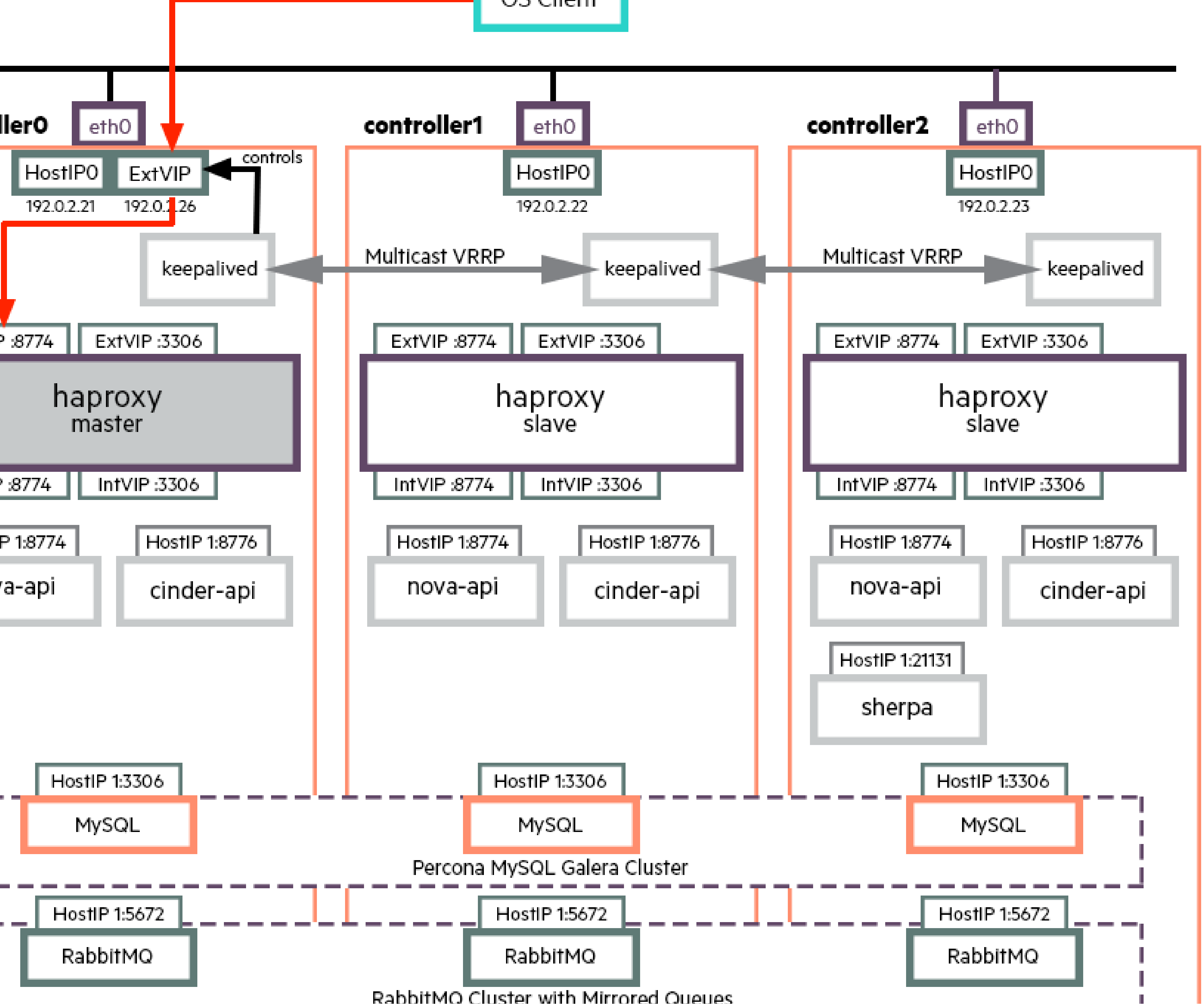
itions

w

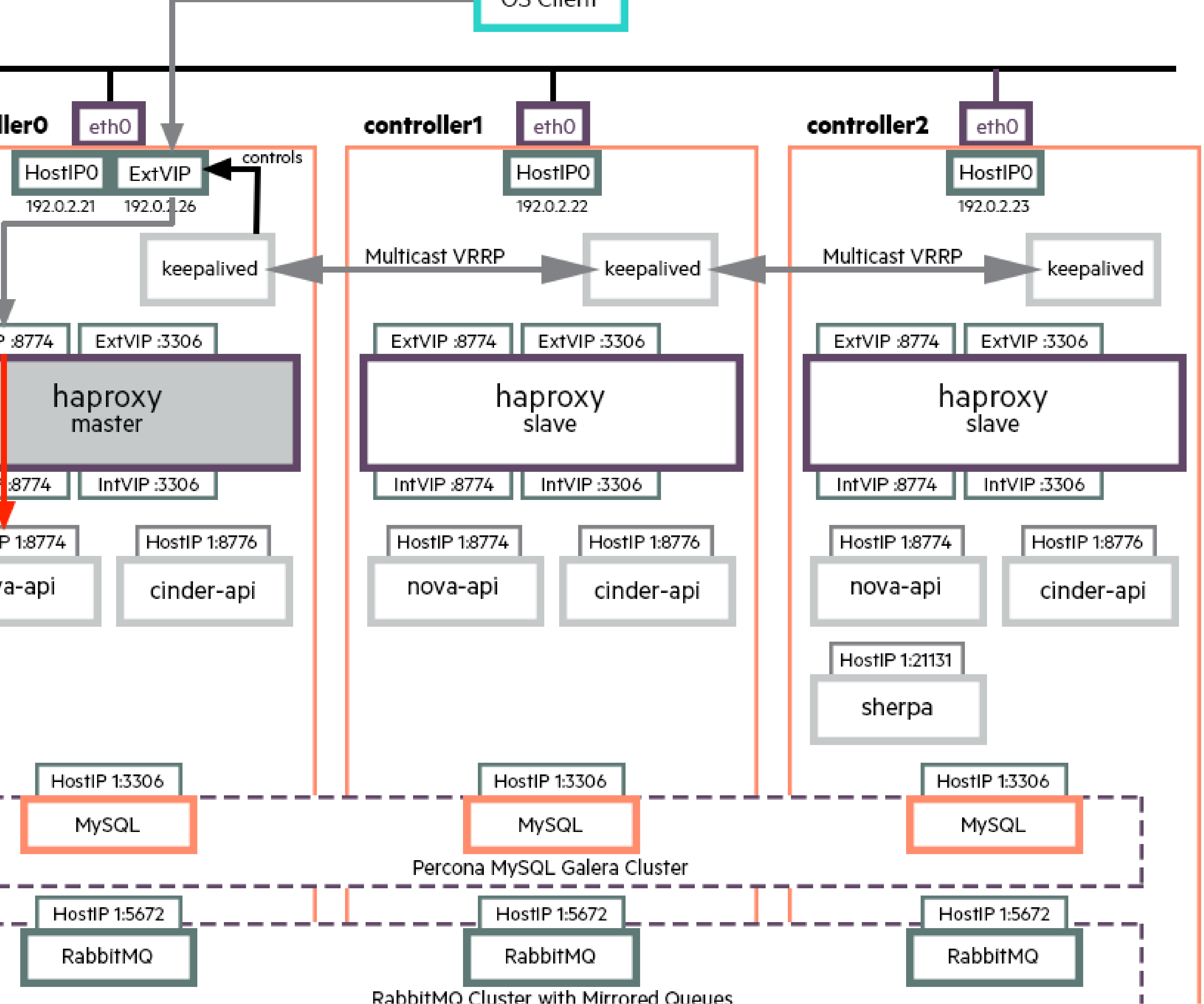
the flow for an API request in an HA deployment. All API requests (internal and external) are sent through the VIP.

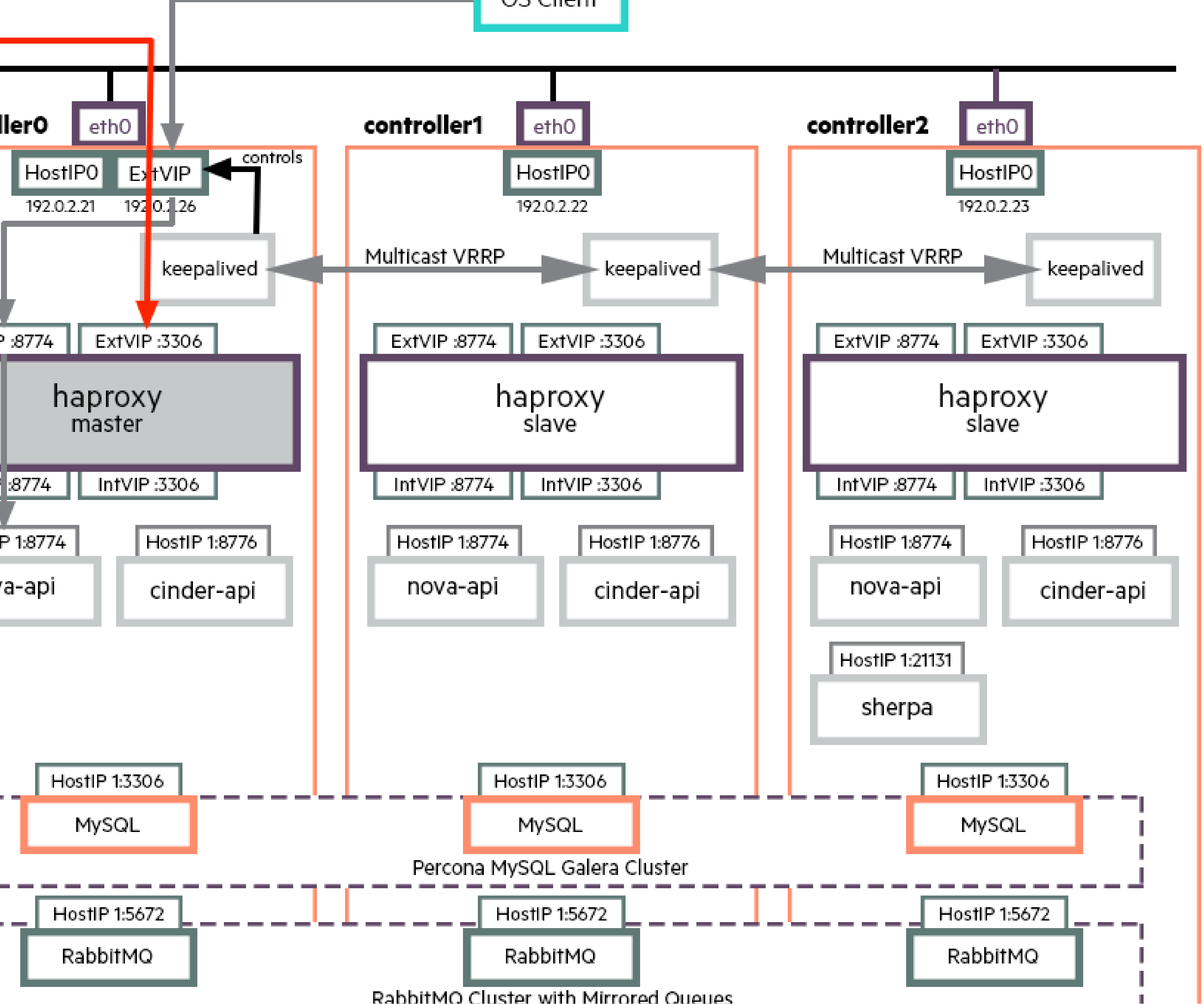


OS Client
ller0
eth0
HostIP0
ExtVIP
controls
192.0.2.21
192.0.2.26
keepalived
P :8774
ExtVIP :3306
haproxy
master
P :8774
IntVIP :3306
P 1:8774
HostIP 1:8776
a-api
cinder-api
controller1
eth0
HostIP0
192.0.2.22
Multicast VRRP
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
controller2
eth0
HostIP0
192.0.2.23
Multicast VRRP
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
RabbitMQ Cluster with Mirrored Queues

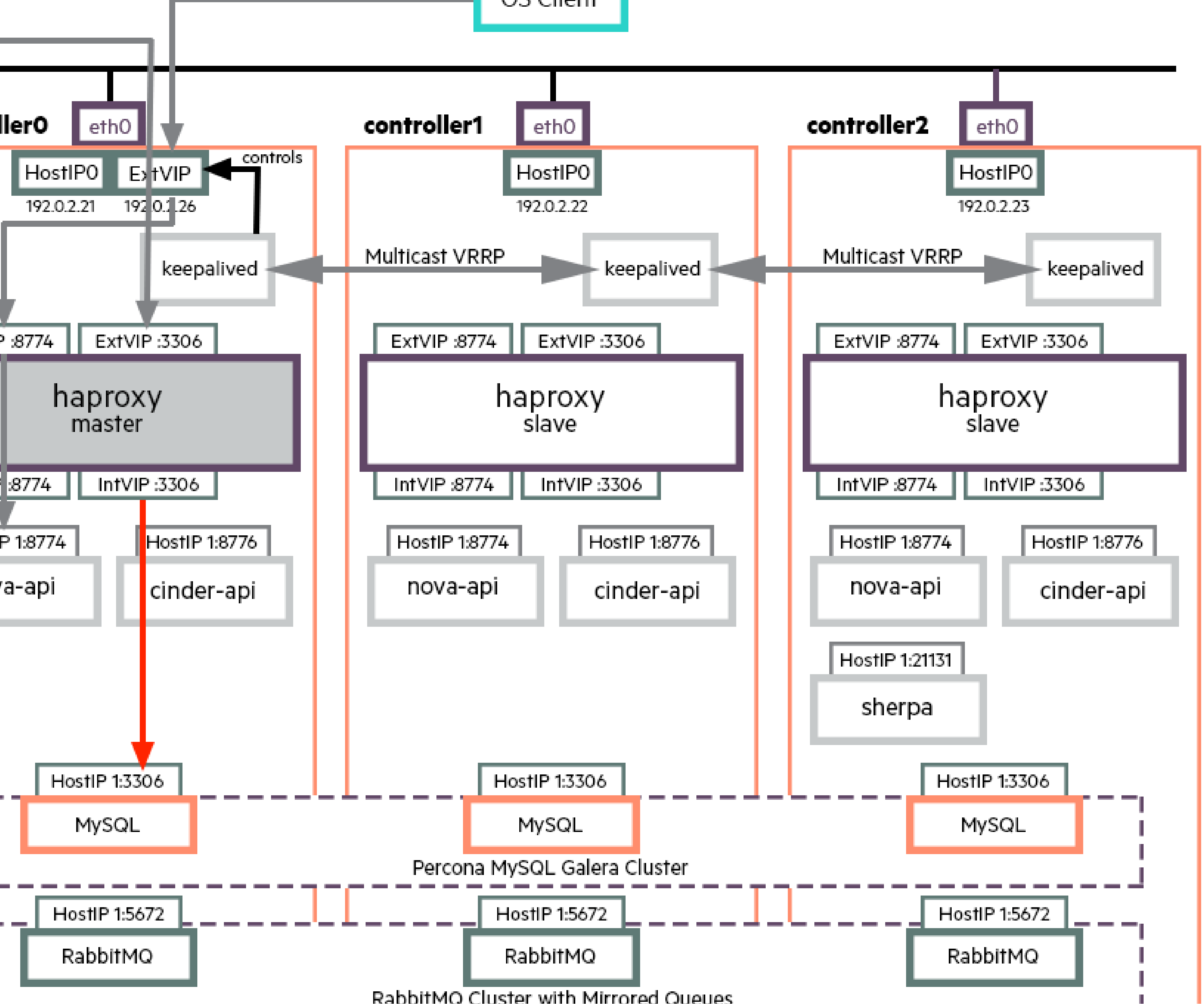


OS Client
ller0
eth0
controller1
eth0
controller2
eth0
HostIP0
ExtVIP
controls
192.0.2.21
192.0.2.26
HostIP0
192.0.2.22
HostIP0
192.0.2.23
keepalived
Multicast VRRP
keepalived
Multicast VRRP
keepalived
P :8774
ExtVIP :3306
ExtVIP :8774
ExtVIP :3306
ExtVIP :8774
ExtVIP :3306
haproxy
master
haproxy
slave
haproxy
slave
P :8774
IntVIP :3306
IntVIP :8774
IntVIP :3306
IntVIP :8774
IntVIP :3306
P 1:8774
HostIP 1:8776
HostIP 1:8774
HostIP 1:8776
HostIP 1:8774
HostIP 1:8776
va-api
cinder-api
nova-api
cinder-api
nova-api
cinder-api
HostIP 1:21131
sherpa
HostIP 1:3306
HostIP 1:3306
HostIP 1:3306
MySQL
MySQL
MySQL
Percona MySQL Galera Cluster
HostIP 1:5672
HostIP 1:5672
HostIP 1:5672
RabbitMQ
RabbitMQ
RabbitMQ
RabbitMQ Cluster with Mirrored Queues


OS Client
ller0
eth0
HostIP0
ExtVIP
controls
192.0.2.21
192.0.2.26
keepalived
Multicast VRRP
keepalived
Multicast VRRP
keepalived
controller1
eth0
HostIP0
192.0.2.22
controller2
eth0
HostIP0
192.0.2.23
ExtVIP :3306
haproxy
master
IntVIP :3306
HostIP 1:8776
cinder-api
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
nova-api
HostIP 1:8776
cinder-api
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
nova-api
HostIP 1:8776
cinder-api
HostIP 1:21131
sherpa
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
RabbitMQ Cluster with Mirrored Queues



OS Client
ller0
eth0
controller1
eth0
controller2
eth0
HostIP0
ExtVIP
controls
192.0.2.21
192.0.2.26
HostIP0
192.0.2.22
HostIP0
192.0.2.23
keepalived
Multicast VRRP
keepalived
Multicast VRRP
keepalived
P :8774
ExtVIP :3306
ExtVIP :8774
ExtVIP :3306
ExtVIP :8774
ExtVIP :3306
haproxy
master
haproxy
slave
haproxy
slave
:8774
IntVIP :3306
IntVIP :8774
IntVIP :3306
IntVIP :8774
IntVIP :3306
P 1:8774
HostIP 1:8776
HostIP 1:8774
HostIP 1:8776
HostIP 1:8774
HostIP 1:8776
va-api
cinder-api
nova-api
cinder-api
nova-api
cinder-api
HostIP 1:21131
sherpa
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
RabbitMQ Cluster with Mirrored Queues



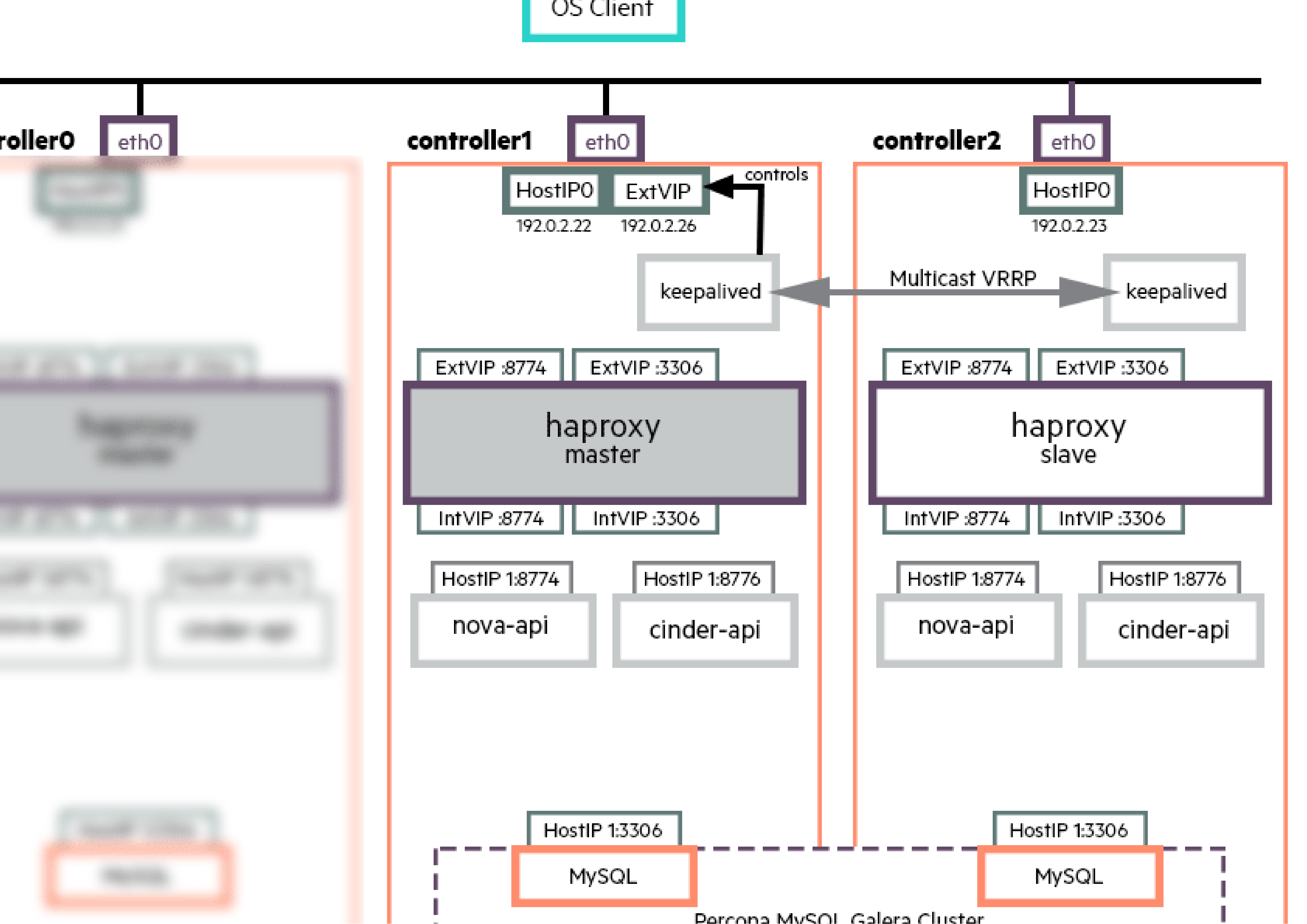
OS Client
ller0
eth0
HostIP0
192.0.2.21
ExtVIP
192.0.2.26
controls
keepalived
Multicast VRRP
P :8774
ExtVIP :3306
haproxy
master
:8774
IntVIP :3306
P 1:8774
HostIP 1:8776
va-api
cinder-api
controller1
eth0
HostIP0
192.0.2.22
keepalived
Multicast VRRP
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
controller2
eth0
HostIP0
192.0.2.23
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:21131
sherpa
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
HostIP 1:5672
RabbitMQ
RabbitMQ Cluster with Mirrored Queues

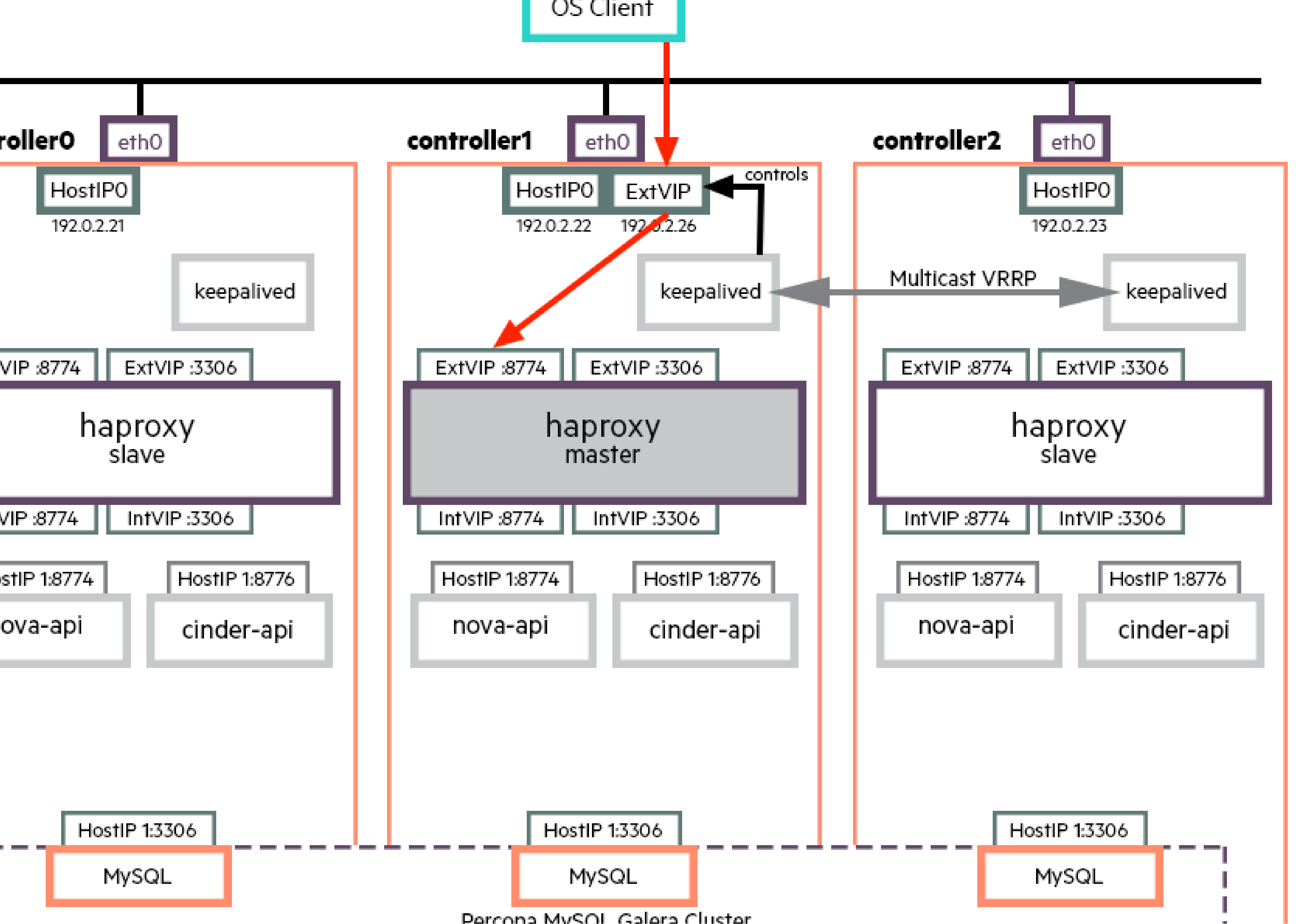
k installer deploys highly available configurations of OpenStack cloud services, resilient against single points of failure. Step through the included flow for an API request in an HA deployment. All API requests
eepalived has currently configured the VIP on the Controller0 node; client sends Nova request to VIP:8774% 2a. HA proxy (listening on VIP:8774) receives the request and selects Controller0 from the list of av
The request is forwarded to the Controller0IP:8774. 2b and 2c are configured Load Balancers% 3. nova-api on Controller0 receives the request and determines that a database change is required. It connects to the
P:3306) receives the database connection request and selects Controller0 from the list of available nodes (Controller0, Controller1, Controller2). The connection request is forwarded to Controller0IP:3306
Flow.png%../../media/ha30/HPE_HA_Flow-1.png%../../media/ha30/HPE_HA_Flow-2.png%../../media/ha30/HPE_HA_Flow-3.png%../../media/ha30/HPE_HA_Flow-4.png

A API Request Message Flow with the following *High Availability Request Flow Diagram*

loss of a controller node is handled as follows:

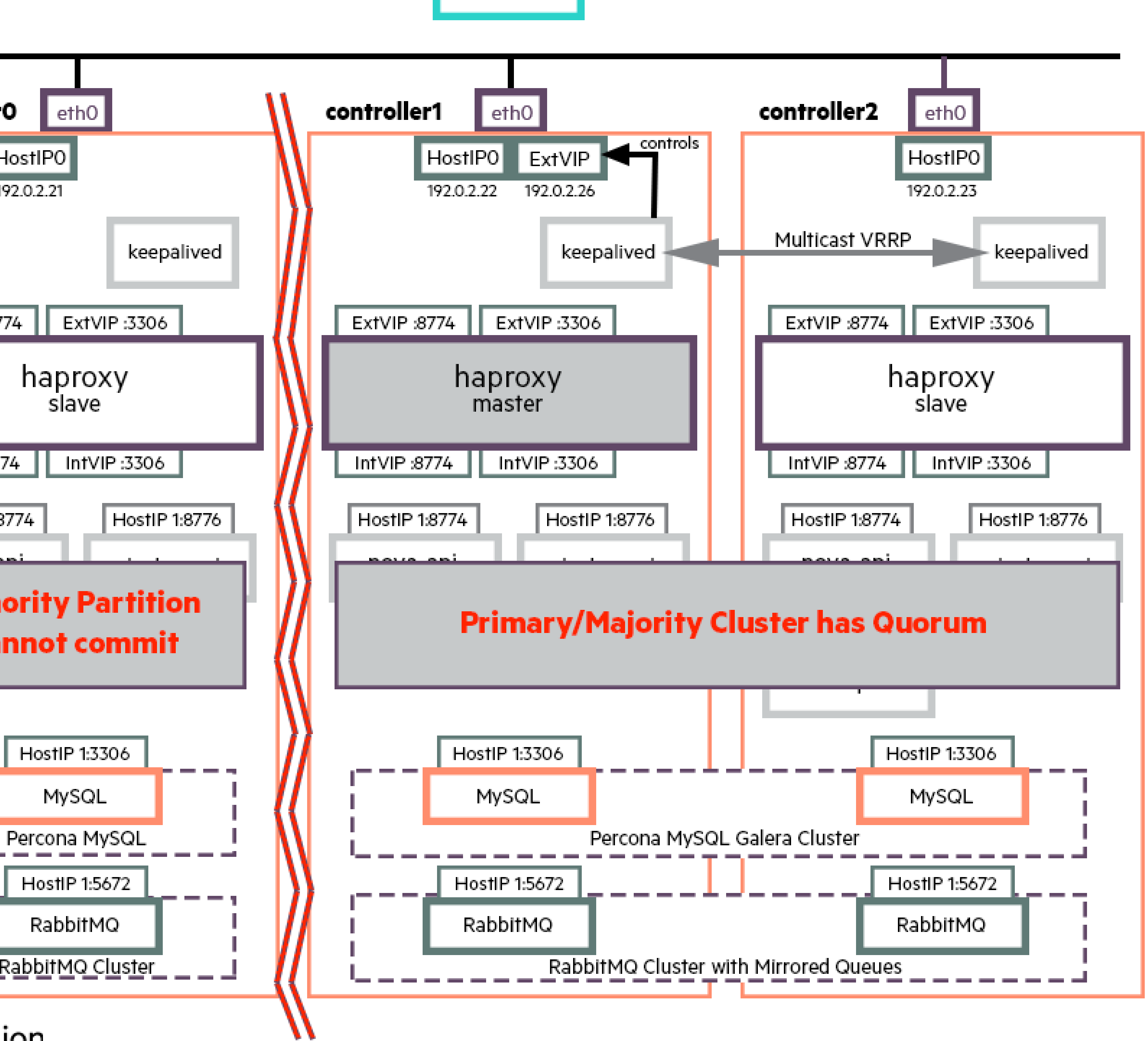
0, which is currently in control of the VIP, is lost, as shown in the diagram below:


OS Client
roller0
eth0
controller1
eth0
controller2
eth0
HostIP0
ExtVIP
controls
192.0.2.22
192.0.2.26
HostIP0
192.0.2.23
keepalived
Multicast VRRP
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
master
IntVIP :8774
IntVIP :3306
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL
HostIP 1:3306
MySQL
Percona MySQL Galera Cluster


OS Client
eth0
HostIP0
192.0.2.21
keepalived
ExtVIP :3306
haproxy
slave
IntVIP :3306
HostIP 1:8776
cinder-api
HostIP 1:3306
MySQL
controller1
eth0
HostIP0
ExtVIP
controls
192.0.2.22
192.0.2.26
keepalived
Multicast VRRP
ExtVIP :8774
ExtVIP :3306
haproxy
master
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL
controller2
eth0
HostIP0
192.0.2.23
keepalived
ExtVIP :8774
ExtVIP :3306
haproxy
slave
IntVIP :8774
IntVIP :3306
HostIP 1:8774
HostIP 1:8776
nova-api
cinder-api
HostIP 1:3306
MySQL

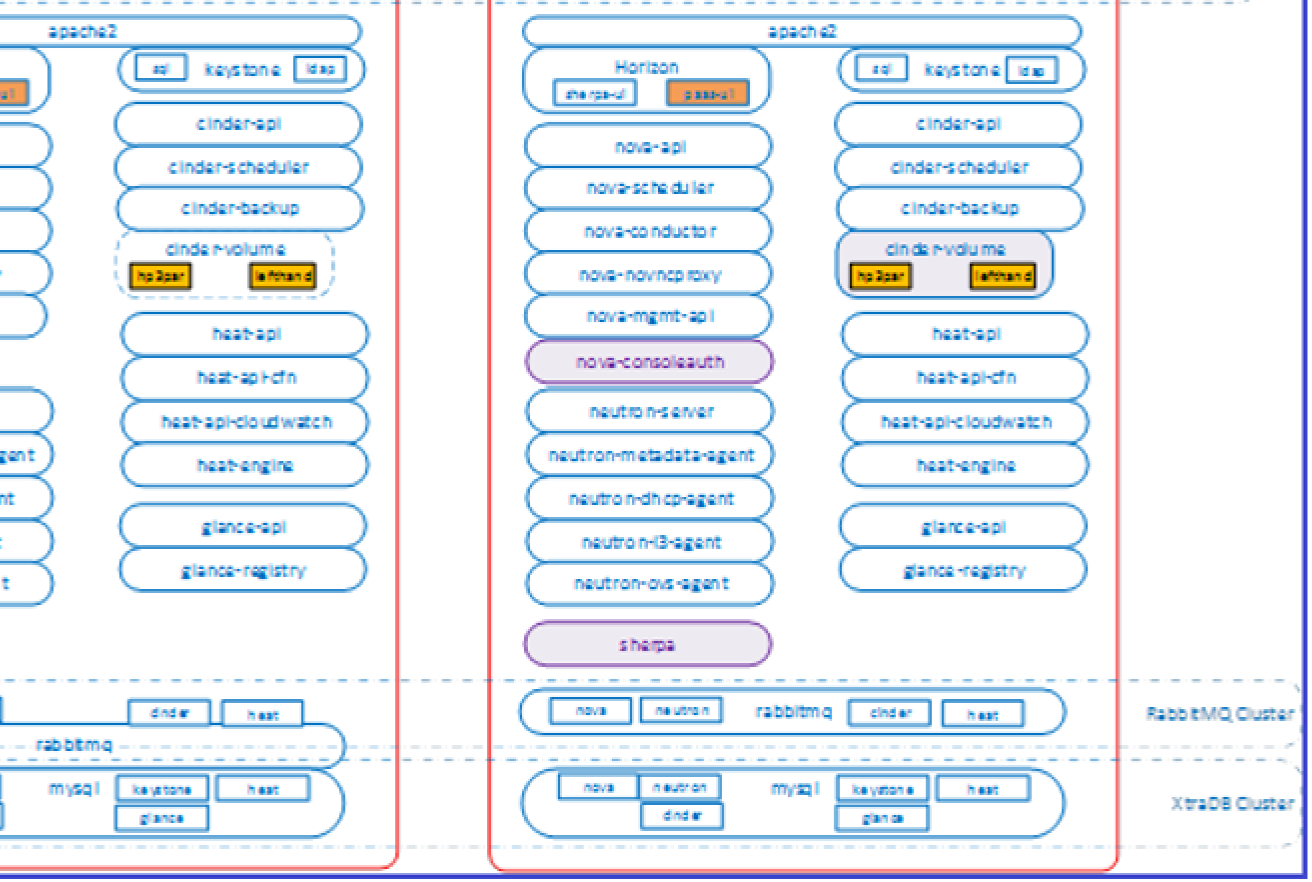
etup to tolerate network failures, specifically those that result in a partition of the cluster, whereby one of the three nodes in the control plane cannot communicate with the remaining two nodes of the cluster. Th
he main HA components of the controller.

artitions is illustrated in the diagram below. Galera has a quorum mechanism so when there is a partition in the cluster, the primary or quorate partition can continue to operate as normal, whereas the non-primary
e example below, Controller0 is partitioned from the rest of the control plane. As a result, requests can only be satisfied on Controller1 or Controller2. Controller0 will continue to attempt to rejoin the cluster:



ion

e errors against the mysql instance on Controller0, it removes that node from its pool for future database requests.



on all three controller nodes, but kept active on only one node at a time. By default, cinder-volume is kept active on the controller. If the controller fails, you must enable and start the cinder-volume service on o
e controller is restored, you must shut down the Cinder volume service from all other nodes and start it on the controller to ensure it runs as a singleton.

nchronized across all the 3 nodes, Cinder volume can be run on any of the nodes at any given time. Ensure that it is run on only one node at a time.

Cinder Volume after controller failure is documented in*HPE Helion OpenStack 5.0: Managing Cinder Volume and Backup Services*.

ova consoleauth service will become unavailable and users will not be able to connect to their VM consoles via VNC. The service will be restored once you restore the controller.

ailed Controller Nodes

ee node controller cluster provides a robust, highly available control plane of OpenStack services. Controllers not running any of the singleton services can be shut down for a short duration for maintenance activ
ntroller running any of the singleton services cannot be shut down without affecting cloud service availability.

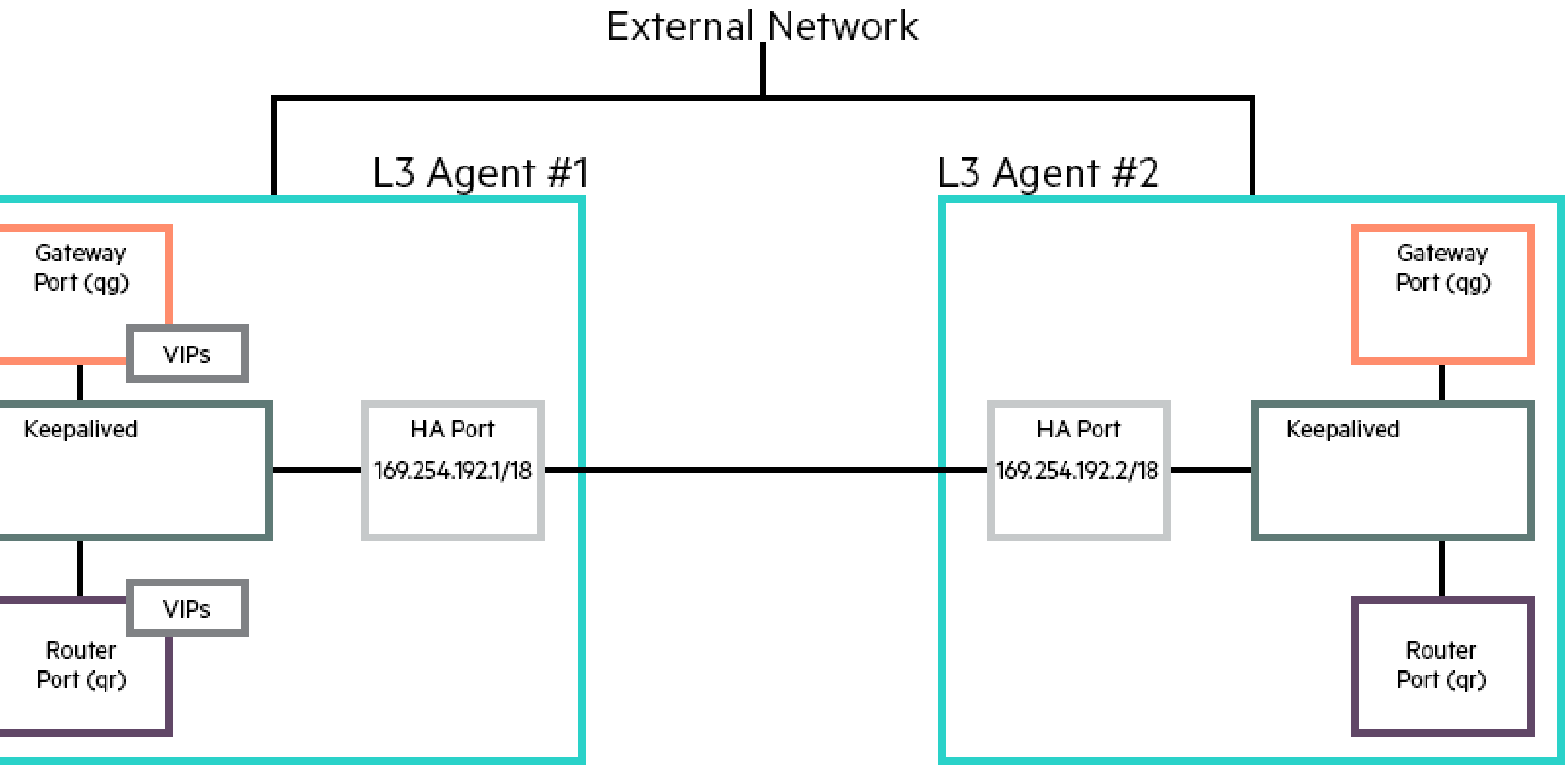
ign is only robust against single points of failure and may not protect you against multiple levels of failure. As soon as first-level failure occurs, you must try to fix the symptom/root cause and recover from the f

ne of the controller servers suffers an irreparable hardware failure, you can decommission and delete it from the cluster. You can then deploy the failed controller on a new server and connect it back into the orig
ion OpenStack 5.0: Replacing a Controller Node.

ng - Centralized

bility into a system involves implementing redundancies in the component that is being made highly available. In Centralized Virtual Router (CVR), that element is the Layer 3 agent a.k.a L3 agent. By making L
nigrated from the primary L3 agent to a secondary L3 agent. The implementation efficiency of an HA subsystem is measured by the number of packets that are lost when the secondary L3 agent is made the maste

HA



es with several benefits.

ailover mechanism does not involve interprocess communication overhead (order of 10s of seconds). By not using an RPC mechanism to invoke the secondary agent to assume the primary agents role enables V

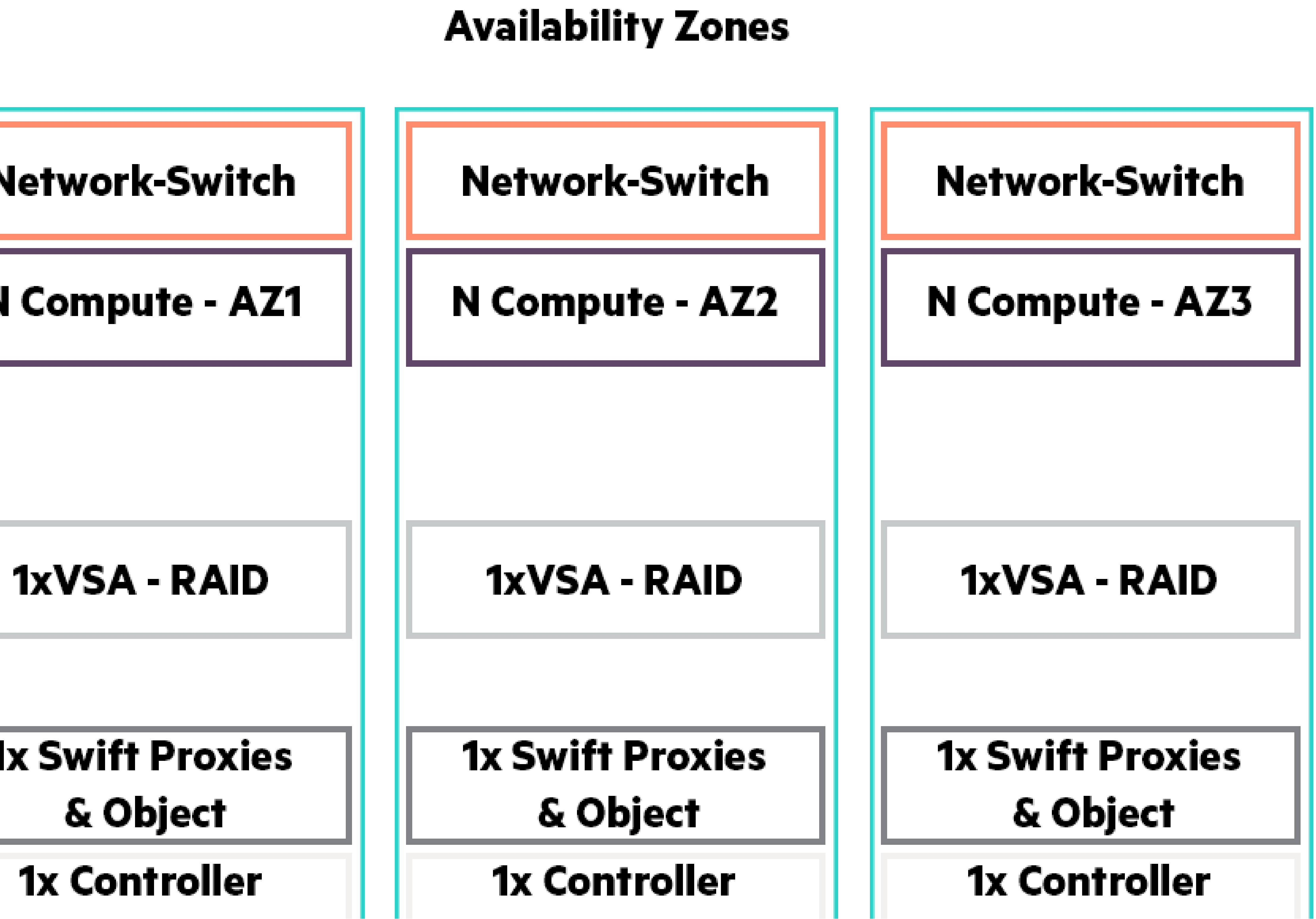
secondary routers are all active. As the routers are running, it is a matter of making the router aware of its primary/master status. This switchover takes less than 2 seconds instead of 60+ seconds it would have ta

 heartbeat link between the primary and secondary. That link in HPE Helion OpenStack 3.0 uses keepalived package of the pacemaker resource manager. The heartbeats are sent at a 2 second intervals between

Virtual Router (DVR) function delivers HA through its distributed architecture. The one centralized function remaining is source network address translation (SNAT), where high availability is provided by DV

on a per router basis and requires that two or more L3 agents capable of providing SNAT services be running on the system. If a minimum number of L3 agents is configured to 1 or lower, the neutron server wi
ents must be running on a control-plane node, L3 agents running on a compute node do not provide SNAT services.

eating HA routers, see: *HPE Helion OpenStack 5.0: Creating a Highly Available Router*


Availability Zones
1xVSA - RAID
& Object
1x Controller
Network-Switch
N Compute - AZ2
1xVSA - RAID
1x Swift Proxies
& Object
1x Controller
Network-Switch
N Compute - AZ3
1xVSA - RAID
1x Swift Proxies
& Object
1x Controller

ble support for these types of availability zones in the current release.

HPE Helion OpenStack is deployed in a single availability zone upon installation. Multiple availability zones can be configured by an administrator post-install, if required. Refer to the *Chapter 5: Scaling* (in the

nova-compute nodes either during initial installation, or by adding compute nodes post initial installation.

es post initial installation, you can specify the target physical servers for deploying the compute nodes.

Compute Nodes after Initial Installation

s

Nova availability zones can be used to segregate Nova compute nodes across different failure zones.

pervisor

n ESX Hypervisor can be made highly available using the HA feature of VMware ESX Clusters. For more information on VMware HA, please refer to your VMware ESX documentation.

reVirtual VSA

ock storage volumes are provided by the network RAID 10 implementation in the HPE StoreVirtual VSA software. You can deploy the VSA nodes in three node cluster and specify Network RAID 10 protection

erating system of the StoreVirtual VSA ensures that the two-way replication maintains two mirrored copies of data for each volume.

ability ensures that failure of any single server does not cause data loss, and maintains data access to the clients.

SA nodes of the cluster can be strategically deployed in different zones of your data center for maximum redundancy and resiliency. For more information on how to deploy VSA nodes on desired target servers, ument.

oss Availability Zones/Racks

age above, the input model example has 3 VSA servers in three different server-groups (Racks) (server-groups are are logical separations). You can configure these server-groups in different physical Racks to pr a 1500 volumes limit per VSA cluster.

er with the 3 new nodes.

es

e not supported for general consumption in the current release.

ift

achieved at two levels.

multiple Swift proxy nodes. Client requests are directed to all Swift proxy nodes by the HA Proxy load balancer in round-robin fashion. The HA Proxy load balancer regularly checks the node is responding, so t wift service will continue to operate and respond to client requests as long as at least one Swift proxy server is running.

n the middle of a transaction, the transaction fails. However it is standard practice for Swift clients to retry operations. This is transparent to applications that use the python-swiftclient library.

oud models contain three Swift proxy nodes. However, it is possible to add additional clusters with additional Swift proxy nodes to handle a larger workload or to provide additional resiliency.

is stored. This happens for account, container and object data. The example cloud models recommend a replica count of three. However, you may change this to a higher value if needed.

replicas of the same item on disk, it ensures that as far as possible, each replica is stored in a different zone, server or drive. This means that if a single server of disk drives fails, there should be two copies of th

ift will continue to store three replicas. The replicas that would normally be stored on the failed drive are “handed off” to another drive on the system. When the failed drive is replaced, the data on that drive is re ocess re-creates the “missing” replicas by copying them to the drive using one of the other remaining replicas. While this is happening, Swift can continue to store and retrieve data.

Applications and Workloads

s to be deployed in the cloud must be aware of the cloud architecture and potential points of failure and architect their applications accordingly for high availability.

eration:

lures and plan for retries

APIs: invocations can fail - you should carefully evaluate the response of each invocation, and retry in case of failures.
die - monitor and restart them
calls can fail - retry should be successful
nnection can hiccup - retry should be successful
your application tiers

ontaining stateless services such as Web application tier or Web service API tier and put them behind load balancers (you must implement your own HA Proxy type load balancer in your application VMs until H e).
ed VMs into different Nova availability zones.
s state information on its local disk (Ephemeral Storage), and you cannot afford to lose it, then boot the VM off a Cinder volume.
apshots of the VM which will back it up to Swift through Glance.
hemeral may get corrupted (but not your backup data in Swift and not your data on Cinder volumes).
pshots of Cinder volumes and also back up Cinder volumes or your data exports into Swift.
own highly available stateful services, use readily available HPE Helion OpenStack platform services such as Designate, the DNS service.

ilable?

leton service, it can only run on a single node at a time. While nova-consoleauth is not high availability, some work has been done to provide the ability to switch nova-consoleauth to another controller node in c
ting Nova-consoleauth can be found in the *HPE Helion OpenStack 5.0: Troubleshooting Compute Service* guide.

up Services

Services are not high availability and started on one controller node at a time. More information on Cinder Volume and Backup Services can be found in *HPE Helion OpenStack 5.0: Managing Cinder Volume*

singleton service, which can only run on a single node at a time. A manual setup process for this job will be required in case of a node failure. More information on enabling the cron job for Keystone on the othe

bility Guide

® 5.0: Third Party Integrations

umentation showing how to integrate HPE Helion OpenStack 5.0 with third party solutions.

umentation showing how to integrate HPE Helion OpenStack 5.0 with third party solutions.

gration

5.0 supports the integration of 3rd-party components with a HPE Helion OpenStack platform deployment, whether that is a completely separate service or a plugin/driver to an existing service in the HPE Helion
integration of a range of different types of content.

5.0: Splunk Integration

nonstrates the possible integration between the HPE Helion OpenStack 5.0 centralized logging solution and Splunk including the steps to setup and forward logs.

® 5.0: Helion Lifecycle Manager Overview

nation on the Input Model and the Example Configurations.

nation on the Input Model and the Example Configurations.

® 5.0: Input Model

t Model
5.0 Concepts

nd Regions

® 5.0: Introduction to the Input Model

ow the HPE Helion OpenStack input model can be used to define and configure the cloud.

nfiguration of the cloud
on
nments

ed by the configuration processor which parses and validates the input model and outputs the effective configuration that will be deployed to each server that makes up your cloud.

as follows:

ns the ideas behind the declarative model approach used in HPE Helion OpenStack 5.0 and the core concepts used in describing that model
tion provides a description of each of the configuration entities in the input model
s section we provide samples and definitions of some of the more important configuration entities

enStack 5.0

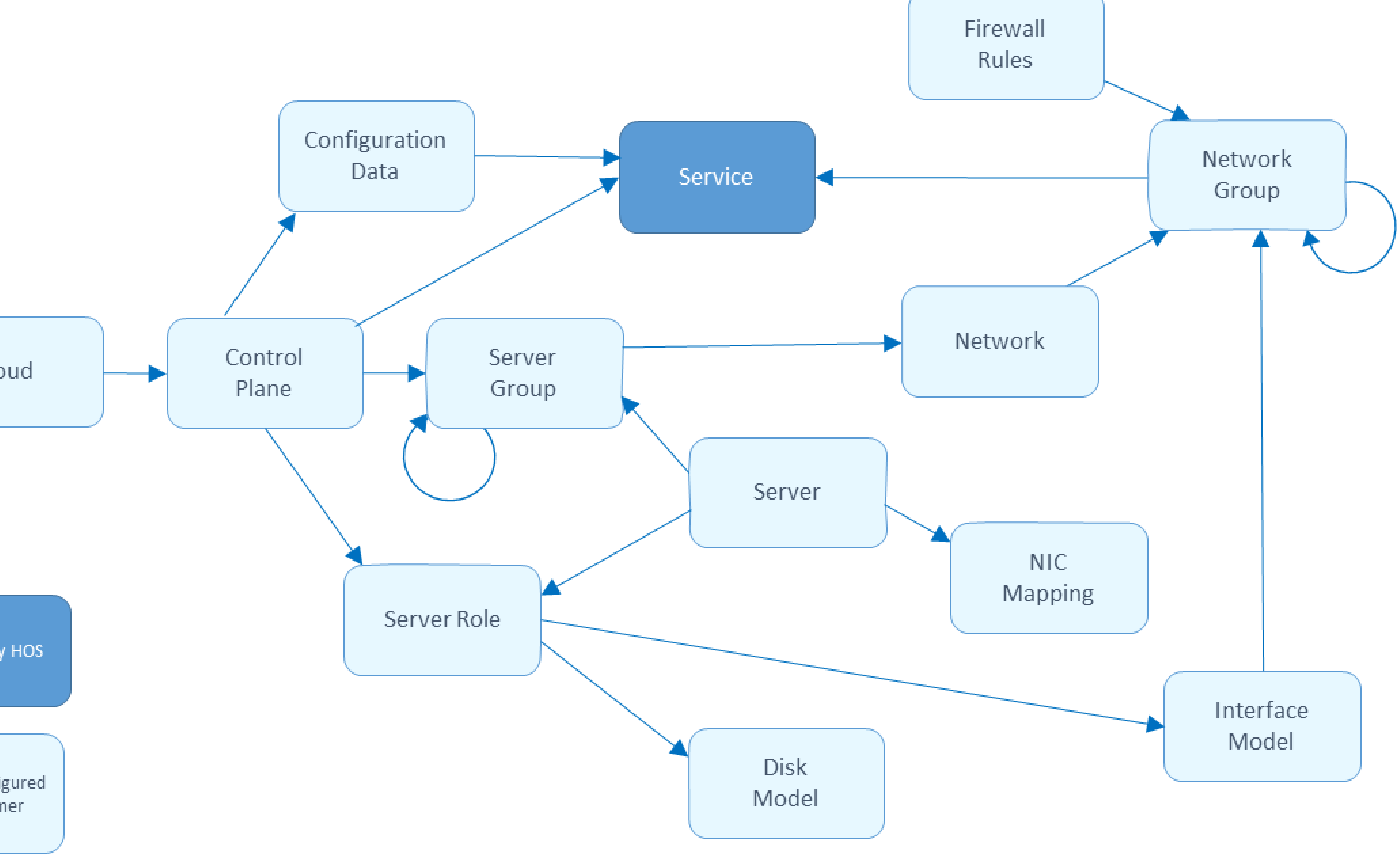
introduces the following additions to the cloud model:

Magnum) Support has been included.
e addition.

k® 5.0: Concepts

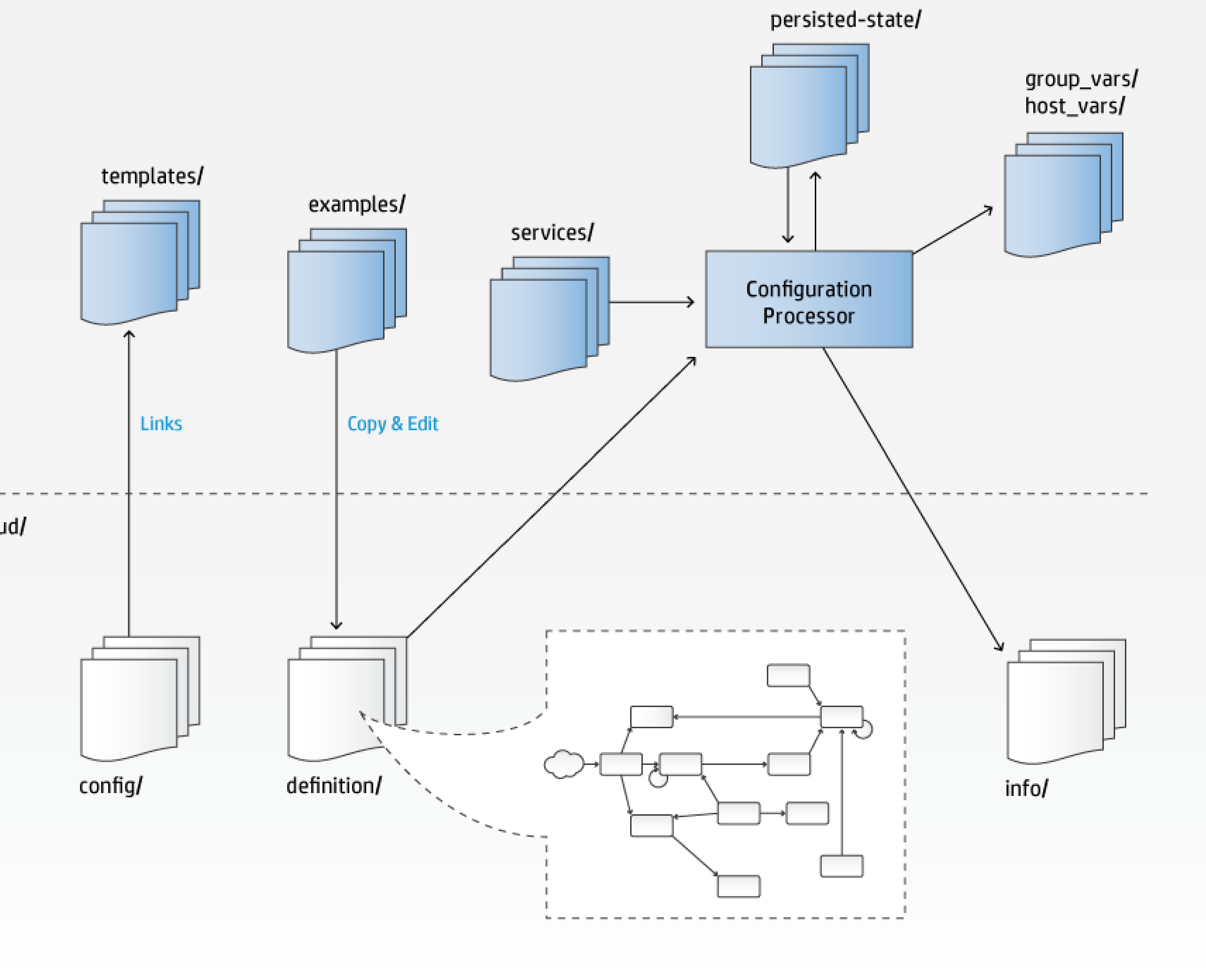
5.0 cloud is defined by a declarative model that is described in a series of configuration objects. These configuration objects are represented in YAML files which together constitute the various example configu
mples can be used nearly unchanged, with the exception of necessary changes to IP addresses and other site and hardware-specific identifiers. Alternatively, the examples may be customized to meet site requirem

ws the set of configuration objects and their relationships. All objects have a name that you may set to be something meaningful for your context. In the examples these names are provided in capital letters as a c
OpenStack, rather it is the relationships between them that define the configuration.



r reads and validates the input model described in the YAML files discussed above, combines it with the service definitions provided by HPE Helion OpenStack and any persisted state information about the curr n be used to deploy the cloud. It also produces a set of information files that provide details about the configuration.

he file systems on the HPE Helion OpenStack deployment server and the configuration processor is shown in the following diagram. Below the line are the directories that you, the cloud administrator, interact w ned by HPE Helion OpenStack.



on

m the ~/helion/my_cloud/definition directory. Although the supplied examples use separate files for each type of object in the model, the names and layout of the files have no significance to the con this directory. Cloud administrators are therefore free to use whatever structure is best for their context. For example, you may decide to maintain separate files or sub-directories for each physical rack of server

s use the conventional upper casing for object names, but these strings are used only to define the relationship between objects. They have no specific significance to the configuration processor.

r more **services** *distributed across* **clusters** *and* **resource groups**.

s with a particular **server-role**.

he operating environment for a set of **services**; normally consisting of a set of shared services (MySQL, RabbitMQ, HA Proxy, Apache, etc), OpenStack control services (API, schedulers, etc) and the **resources**

single **control-plane** which runs all of the **services**. A more complex cloud may have multiple **control-planes** to allow for more than one instance of some services. Services that need to consume (use) another
consuming Neutron) always use the service within the same **control-plane**. In addition a control-plane can describe which services can be consumed from other control-planes. It is one of the functions of the co
e sure that each consumer/service is provided with the configuration details to connect to the appropriate provider/service.

tured as **clusters** and **resources**. The **clusters** are typically used to host the OpenStack services that manage the cloud such as API servers, database servers, Neutron agents, and Swift proxies, while the **resourc**
Nova-Compute or Swift-Object services. This is a representation convenience rather than a strict rule, for example it is possible to run the Swift-Object service in the management cluster in a smaller-scale clou

r more **servers** and you can have one or more **clusters** depending on the capacity and scalability needs of the cloud that you are building. Spreading services across multiple **clusters** provides greater scalability,
mon pattern for a large cloud is to run high data volume services such as monitoring and logging in a separate cluster. A cloud with a high object storage requirement will typically also run the Swift service in its

a mechanism for grouping service components in physical servers, but all instances of a component in a **control-plane** work collectively. For example, if HA Proxy is configured to run on multiple clusters withi
ork as a single instance of the ha-proxy service.

es define the type (via a list of **server-roles**) and number of servers (min and max or count) they require.

define a list of failure-zones (**server-groups**) from which to allocate servers.

ntrol Planes and Regions

ns is a collection of URLs that together provide a consistent set of services (Nova, Neutron, Swift, etc). Regions are represented in the Keystone identity service catalog and clients can decide which region they

egions provide a way of segmenting resources for scale, resilience, and isolation.

lane cloud, there is no need for a separate region definition and the control-plane itself can define the region name.

5.0: Services

r more **services**.

f **service-components** that provide a particular feature; for example, Nova provides the compute service and consists of the following service-components: nova-api, nova-scheduler, nova-conductor, nova-novn
tion/identity service Keystone, only consist of a single service-component.

u need to know about a service are the names of the **service-components**. The details of the services themselves and how they interact with each other is captured in service definition files provided by HPE Hel

Helion OpenStack cloud you have to decide where components will run and how they connect to the networks. For example, should they all run in one **control-plane** sharing common services or be distributed
f some services? The HPE Helion OpenStack supplied examples provide solutions for some typical configurations.

ed in the **control-plane**. How they connect to networks is defined in the **network-groups**.

5.0: Server Roles

servers *with a particular set of* **server-role***s*.

the services on physical **servers**, and you're going to need a way to specify which type of servers you want to use where. This is defined via the **server-role**. Each **server-role** describes how to configure the phy
e. You'll generally use a different role whenever the servers are physically different (have different disks or network interfaces) or if you want to use some specific servers in a particular role (for example to cho
control plane).

tionship to four other entities - the disk-model, the interface-model, the memory-model and the cpu-model:

ies how to configure and use a server's local storage. The disk model is described in the next section.
scribes how a server's network interfaces are to be configured and used. This is covered in more details in the networking section.

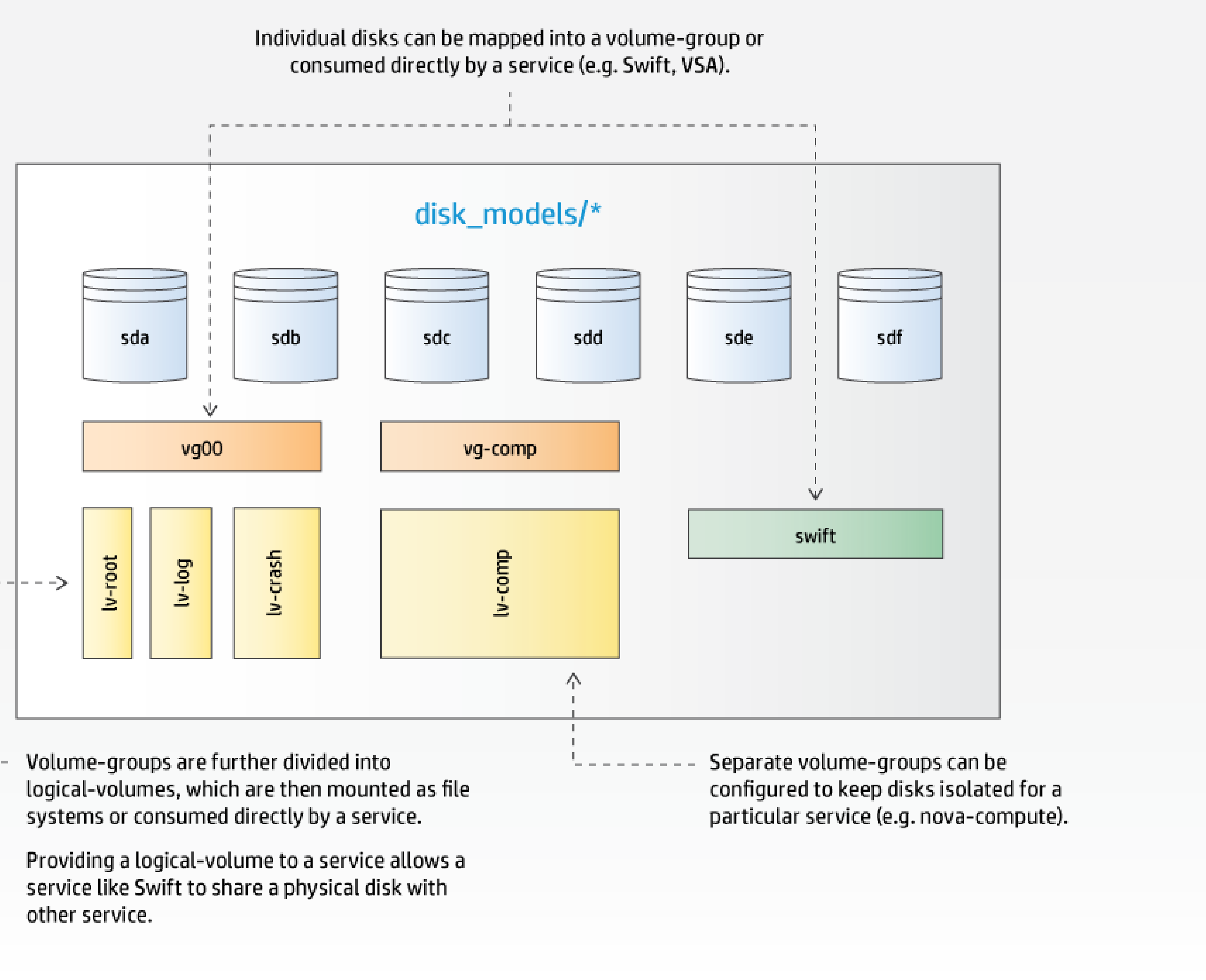
*ed by **services**.*

*l into **logical-volumes**.*

ted as file systems or consumed by services.

cal storage is to be configured and presented to **services**. Disk-models are identified by a name, which you will specify. The HPE Helion OpenStack examples provide some typical configurations. As this is an a a server and the number of disks available, it is impossible to cover all possible permutations you may need to express via modifications to the examples.

devices are assigned to either a **device-group** or a **volume-group**.



on

one or more disks that are to be consumed directly by a service. For example, a set of disks to be used by Swift. The device-group identifies the list of disk devices, the service, and a few service-specific attribute
in the case of Swift this is the ring names). When a device is assigned to a device-group, the associated service is responsible for the management of the disks. This management includes the creation and mounti
grity when it has full control over the file systems and mount points.)

al - it is valid to have a server role without a memory model.

? 5.0: CPU Model

PUs of a server will be used. The model allows CPUs to be assigned for use by components such as Nova (for VMs) and Open vSwitch (for DPDK). It also allows those CPUs to be isolated from the general kern

it is valid to have a server role without a cpu model.

? 5.0: Servers

which determines how they will be used in the cloud.

l) enumerate the resources available for your cloud. In addition, in this definition file you can either provide HPE Helion OpenStack with all of the details it needs to PXE boot and install an operating system ont
tem installation tooling you can simply provide the details needed to be able to SSH into the servers and start the deployment.

he server will be the one used by HPE Helion OpenStack for lifecycle management and must be part of a network which is in the input model. If you are using HPE Helion OpenStack to install the operating syste
server must be installed manually from the HPE Helion OpenStack ISO and this server must be included in the input model as well.

letails used to install or connect to the server, each server defines what its **server-role** is and to which **server-group** it belongs.

? 5.0: Server Groups

*a **server-group**.*

rver-groups as failure zones for server allocation.

*ociated with a list of **networks**.*

*other **server-groups**.*

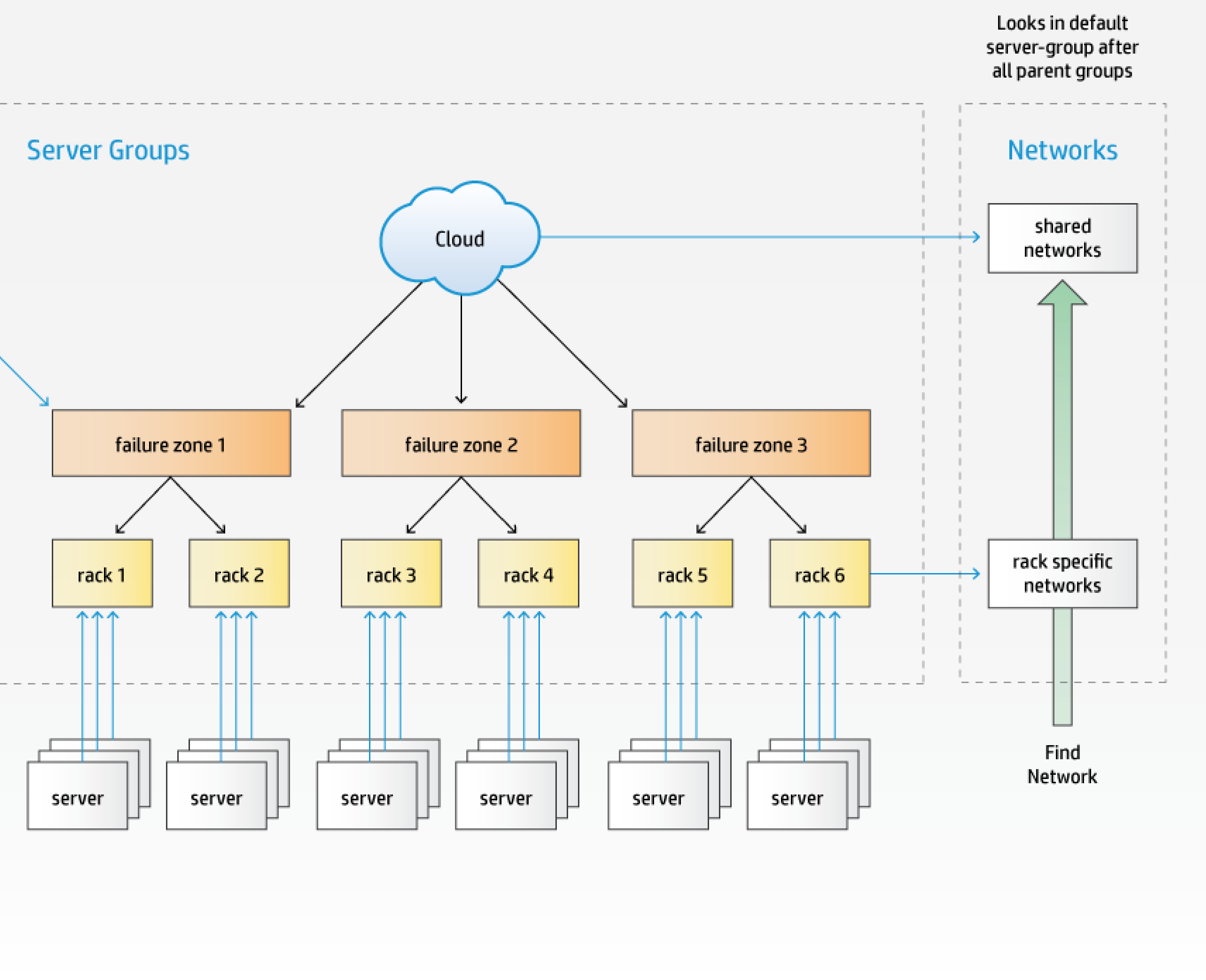
ysical servers in a number of racks or enclosures in a data center is common. Such racks generally provide a degree of physical isolation that allows for separate power and/or network connectivity.

ck model we support this configuration by allowing you to define a hierarchy of **server-groups**. Each **server** is associated with one **server-group**, normally at the bottom of the hierarchy.

onal part of the input model - if you don't define any then all **servers** and **networks** will be allocated as if they are part of the same **server-group**.

rver Groups and Failure Zones

ist of **server-groups** as the failure zones from which it wants to use servers. All servers in a **server-group** listed as a failure zone in the **control-plane** and any **server-groups** they contain are considered part of
ample shows how three levels of **server-groups** can be used to model a failure zone consisting of multiple racks, each of which in turn contains a number of **servers**.



on

he configuration processor will traverse down the hierarchy of **server-groups** listed as failure zones until it can find an available server with the required **server-role**. If the allocation policy is defined to be strict
ones. A **cluster** or **resource-group** can also independently specify the failure zones it wants to use if needed.

ver Groups and Networks

oup) and **networks** in a different **network-group** that span failure zones (the network used to provide floating IP addresses to virtual machines for example).

5.0: Networking

of **services** to specific **clusters** and **resources** we must also be able to define how the **services** connect to one or more **networks**.

be a single L3 network but more typically there are functional and physical layers of network separation that need to be expressed.

ion provides different networks for different types of traffic; for example, it is common practice in even small clouds to separate the External APIs that users will use to access the cloud and the external IP addre ore complex clouds it's common to also separate out virtual networking between virtual machines, block storage traffic, and volume traffic onto their own sets of networks. In the input model, this level of separa

red when there are separate L3 network segments providing the same type of traffic; for example, where each rack uses a different subnet. This level of separation is represented in the input model by the **networ**

work Groups

*networks in a specific **network-group**.*

*routes to other **networks**.*

*te the configuration for **services** via **network-tags***

he traffic separation model and all of the properties that are common to the set of L3 networks that carry each type of traffic. They define where services are attached to the network model and the routing within

ivity, all that has to be captured in the **network-groups** definition is the same service-component names that are used when defining **control-planes**. HPE Helion OpenStack also allows a default attachment to b licitly connected to another **network-group**. So, for example, to isolate Swift traffic, the swift-account, swift-container, and swift-object service components are attached to an "Object" **network-group** and all o **oup** via the default relationship.

vice connects, such as what port it uses, if it should be behind a load balancer, if and how it should be registered in Keystone, and so forth, are defined in the service definition files provided by HPE Helion Ope

ultiple networks, controlling the routing is a major consideration. In HPE Helion OpenStack, routing is controlled at the **network-group** level. First, all **networks** are configured to provide the route to any othe **rk-group** may be configured to provide the route any other **networks** in the same **network-group**; for example, if the internal APIs are in a dedicated **network-group** (a common configuration in a complex ne be segmented) then other **network-groups** may need to include a route to the internal API **network-group** so that services can access the internal API endpoints. Routes may also be required to define how to ac lt route.

OpenStack deployment, networks are configured to act as the default route for all traffic that was received via that network (so that response packets always return via the network the request came from).

nStack will configure the routing rules on the servers it deploys and will validate that the routes between services exist in the model, but ensuring that gateways can provide the required routes is the responsibility r provides information about the routes it is expecting to be configured.

f how the configuration processor validates routes, refer to *Network Route Validation*.

d Balancers

specific type of routing and are defined as a relationship between the virtual IP address (VIP) on a network in one **network group** and a set of service endpoints (which may be on **networks** in the same or a diff

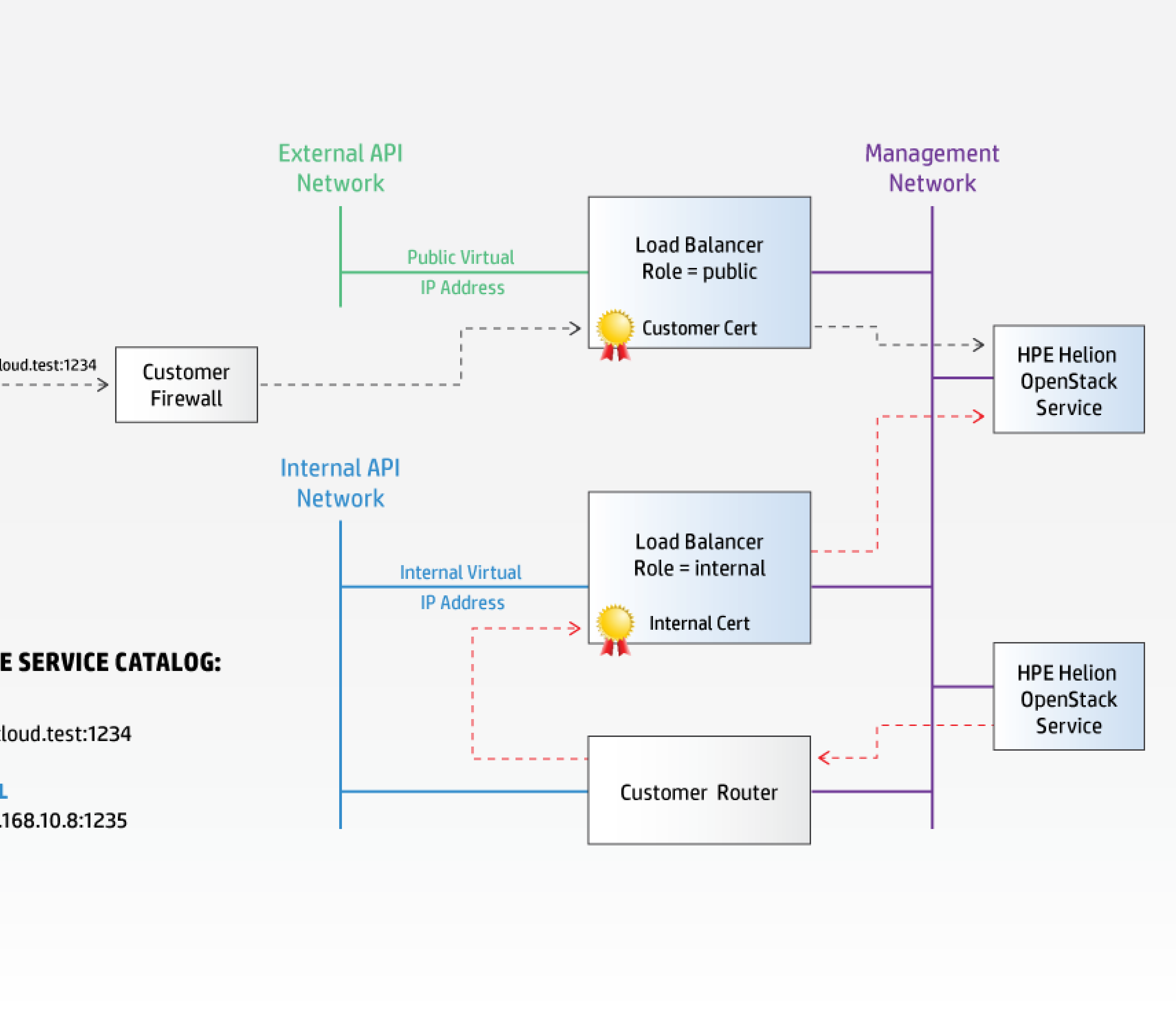
efined providing a virtual IP on a **network-group**, it follows that those **network-group**s can each only have one **network** associated to them.

on includes a list of **service-components** and endpoint roles it will provide a virtual IP for. This model allows service-specific **load-balancers** to be defined on different **network-groups**. A "default" value is us a virtual IP address and are not explicitly configured in another **load-balancer** configuration. The details of how the **load-balancer** should be configured for each service, such as which ports to use, how to che n OpenStack supplied service definition files.

aration of Public, Admin, and Internal Endpoints

r a **load-balancer** make it possible to configure separate **load-balancers** for public and internal access to services, and the configuration processor uses this information to both ensure the correct registrations in he correct endpoint. HPE Helion OpenStack services are configured to only connect to other services via internal virtual IP addresses and endpoints, allowing the name and security certificate of public endpoint ot be resolvable/accessible from the servers making up the cloud.

er defined in the input model will be allocated a separate virtual IP address even when the load-balancers are part of the same **network-group**. Because of the need to be able to separate both public and internal


External API Network
Management Network
Load Balancer Role = public
Public Virtual IP Address
Customer Cert
loud.test:1234
Customer Firewall
HPE Helion OpenStack Service
Internal API Network
Load Balancer Role = internal
Internal Virtual IP Address
Internal Cert
E SERVICE CATALOG:
loud.test:1234
L
.168.10.8:1235
HPE Helion OpenStack Service
Customer Router

on

work Tags

ork-group.

definitions. Each **network** defines the details of its VLAN, optional address details (CIDR, start and end address, gateway address), and which **network-group** it is a member of.

rface Model

nterface-model *that describes how its network interfaces are to be configured and used.*

d onto specific network interfaces via an **interface-model**, which describes the network devices that need to be created (bonds, ovs-bridges, etc) and their properties.

ke a template; it can define how some or all of the **network-groups** are to be mapped for a particular combination of physical NICs. However, it is the **service-components** on each server that determine which **n**
networks will be configured. This means that **interface-models** can be shared between different **server-roles**. For example, an API role and a database role may share an interface model even though they may
subset of the **network-groups**.

, physical ports are identified by a device name, which in turn is resolved to a physical port on a server basis via a **nic-mapping**. To allow different physical servers to share an **interface-model**, the **nic-mappin**

can also used to describe how network devices are to be configured for use with DPDK, SR-IOV, and PCI Passthrough.

C Mapping

an a single physical network port, a **nic-mapping** is required to unambiguously identify each port. Standard Linux mapping of ports to interface names at the time of initial discovery (e.g. eth0, eth1, eth2, ...) is n
ng of PCI bus address to interface name is instead.

d to specify the device type for interfaces that are to be used for SR-IOV or PCI Passthrough. Each HPE Helion OpenStack release includes the data for the supported device types.

ewall Configuration

r uses the details it has about which networks and ports **service-components** use to create a set of firewall rules for each server. The model allows additional user-defined rules on a per **network-group** basis.

5.0: Configuration Data

to provide settings which have to be applied in a specific context, or where the data needs to be verified against or merged with other values in the input model.

g a Neutron provider network to be used by Octavia, the network needs to be included in the routing configuration generated by the Configuration Processor.

® 5.0: Configuration Objects

5.0: Cloud Configuration

uration file, **cloudConfig.yml**, defines some global values for the HPE Helion OpenStack Cloud, as described in the table below.

e start of the control plane definition file.

```
cale-kvm-vsa

:
ix: helion
efix: -m
```

```
rue
pped packets
true

s:
 /var/audit
isabled
rvices:
ne
```

ey	Value Description
	An administrator-defined name for the cloud
	Provides control over some parts of the generated names (see *Name Generation*) Consists of two values: • host-prefix - default is to use the cloud name (above) • member-prefix - default is "-m"
	A list of external NTP servers your cloud has access to. If specified by name then the names need to be resolvable via the external DNS nameservers you specify in the next section. A server" component will be configured to use these external NTP servers.
	DNS configuration data that will be applied to all servers. See example configuration for a full list of values.
	SMTP client configuration data that will be applied to all servers. See example configurations for a full list of values.
)	Used to enable/disable the firewall feature and to enable/disable logging of dropped packets. The default is to have the firewall enabled.
	Used to enable/disable the production of audit data from services. The default is to have audit disabled for all services.

' 5.0: Control Plane

ne start of the control plane definition file.

```




rol-plane-1
ane-prefix: cp1
e: region1
nes:



ion-data:
N-CONFIG-CP1
A-CONFIG-CP1
vice-components:
```

```
r-prefix: c1
-role: CONTROLLER-ROLE
-count: 3
tion-policy: strict
e-components:
fecycle-manager
p-server
ift-ring-builder
sql
-cluster



compute
ce-prefix: comp
-role: COMPUTE-ROLE
tion-policy: any
unt: 0
e-components:
tp-client
ova-compute
ova-compute-kvm
eutron-l3-agent
```

Key	Value Description
	This name identifies the control plane. This value is used to persist server allocations (see *Persisted I* once servers have been allocated.
nal)	The control-plane-prefix is used as part of the hostname (see *Name Generation*). If not specified, the
	This name identifies the Keystone region within which services in the control plane will be registerec
nts (optional)	This lists a set of service components that run on all servers in the control plane (clusters and resourc
	A list of **server-group** names that servers for this control plane will be allocated from. If no failure-z not associated with a **server-group** will be used. (see *Server Groups and Failure Zones* for a descrip zones.)
al)	A list of configuration data settings to be used for services in this control plane (see *Configuration D*
	A list of clusters for this control plane (see *Clusters*).
	A list of resource groups for this control plane (see *Resources*).

sters

Key	Value Description
	Cluster and resource names must be unique within a control plane. This value is used to persist serve *Data*) and cannot be changed once servers have been allocated.
	The cluster prefix is used in the hostname (see *Name Generation*). If not supplied then the cluster nar
	This can either be a string (for a single role) or a list of roles. Only servers matching one of the specif

	control plane are also deployed.)
	Defines the number of servers to add to the cluster. The number of servers that can be supported in a cluster depends on the services it is running. For ex can only be deployed on clusters on 1 (non-HA) or 3 (HA) servers. Other services may support differ If min-count is specified, then at least that number of servers will be allocated to the cluster. If min-c to a value of 1. If max-count is specified, then the cluster will be limited to that number of servers. If max-count is n matching the required role and failure-zones will be allocated to the cluster. Specifying member-count is equivalent to specifying min-count and max-count with the same value.
	A list of **server-groups** that servers will be allocated from. If specified, it overrides the list of values If not specified, the control-plane value is used. (see *Server Groups and Failure Zones* for a descripti zones).
)	Defines how failure zones will be used when allocating servers. **strict**: Server allocations will be distributed across all specified failure zones. (if max-count is not a v of the number of zones, then some zones may provide one more server than other zones) **any**: Server allocations will be made from any combination of failure zones. The default allocation-policy for a cluster is *strict* .
al)	A list of configuration-data settings that will be applied to the services in this cluster. The values for with any values defined as part of the configuration-data list for the control-plane. If a value is specif value defined here takes precedence.

ources

Key	Value Description
	The name of this group of resources. Cluster names and resource-node names must be unique within clusters and resources cannot share names within a control-plane. This value is used to persist server allocations (see *Persisted Data*) and cannot be changed once serve
	The resource-prefix is used in the name generation. (see *Name Generation*)
	This can either be a string (for a single role) or a list of roles. Only servers matching one of the specif allocated to this resource group. (see *Server Roles* for a description of server roles).
	The list of **service-components** to be deployed on the servers in this resource group. (The common-s control plane are also deployed.)

	The number of servers that can be supported in a cluster depends on the services it is running. For ex can only be deployed on clusters on 1 (non-HA) or 3 (HA) servers. Other services may support diffe If min-count is specified, then at least that number of servers will be allocated to the cluster. If min-c to a value of 1. If max-count is specified, then the cluster will be limited to that number of servers. If max-count is n matching the required role and failure-zones will be allocated to the cluster. Specifying member-count is equivalent to specifying min-count and max-count with the same value.
	A list of **server-groups** that servers will be allocated from. If specified, it overrides the list of values If not specified, the control-plane value is used. (see *Server Groups and Failure Zones* for a descripti zones).
)	Defines how failure zones will be used when allocating servers. **strict**: Server allocations will be distributed across all specified failure zones. (if max-count is not a v of the number of zones, then some zones may provide one more server than other zones) **any**: Server allocations will be made from any combination of failure zones. The default allocation-policy for resources is *any*.
al)	A list of configuration-data settings that will be applied to the services in this cluster. The values for with any values defined as part of the configuration-data list for the control-plane. If a value is specif value defined here takes precedence.

d Balancer Definitions in Control Planes

enStack 5.0, a load-balancer may be defined within a control-plane object, and referenced by name from a network-groups object. The following example shows load balancer `extlb` defined in control-plane cp
group. See section Load balancers for a complete description of load balance attributes.

```
NAL-API
ers:




ers:
r: ip-cluster
xtlb
l-name:
ponents:
ault

lic
le: cp1-extlb-cert
```

5.0: Load Balancers

ned as part of a network-group object, or as part of a control-plane object. When a load-balancer is defined in a control-plane, it must be referenced by name only from the associated network-group object.

```
ers:
r: ip-cluster
xtlb
l-name:

ponents:
ault

lic
le: cp1-extlb-cert
```

Key	Value Description
	An administrator defined name for the load balancer. This name is used to make the association from
	The service component that implements the load balancer. Currently only `ip-cluster` (ha-proxy) will provide support for external load balancers.
	The list of endpoint roles that this load balancer provides (see below). Valid roles are `public`, `inte` separation of concerns, the role `public` cannot be combined with any other role. See Load Balancer provides endpoint separation.
	The list of service-components for which the load balancer provides a non-encrypted virtual IP addre
	The list of service-components for which the load balancer provides TLS-terminated virtual IP addre
	The name to be registered in Keystone for the publicURL. If not specified, the virtual IP address will value cannot be changed after the initial deployment.
	The name of the certificate file to be used for tls endpoints. If not specified, a file name will be constr `name>-<lb-name>-cert`, where `cp-name` is the control-plane name and `lb-name` is the load-

? 5.0: Servers

object is used to list the available servers for deploying the cloud.

s an input file to the operating system installation process, in which case some additional fields (identified below) will be necessary.

```
68.10.0
255.255.0


ler1
2.168.10.3
OLLER-ROLE
p: RACK1
: HP-DL360-4PORT
2:72:8d:ac:7c:6f
.168.9.3
d: password
```

```
.168.9.4
d: password
dmin
```

Key	Value Description
	An administrator-defined identifier for the server. IDs must be unique and are used to track server all
	The IP address is used by the configuration processor to install and configure the service components This IP address must be within the range of a **network** defined in this model. When the servers file is being used for operating system installation, this IP address will be assigned process, and the associated **network** must be an untagged VLAN.
	The value to use for the hostname of the server. If specified this will be used to set the hostname valu be reflected in systems such as Nova, Monasca, etc. If not specified the hostname will be derived bas and the network defined to provide hostnames.
	Identifies the **server-role** of the server. (see *Server Roles* for a description of server roles)
	Name of the **nic-mappings** entry to apply to this server. (see *NIC Mappings*)
	Identifies the **server-groups** entry that this server belongs to. (see *Server Groups*)
	Must be set to true is the server needs to be configured to boot from SAN storage. Default is False
	A list of network devices that will be used for accessing FCoE storage. This is only needed for device not devices such as Emulex which present as a FC device.
	A string of additional variables to be set when defining the server as a host in Ansible. For example,
	Needed when the servers file is being used for operating system installation. This identifies the MAC be used to network install the operating system.
	The name of the cobbler server profile to be used when the servers file is used for operating system i • hlinux-x86_64 (default) • rhel72-x86_64 • rhel72-x86_64-multipath **Important:** RHEL is only supported for KVM compute hosts. Note that you need to add a - value when using multipath with RHEL.
	Provides additional command line arguments to be passed to the booting network kernel. For exampl mode for the install to low resolution which can be useful for remote console users.
	Needed when the servers file is being used for operating system installation. This provides the IP add (e.g. IPMI, iLO) subsystem.
	Needed when the servers file is being used for operating system installation. This provides the user n (e.g. ipmi-ip, iLO) subsystem.
	Needed when the servers file is being used for operating system installation. This provides the user p management (e.g. ipmi-ip, iLO) subsystem.
	Needed when the servers file is being used for operating system installation. Additional options to pa

```
D
oups:




AL-API-NET
AL-VM-NET
NET
MENT-NET


roup for each failure zone


ups:



ups:



ups:



roup for each rack

1
2
3
```

Key	Value Description
	An administrator-defined name for the server group. The name is used to link server-groups together be used as failure zones in a **control-plane**. (see *Control Plane*)
	A list of server-group names that are nested below this group in the hierarchy. Each server group can server group (i.e. in a strict tree topology).
	A list of network names (see *Networks*). See *Server Groups and Networks* for a description of how n via server groups.

5.0: Server Roles

ion object is a list of the various server roles that you can use in your cloud. Each server role is linked to other configuration objects:

odels)
erface Models)

```
ROLLER-ROLE
model: CONTROLLER-INTERFACES
: CONTROLLER-DISKS

UTE-ROLE
model: COMPUTE-INTERFACES
: COMPUTE-DISKS
el: COMPUTE-MEMORY
 COMPUTE-CPU


ROLE
model: VSA-INTERFACES
: VSA-DISKS
```

Key	Value Description
	An administrator-defined name for the role.
	The name of the **interface-model** to be used for this server-role. Different server-roles can use the same interface-model.
	The name of the **disk-model** to use for this server-role. Different server-roles can use the same disk-model.
	The name of the **memory-model** to use for this server-role. Different server-roles can use the same memory-model.
	The name of the **cpu-model** to use for this server-role. Different server-roles can use the same cpu-model.

5.0: Disk Models

tion object is used to specify how the directly attached disks on the server should be configured. It can also identify which service or service component consumes the disk, e.g. Swift object server, and provide se

evices or as logical volumes and the disk model provides a configuration item for each.

been installed by the HPE Helion OpenStack installation process then the root disk will already have been set up as a volume-group with a single logical-volume. This logical-volume will have been created on a uration files as `/dev/sda_root`. This is due to the fact that different BIOS systems (UEFI, Legacy) will result in different partition numbers on the root disk.

```
KS
```

Key	Value Description
	The name of the disk-model that is referenced from one or more server-roles.
	A list of volume-groups to be configured (see below). There must be at least one volume-group descr
	A list of device-groups (see below)

ume Groups

uration object is used to define volume groups and their constituent logical volumes.

e not exact analogs of device-groups. A volume-group specifies a set of physical volumes used to make up a volume-group that is then subdivided into multiple logical volumes.

k operating system installation automatically creates a volume-group name "hlm-vg" on the first drive in the system. It creates a "root" logical volume there. The volume-group can be expanded by adding more p
create more logical-volumes on this volume-group to provide dedicated capacity for different services or file system mounts.

```
vg
olumes:
da_root

lumes:
root
35%
: ext4
 /

log
50%
 /var/log
: ext4
pts: -O large_file



omp
olumes:
db
lumes:
compute
95%
 /var/lib/nova
: ext4
pts: -O large_file
```

Key	Value Descriptions
	The name that will be assigned to the volume-group

	As installed by the HPE Helion OpenStack operating system install process, the volume group "hlm-v (sda_root) on the first disk. This can be expanded by adding additional disk(s). **Important:** Multipath storage should be listed as the corresponding `/dev/mapper/mpat`
	A list of logical volume devices to create from the above named volume group.
	The name to assign to the logical volume.
	The size, expressed as a percentage of the entire volume group capacity, to assign to the logical volu
	The file system type to create on the logical volume. If nonE specified, the volume is not formatted.
	Options, e.g. `-O large_file` to pass to the mkfs command.
	The `mode` changes the root file system mode bits, which can be either a symbolic representation or a bit patten for the new mode bits.
	Mount point for the file system.
nal, consumer dependent)	These will vary according to the service consuming the device group. The examples section provides services. Note, not all services support the use of logical volumes. VSA requires raw devices.

vice Groups

ration object provides the mechanism to make the whole of a physical disk available to a service.

```
a-data
:
vsa
 data

: /dev/sdc
a-cache
:
vsa
 adaptive-optimization

: /dev/sdb
```

Key	Value Descriptions
	An administrator-defined name for the device group.
	A list of named devices to be assigned to this group. There must be at least one device in the group. Multipath storage should be listed as the corresponding /dev/mapper/mpath<x>
	Identifies the name of one of the storage services (e.g. one of the following: Swift, Cinder, Ceph, VS disks in this device group.

the number of pages of a particular size to be configured at the server level or at the numa-node level.

uld configure :

of numa nodes 0 and 1

buted across all numa nodes)

ted across all numa nodes)

```
TE-MEMORY-NUMA
e-page-size: 2M

M
5
de: 0
M
5
de: 1
G
3
M
6
```

Key	Value Description
	The name of the memory-model that is referenced from one or more server-roles.
tional)	The default page size that will be used is specified when allocating huge pages. If not specified, the default is set by the operating system.
	A list of huge page definitions (see below).

ge Pages

Key	Value Description
	The page size in kilobytes, megabytes, or gigabytes specified as *n*X where: ***n*** is an integer greater than zero **X** is one of "K", "M" or "G"
	The number of pages of this size to create (must be greater than zero).
	If specified the pages will be created in the memory associated with this numa node. If not specified the pages are distributed across numa nodes by the operating system.

5.0: CPU Models

uration object describes how CPUs are assigned for use by service components such as Nova (for VMs) and Open vSwitch (for DPDK), and whether or not those CPUs are isolated from the general kernel SMP b

```
TE-CPU
:
nts:
a-compute-kvm

cessor-ids: 0-1,3,5-7
e: vm
nts:
nvswitch

cessor-ids: 4,12
late: False
e: eal
cessor-ids: 2,10
e: pmd
```

Key	Value Description
	An administrator-defined name for the cpu model.
	A list of CPU assignments (see *below*).

U Assignments

Key	Value Description
	A list of components to which the CPUs will be assigned.
	A list of CPU usage objects (see *below*).

U Usage

ey	Value Description
	A list of CPU IDs as seen by the operating system.
	A boolean value which indicates if the CPUs are to be isolated from the general kernel SMP balancing and scheduling algorithms. The specified processor IDs will be configured in th parameter. The default value is True.
	A role within the component for which the CPUs will be used.

mponents and Roles in the CPU Model

Component	Role	Description

	pmd	The specified processor IDs will be configured in the Open vSwitch pmd-cpu-mask option. Refer to the Open vSwit vswitchd.conf.db man page for details.

5.0: Interface Models

guration object describes how network interfaces are bonded and the mapping of network groups onto interfaces. Interface devices are identified by name and mapped to a particular physical port by the **nic-map**

```
s:
RFACE_SET_CONTROLLER
terfaces:
 BONDED_INTERFACE
e:
e: bond0
data:
vider: linux
ices:
 name: hed3
 name: hed4
ions:
ode: active-backup
iimon: 200
rimary: hed3
rk-groups:
EXTERNAL_API
EXTERNAL_VM
GUEST

 UNBONDED_INTERFACE
e:
me: hed0
rk-groups:
MGMT


faces:
 FCOE_DEVICES
es:
th7
th8


RFACE_SET_DPDK
terfaces:
 BONDED_DPDK_INTERFACE
e:
e: bond0
data:
vider: openvswitch
ices:
 name: dpdk0
```

```
me: dpdk2
rk-groups:
PHYSNET2
es:
s:
me: dpdk0
me: dpdk1
me: dpdk2
iver: igb_uio
ents:
envswitch
tions:
me: socket-mem
lue: 1024,0
me: n
lue: 2
ent-options:
me: n-dpdk-rxqs
lue: 64
```

Key	Value Description
	An administrator-defined name for the interface model.
	A list of network interface definitions.
	A list of network interfaces that will be used for Fibre Channel over Ethernet (FCoE). This is only ne a native FCoE device, not cards such as Emulex which present FCoE as a FC device. **Important:** The devices must be "raw" device names, not names controlled via a nic-mappi
	A list of DPDK device definitions.

work-interfaces

figuration object has the following attributes:

Key	Value Description
	An administrator-defined name for the interface
	A dictionary containing the network device name (as seen on the associated server) and associated pr *device* for details).
	Used to define a bond. See *Bonding* for details.
f forced-network-groups is defined)	A list of one or more **network-groups** (see *Network Groups*) containing **networks** (see *Networks*) th interface. Networks in these groups will only be configured if there is at least one **service-componen** the list of component-endpoints defined in the **network-group**.
tional if network-groups is defined)	A list of one or more **network-groups** (see *Network Groups*) containing **networks** (see *Networks*) th

Key	Value Description
	When configuring a bond, this is used as the bond device name - the names of the devices to be bond section. If the interface is not bonded, this must be the name of the device specified by the nic-mapping (see N
	Indicates that the interface is to be used for SR-IOV. The value is the number of virtual functions to b specified by the nic-mapping must have a valid nice-device-type. vf-count cannot be specified on bonded interfaces Interfaces used for SR-IOV must be associated with a network with `tagged-vlan: false`.
	Only valid when vf-count is specified. If set to true then the interface is to be used for virtual functio will not be used. The default value is False.
	If set to true then the interface is used for PCI passthrough. The default value is False.

nding

used to configure a bond device, and consists of the following attributes:

Key	Value Descriptions
	Identifies the software used to instantiate the bond device. The supported values are • **linux** to use the Linux bonding driver. • **openvswitch** to use Open vSwitch bonding.
	A dictionary containing network device names used to form the bond. The device names must be the **nic-mapping** (see *NIC mapping*.
	A dictionary containing bond configuration options. The *linux* provider options are described in the *E for the "linux" Provider* section. The *openvswitch* provider options are described in the section *Bond "openvswitch" provider*.

nd configuration options for the "linux" provider

upports a large number of parameters that control the operation of the bond, as described in the *Linux Ethernet Bonding Driver HOWTO* document. The parameter names and values may be specified as key-valu

elion OpenStack examples are:

	• balance-rr - Transmit packets in sequential order from the first available slave through the last. • active-backup - Only one slave in the bond is active. A different slave becomes active if, and only • balance-xor - Transmit based on the selected transmit hash policy. • broadcast - Transmits everything on all slave interfaces. • 802.3ad - IEEE 802.3ad Dynamic link aggregation. • balance-tlb - Adaptive transmit load balancing: channel bonding that does not require any special • balance-alb - Adaptive load balancing: includes balance-tlb plus receive load balancing (rlb) for I any special switch support.
	Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state link failures. Accepts values in milliseconds.
	The device to use as the primary when the mode is one of the possible values below: • active-backup • balance-tlb • balance-alb

nd Data Options for the "openvswitch" Provider

ions for Open vSwitch bonds are:

Key	Value Descriptions
	Specifies the bonding mode. Possible values include: • active-backup • balance#tcp • balance#slb Refer to the Open vSwitch `ovs-vswitchd.conf.db` man page for details.

e-interfaces

ration object has the following attributes:

Key	Value Description
	An administrator-defined name for the group of FCOE interfaces
	A list of network devices that will be configured for FCOE Entries in this must be the name of a device specified by the nic-mapping (see *NIC Mappings*).

k-devices

ation object has the following attributes:

Key	Value Descriptions
	A list of network devices to be configured for DPDK. See *dpdk-devices devices*.

	Note that the cpu-model should be used to specify the processor IDs to be used by EAL for this comp option will be set automatically based on the information in the cpu-model, and so should not be spec *CPU Models*.
	A list of key-value pairs that may be used to set component-specific configuration options.

k-devices devices

bject within dpdk-devices has the following attributes:

Key	Value Descriptions
	The name of a network device to be used with DPDK. The device names must be the logical-name sp *NIC Mappings*).
	Defines the userspace I/O driver to be used for network devices where the native device driver does r capabilities. The default value is `igb_uio`.

DK component-options for the openvswitch component

upported for use with the openvswitch component:

Name	Value Descriptions
	Number of rx queues for each DPDK interface. Refer to the Open vSwitch documentation and the `ov` page for details.

ould be used to define the cpu affinity of the Open vSwitch PMD (Poll Mode Driver) threads. The Open vSwitch `pmd-cpu-mask` option will be set automatically based on the information in the cpu-model. Se

5.0: NIC Mappings

ration object is used to ensure that the network device name used by the operating system always maps to the same physical device. A **nic-mapping** is associated to a **server** in the server definition file. (see *Serv* s with any other devices configured during the operating system install as well as any interfaces that are not being managed by HPE Helion OpenStack, ensuring that all devices on a baremetal machine are speci ustrates:

```
360-4PORT
rts:
-name: hed1
imple-port
ress: "0000:07:00.0"

-name: hed2
imple-port
ress: "0000:08:00.0"
ice-type: '8086:10fb'
```

```
-name: hed4
ulti-port
ress: "0000:09:00.0"
tributes:
t-num: 1
```

ings list has the following attributes:

Key	Value Description
	An administrator-defined name for the mapping. This name may be used in a server definition (see *S* that server.
	A list containing device name to address mapping information.

ports list has the following attributes:

Key	Value Description
	The network device name that will be associated with the device at the specified *bus-address*. The lo used as a device name in network interface model definitions. (see *Interface Models*)
	The type of port. HPE Helion OpenStack 5.0 supports "simple-port" and "multi-port". Use "simple-p bus-address. Use "multi-port" if your hardware requires a "port-num" attribute to identify a single po examples of such a device is: • Mellanox Technologies MT26438 [ConnectX VPI PCIe 2.0 5GT/s - IB QDR / 10GigE Virtualiza
	PCI bus address of the port. Enclose the bus address in quotation marks so yaml does not misinterpre characters. See *Pre-Install Checklist - Information for nic_mappings.yml* for details on how to determ
type is `multi-port`)	Provides a list of attributes for the physical port. The current implementation supports only one attrib devices share a bus-address. Use the "port-num" attribute to identify which physical port on the multi *Install Checklist - Information for nic_mappings.yml* for details on how to determine this value.
	Specifies the PCI vendor ID and device ID of the port in the format of <vendor_id>:<device_ 8086:10fbs.

5.0: Network Groups

overall network topology, including where service-components connect, what load balancers are to be deployed, which connections use TLS, and network routing. They also provide the data needed to map Neu

```
RNAL-API
uffix: extapi

cers:
er: ip-cluster
extlb
```

```
ERNAL-VM

on.l3_agent.external_network_bridge

ST
suffix: guest

on.networks.vxlan

AGEMENT
suffix: mgmt
 true

-endpoints:
lt


lt

ncers:
der: ip-cluster
 lb
nents:
efault
:
nternal
dmin


on.networks.vlan:
vider-physical-network: physnet1
```

Key	Value Description
	An administrator-defined name for the network group. The name is used to make references from oth
ional)	The list of **service-components** that will bind to or need direct access to networks in this network-gro
	If set to true, the name of the address associated with a network in this group will be used to set the h **Important:** `hostname` **must** be set to `true` for one, and only one, of your network groups
)	If supplied, this string will be used in the name generation (see *Name Generation*). If not specified, th will be used.
	A list of load balancers to be configured on networks in this network-group. Because load balances n network group that contains a load balancer can only have one network associated with it. For clouds consisting of a single control plane, a load balancer may be fully defined within a `netwo` balancer definitions in network groups. Starting in HPE Helion OpenStack 5.0, a load balancer may be defined within a `control-plane` o from a `network-group` object. See *Load balancer definitions* in control planes.

	define the default route. A network group with no services attached to it can be used to define routes to external networks. The name of a Neutron provide network defined via configuration-data (see *here*) can also be include
	A list of network tags. Tags provide the linkage between the physical network configuration and the I Starting in HPE Helion OpenStack 5.0, network tags may be defined as part of a Neutron `configur` than as part of a `network-group` object (see section *Configuration Data*).
	Specifies the MTU value required for networks in this network group If not specified a default value See notes *here* on how MTU settings are applied to interfaces when there are multiple tagged networl

has the following attributes:

Key	Value Description
	An administrator-defined name for the load balancer.
	The service component that implements the load balancer. Currently only "ip-cluster" (ha-proxy) is s provide support for external load balancers.
	The list of endpoint roles that this load balancer provides (see below). Valid roles are "public", "inter separation of concerns, the role "public" cannot be combined with any other role. See *Load Balancer* provides endpoint separation.
	The list of **service-components** for which the load balancer provides a non-encrypted virtual IP addr
	The list of **service-components** for which the load balancer provides TLS-terminated virtual IP addr OpenStack 3.0, TLS is now supported for internal as well as public endpoints.
	The name to be registered in Keystone for the publicURL. If not specified, the virtual IP address will value cannot be changed after the initial deployment.
	The name of the certificate file to be used for TLS endpoints.

d Balancer Definitions in Network Groups

ngle control-plane, a `load-balancer` may be fully defined within a `network-groups` object as shown in the examples above. See section *Load Balancers* for a complete description of load balancer attrib

enStack 5.0, a `load-balancer` may be defined within a `control-plane` object in which case the network-group provides just a list of load balancer names as shown below. See section *Load Balancer* defi

```



RNAL-API
uffix: extapi

cers:


```

me can be used in multiple control-planes to make the above list simpler.

work Tags

Tag	Value Description
	This tag causes Neutron to be configured to use VxLAN as the underlay for tenant networks. The ass the VxLAN traffic.
ional)	Used to specify the VxLAN identifier range in the format “<min-id>:<max-id>”. The default range i range in quotation marks. Multiple ranges can be specified as a comma-separated list.

D range:

```
works.vxlan
```

ed ID range:

```
works.vxlan:
lan-id-range: “1:20000”
```

er-defined ID range:

```
works.vxlan:
lan-id-range: “1:2000,3000:4000,5000:6000”
```

Tag	Value Description
	This tag causes Neutron to be configured for provider VLAN networks, and optionally to use VLAN networks. The associated network group will carry the VLAN traffic. This tag can be specified on mu NOTE: this tag does not cause any Neutron networks to be created, that must be done in Neutron afte
:	The provider network name. This is the name to be used in the Neutron API for the *provider:physica* objects.
onal)	This attribute causes Neutron to use VLAN for tenant networks; omit this attribute if you are using p the VLAN ID range for tenant networks, in the format “<min-id>:<max-id>”. Enclose the range in q can be specified as a comma-separated list.

lan only (may be used with tenant VxLAN):

```
works.vlan:
physical-network: physnet1
```

l provider VLAN:

Tag	Value Description
	This tag causes Neutron to be configured for provider flat networks. The associated network group w be specified on multiple network groups. NOTE: this tag does not cause any Neutron networks to be created, that must be done in Neutron afte
:	The provider network name. This is the name to be used in the Neutron API for the *provider:physica* objects. When specified on multiple network groups, the name must be unique for each network grou

lat network:

```
works.flat:
physical-network: flatnet1
```

_network_bridge

Tag	Value Description
network_bridge	This tag causes the Neutron L3 Agent to be configured to use the associated network group as the Ne floating IP addresses. A CIDR **should not** be defined for the associated physical network, as that wil network to be configured in the hypervisor. When this tag is used, provider networks cannot be used NOTE: this tag does not cause a Neutron external networks to be created, that must be done in Neutro

agent.external_network_bridge:

```
agent.external_network_bridge
```

U (Maximum Transmission Unit)

onally specify an MTU for its networks to use. Because a network-interface in the interface-model may have a mix of one untagged-vlan network group and one or more tagged-vlan network groups, there are so twork group.

ts of untagged-vlan network(s) then its specified MTU must be greater than or equal to the MTU of any tagged-vlan network groups which are co-located on the same network-interface.

work group with untagged VLANs, NET-GROUP-1, which is going to share (via a Network Interface definition) a device (eth0) with two network groups with tagged VLANs: NET-GROUP-2 (ID=201, MTU=

e an MTU which is large enough to accommodate the VLAN in NET-GROUP-3. Since NET-GROUP-1 has untagged VLANS it will also be using this device and so it must also have an MTU of 9000, which re

```
 <------ this MTU comes from NET-GROUP-1

201@eth0 (1550)
301@eth0 (9000)
```

```
   <------ because of NET-GROUP-3

01@bond0 (3000)
01@bond0 (1550)
01@bond0 (9000)
```

5.0: Networks

sents a physical L3 network used by the cloud infrastructure. Note that these are different from the network definitions that are created/configured in Neutron, although some of the networks may be used by Neut

```
EXTERNAL_VM
2
n: true
oup: EXTERNAL_VM

GUEST
3
n: true
.1.0/24
: 10.1.1.1
oup: GUEST

MGMT
0
n: false
.1.0/24

0-10.2.1.20
4
0-10.2.1.36
: 10.2.1.1
oup: MGMT
```

Key	Value Description
	The name of this network. The network *name* may be used in a server-group definition (see *Server G* network from within a network-group to be associated with a set of servers.
	The name of the associated network group.
	The IEEE 802.1Q VLAN Identifier, a value in the range 1 through 4094. A *vlanid* must be specified
	May be set to “true” or “false”. If true, packets for this network carry the *vlanid* in the packet header; VLAN-tagged frames in IEEE 802.1Q.
	The IP subnet associated with this network.

	default value is the first host address within the CIDR (e.g. the .1 address). The `addresses` parameter provides more flexibility than the `start-address` and `end-addre` preferred means of specifying this data.
eprecated)	An IP address within the *CIDR* which will be used as the start of the range of IP addresses from whic allocated. The default value is the first host address within the *CIDR* (e.g. the .1 address). **Important:** This parameter is deprecated in favor of the new `addresses` parameter. This future release.
precated)	An IP address within the *CIDR* which will be used as the end of the range of IP addresses from which allocated. The default value is the last host address within the *CIDR* (e.g. the .254 address of a /24). **Important:** This parameter is deprecated in favor of the new `addresses` parameter. This future release.
	The IP address of the gateway for this network. Gateway addresses must be specified if the associate routes.

5.0: Firewall Rules

r will automatically generate "allow" firewall rules for each server based on the services deployed and block all other ports. The firewall rules in the input model allow the customer to define additional rules for

s are applied after all rules generated by the Configuration Processor.

```
oups:
NT

-API

P echo request (ping)
low
p-prefix:  0.0.0.0/0
ype
ge-min: 8
ode
ge-max: 0
: icmp
```

Key	Value Description
	An administrator-defined name for the group of rules.
	A list of **network-group** names that the rules apply to. A value of "all" matches all network-groups.
	A list of rules. Rules are applied in the order in which they appear in the list, apart from the control p (see above). The order between sets of rules is indeterminate.

	Must "allow"
	Range of remote addresses in CIDR format that this rule applies to.
	Defines the range of ports covered by the rule. Note that if the protocol is "icmp" then port-range-mi range-max is the ICMP code.
	Must be one of "tcp", "udp", or "icmp".

5.0: Configuration Data

values to be passed into the model to be used in the context of a specific control plane or cluster. The content and format of the data is service specific.

```
ata:
RON-CONFIG-CP1



rovider_networks:
CTAVIA-MGMT-NET
r:
work_type: vlan
sical_network: physnet1
mentation_id: 106
72.30.1.0/24
way:  True
dhcp: True
ion_pools:
rt: 172.30.1.10
: 172.30.1.250
utes:
te to MANAGEMENT-NET-1
tination: 192.168.245.0/24
thop:  172.30.1.1

xternal_networks:
xt-net
72.31.0.0/24
: 172.31.0.1
r:
work_type: vlan
sical_network: physnet1
mentation_id: 107
ion_pools:
rt: 172.31.0.2
: 172.31.0.254

s:
-group: MANAGEMENT
```

Key	Value Description
	An administrator-defined name for the set of configuration data.
	A list of services that the data applies to. Note that these are service names (e.g. `neutron`, `octavi` names (`neutron-server`, `octavia-api`, etc).
	A service specific data structure (see below).
eutron-only)	A list of network tags. Tags provide the linkage between the physical network configuration and the Starting in HPE Helion OpenStack 5.0, network tags may be defined as part of a Neutron `configu` than as part of a `network-group` object.

utron network-tags

Key	Value Description
	The name of the network-group with which the tags are associated.
	A list of network tags. Tags provide the linkage between the physical network configuration and the See section Network Tags.

utron Configuration Data

Key	Value Description
s	A list of provider networks that will be created in Neutron.
	A list of external networks that will be created in Neutron. These networks will have the "router:exte

tron-provider-networks

Key	Value Description
	The name for this network in Neutron. This name must be distinct from the names of any Network Groups in the model to enable it to be inc network group.
	Details of network to be created • network_type • physical_network • segmentation_id These values are passed as `--provider:` options to the Neutron `net-create` command
	The CIDR to use for the network. This is passed to the Neutron `subnet-create` command.
	A boolean value that specifies if the network can be shared.

	These values are passed to the Neutron `subnet-create` command.
	A list of routes to be defined for the network. Each route consists of a `destination` in cidr format These values are passed to the Neutron `subnet-create` command.
	A gateway address for the network. This value is passed to the Neutron `subnet-create` command.
	A Boolean value indicating that the gateway should not be distributed on this network. This is translated into the `no-gateway` option to the Neutron `subnet-create` command
	A Boolean value indicating that DHCP should be enabled. The default if not specified is to not enabl This value is passed to the Neutron `subnet-create` command.

tron-external-networks

Key	Value Description
	The name for this network in Neutron. This name must be distinct from the names of any Network Groups in the model to enable it to be inc network group.
	The provider attributes are specified when using Neutron provider networks as external networks. Pro specified when the external network is configured with the `neutron.l3_agent.external_ne` Standard provider network attributes may be specified: • network_type • physical_network • segmentation_id These values are passed as `--provider:` options to the Neutron `net-create` command
	The CIDR to use for the network. This is passed to the Neutron `subnet-create` command.
)	A list of start and end address pairs that limit the set of IP addresses that can be allocated for this netw These values are passed to the Neutron `subnet-create` command.
	A gateway address for the network. This value is passed to the Neutron `subnet-create` command.

avia Configuration Data

```
rk_name: OCTAVIA-MGMT-NET
```

Key	Value Description
	The name of the Neutron provider network that Octavia will use for management access to load balar

nic Configuration Data

```
ata:
IC-CONFIG-CP1



network: guest-network
de_cleaning: true
eview: false

anager_url:
sername:
ncrypted_password:
llow_insecure_connections:
t_file:
ent_drivers: true
```

on configuring Ironic for details of the above attributes.

ft Configuration Data

```
ata:
ONFIG-CP1



ne_rings:
es:

r-groups:
Z1

r-groups:
Z2

r-groups:
```

```
lica-count: 3

 container
ay-name: Container Ring
art-hours: 16
tion-power: 12
cation-policy:
lica-count: 3

 object-0
ay-name: General
lt: yes
art-hours: 16
tion-power: 12
cation-policy:
lica-count: 3
```

on *HPE Helion OpenStack 5.0: Understanding Swift Ring Specifications* for details of the above attributes.

5.0: Pass Through

nitions, certain configuration values can be assigned and used.

```
ue



eck: false
_cluster: Cluster1
_id: BC9DED4E-1639-481D-B190-2B54A2BF5674
_ip: 10.1.200.41
_port: 443
_username: administrator@vsphere.local
c415b541ca9ecf9608b35b32261e6c0bf275a
```

Key	Value Description
	These values will be used at the cloud level.
	These values will be assigned to a specific server(s) using the server-id.

® 5.0: Other Topics

	Virtual Machine Provisioning	nova	nova-api nova-compute nova-compute-hyperv nova-compute-ironic nova-compute-kvm nova-conductor nova-console-auth nova-esx-compute-prox nova-metadata nova-novncproxy nova-scheduler nova-scheduler-ironic
	Bare Metal Provisioning	ironic	ironic-api ironic-conductor
	ESX Integration	eon	eon-api eon-conductor
	Networking	neutron	infoblox-ipam-agent neutron-dhcp-agent neutron-l2gateway-age neutron-l3-agent neutron-lbaas-agent neutron-lbaasv2-agent neutron-metadata-agen neutron-ml2-plugin neutron-openvswitch-a neutron-ovsvapp-agent neutron-server neutron-sriov-nic-age neutron-vpn-agent
	Network Load Balancer	octavia	octavia-api octavia-health-manage
	Domain Name Service (DNS)	designate	designate-api designate-central designate-mdns designate-mdns-extern designate-pool-manage designate-zone-manage
	Ceph Storage	ceph	ceph-monitor ceph-osd

generated from data taken from various parts of the input model as described in the following sections.

s in a cluster have the following form:

```
plane>-<cluster><member-prefix><member_id>-<network>
```

core-m1-mgmt

Name	Description
	Comes from the hostname-data section of the **cloud** object (see *Cloud*)
	is the **control-plane** prefix or name (see *Control Plane*)
	is the **cluster-prefix** name (see *Clusters*)
	comes from the hostname-data section of the **cloud** object (see *Cloud*)
	is the ordinal within the cluster, generated by the configuration processor as servers are allocated to t
	comes from the **hostname-suffix** of the network group to which the network belongs (see *NIC Mapp*

s in a resource group have the following form:

```
plane>-<resource-prefix><member_id>-<network>
```

comp0001-mgmt

Name	Description
	Comes from the hostname-data section of the **cloud** object (see *Cloud*).
	is the **control-plane** prefix or name (see *Control Plane*).
	is the **resource-prefix** value name (see *Resources*).
	is the ordinal within the cluster, generated by the configuration processor as servers are allocated to t zeroes to four digits.
	comes from the **hostname-suffix** of the network group to which the network belongs to (see *NIC Ma*

5.0: Persisted Data

r makes allocation decisions on servers and IP addresses which it needs to remember between successive runs so that if new servers are added to the input model they don't disrupt the previously deployed alloca

ltiple iterations of the input model before deployment HPE Helion OpenStack will only persist data when the administrator confirms that they are about to deploy the results via the "ready-deployment" operation mple:

ed your HPE Helion OpenStack deployment with servers A, B, and C and you want to add two new compute nodes by adding servers D and E to the input model.

input model and re-run the configuration processor it will read the persisted data for A, B, and C and allocate D and E as new servers. The configuration processor now has allocation data for A, B, C, D, and E -- n git) until we get confirmation that the configuration processor has done what you intended and you are ready to deploy the revised configuration.

ations

sted by the administrator-defined server ID (see *Servers*), and include the control plane, cluster/resource name, and ordinal within the cluster or resource group.

the configuration processor persists server allocations even when the server ID no longer exists in the input model -- for example, if a server was removed accidentally and the configuration processor allocated a ult to recover from that situation.

strates the behavior:

our servers with IDs of A, B, C, and D that can all be used in a resource group with `min-size=0` and `max-size=3`. At the end of this deployment they persisted state is as follows:

	Control Plane	Resource Group	Ordinal	State	
	ccp	compute	1	Allocated	mycloud-c
	ccp	compute	2	Allocated	mycloud-c
	ccp	compute	3	Allocated	mycloud-c
				Available	

as not been allocated because the group is at its max size, and there are no other groups that required this server)

n the input model and the configuration processor is re-run, the state is changed to:

	Control Plane	Resource Group	Ordinal	State	
	ccp	compute	1	Allocated	mycloud-c
	ccp	compute	2	Deleted	
	ccp	compute	3	Allocated	mycloud-c
	ccp	compute	4	Allocated	mycloud-c

server B are still retained, but the configuration processor will not generate any deployment data for this server. Server D has been added to the group to meet the minimum size requirement but has been given a addresses than were given to server B.

to the input model the resulting state will be:

	Control Plane	Resource Group	Ordinal	State	
	ccp	compute	1	Allocated	mycloud-c
	ccp	compute	2	Deleted	
	ccp	compute	3	Allocated	mycloud-c
	ccp	compute	4	Allocated	mycloud-c

r will issue a warning that server B cannot be returned to the compute group because it would exceed the max-size constraint. However, because the configuration processor knows that server B is associated wit use it, since that might lead to data loss on that server.

group was increased, then server B would be allocated back to the group, with its previous name and addresses (`mycloud-cp1-compute0002`).

processor relies on the server ID to identify a physical server. If the ID value of a server is changed the configuration processor will treat it as a new server. Conversely, if a different physical server is added wit will assume that it is the original server being returned to the model.

of persisted data for servers that are no longer in the input model by running the configuration processor with the `remove_deleted_servers` option, like below:

ddress allocations that are no longer used in the input model by running the configuration processor with the `free_unused_addresses` option, like below:

```
nsible
-i hosts/localhost config-processor-run.yml -e free_unused_addresses="y"
```

5.0: Server Allocation

r allocates servers to a cluster or resource group in the following sequence:

rsisted with a state of "allocated" are first returned to the **cluster** or **resource group**. Such servers are always allocated even if this contradicts the cluster size, failure-zones, or list of server roles since it is assum

ce group** is still below its minimum size, then any **servers** that are persisted with a state of "deleted", but where the server is now listed in the input model (i.e. the server was removed but is now back), are addec nd **server-role** criteria. If they do not meet the criteria then a warning is given and the **server** remains in a deleted state (i.e. it is still not allocated to any other cluster or group). These **servers** are not part of the cts before they can be redeployed.

ce group** is still below its minimum size, the configuration processor will allocate additional **servers** that meet the **failure-zone** and **server-role** criteria. If the allocation policy is set to "strict" then the failure zo p are not considered until an equal number of servers has been allocated from each zone.

5.0: Server Network Selection

cessor has allocated a **server** to a **cluster** or **resource group** it uses the information in the associated **interface-model** to determine which **networks** need to be configured. It does this by:

components that are to run on the server (from the **control-plane** definition)

etwork-group** each of those components is attached to (from the **network-groups** definition)

are any **network-tags** related to a **service-component** running on this server, and if so, adding those **network-groups** to the list (also from the **network-groups** definition)

are any **network-groups** that the **interface-model** says should be forced onto the server

er-group** hierarchy (as described in *Server Groups and Networks*) to find a **network** in each of the **network-groups** it needs to attach to

lable to a server, either because the **interface-model** doesn't include the required **network-group**, or there is no **network** from that group in the appropriate part of the **server-groups** hierarchy, then the configur

r will also generate an error if the **server** address does not match any of the networks it will be connected to.

5.0: Network Route Validation

cessor has allocated all of the required **servers** and matched them to the appropriate **networks**, it validates that all **service-components** have the required network routes to other **service-components**.

a in the services section of the input model which provides details of which **service-components** need to connect to each other. This data is not configurable by the administrator; however, it is provided as part c

uration processor looks at the list of **service-components** it runs and determines the network addresses of every other **service-component** it needs to connect to (depending on the service, this might be a virtual ice).

network that this **server** is connected to, then there is no routing required. If the target address is on a different **network**, then the Configuration Processor looks at each **network** the server is connected to and lo **oup**. If the **network-group** provides a route to the **network-group** of the target address, then that route is considered valid.

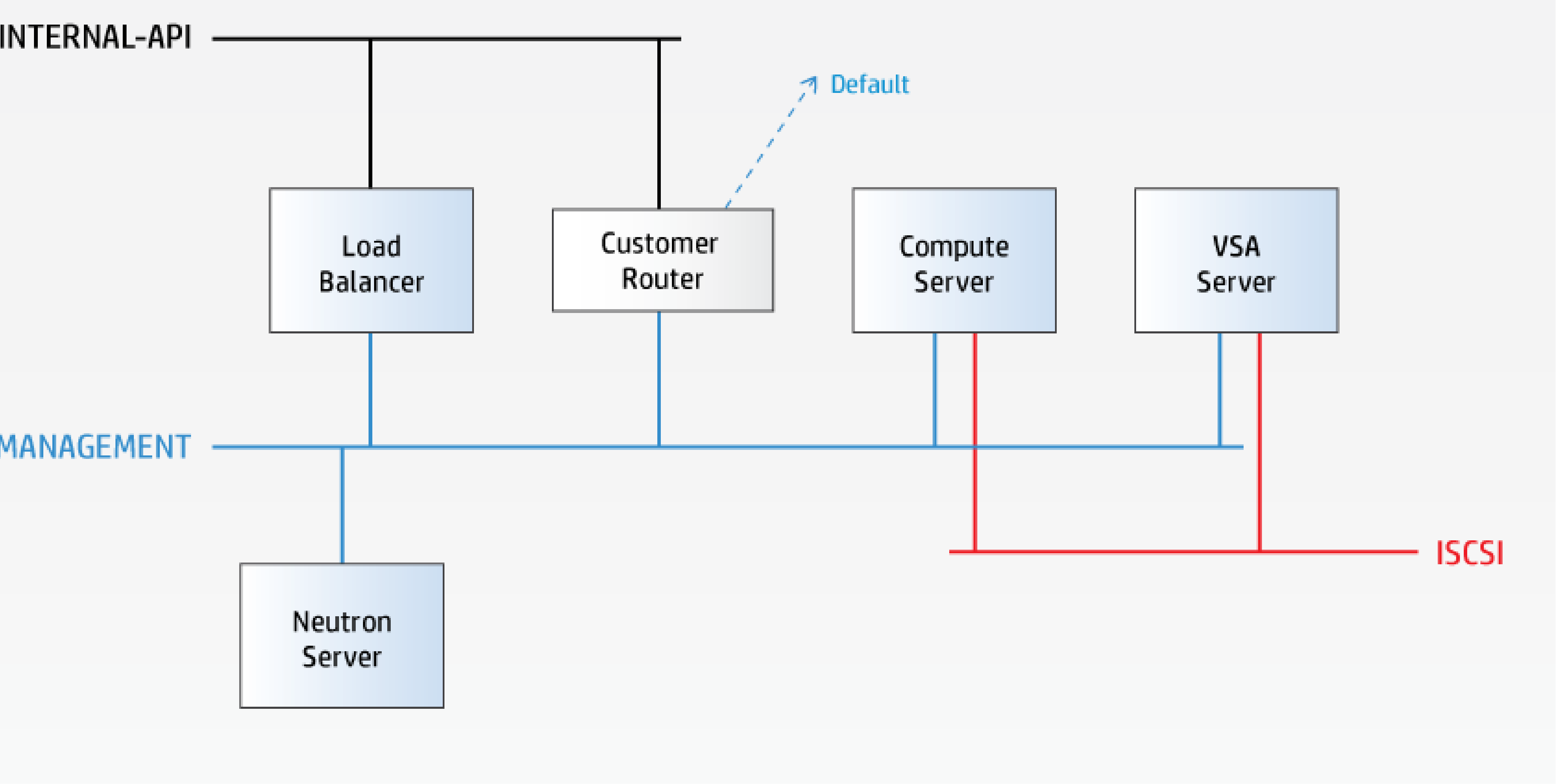
network-group are always considered as routed to each other; **networks** from different **network-groups** must have an explicit entry in the `routes` stanza of the **network-group** definition. Routes to a named t" route.

routes which are using the "default" route since it is possible that the user did not intend to route this traffic. Such warning can be removed by adding the appropriate **network-group** to the list of routes.

r provides details of all routes between networks that it is expecting to be configured in the `info/route_info.yml` file.

outing is defined in the input model, consider the following example:

ured to run nova-compute which requires access to the Neutron API servers and the VSA block storage service. The Neutron API servers have a virtual IP address provided by a load balancer in the INTERNAL



on

n the **network-groups** are:

```
L-API
ix: intapi

s:
: ip-cluster

ts:
ult

rnal
```

```
ult


RNAL-API
ult

CSI
-suffix: iscsi

t-endpoints:
```

or the compute server looks like this:

```

CE_SET_COMPUTE
faces:
D0

bond0
:
s:
e: active-backup
mon: 200
mary: hed5
er: linux
s:
ame: hed4
ame: hed5
roups:
EMENT
```

from nova-compute to the Neutron API, the configuration processor will detect that the target address is on a network in the INTERNAL-API network group, and that the MANAGEMENT network (which is co
work, and thus considers this route valid.

from nova-compute to VSA, the configuration processor will detect that the target address is on a network in the ISCSI network group. However, because there is no service component on the compute server co
group definition) the ISCSI network will not have been configured on the compute server (see *Server Network Selection*. The configuration processor will detect that the MANAGEMENT network-group provide
(it is, of course, valid to route ISCSI traffic); however, because this is using the default route, a warning will be issued:

```
or-2.0       WRN: Default routing used between networks
works are using a 'default' route rule. To remove this warning
licit route in the source network group or force the network to
erface model used by the servers.
RACK1 to ISCSI-NET
mp0001
RACK 2 to ISCSI-NET
mp0002
RACK 3 to SCSI-NET
mp0003
```

u can either add ISCSI to the list of routes in the MANAGEMENT network group (routed ISCSI traffic is still a valid configuration) or force the compute server to attach to the ISCSI network-group by adding i
is:

```
ns:
de: active-backup
imon: 200
imary: hed5
der: linux
es:
name: hed4
name: hed5
-groups:
NAGEMENT
network-groups:
CSI
```

ISCSI network group forced, the configuration processor will attach the compute server to a network in that group and validate the route as either being direct or between networks in the same network-group.

`fo.yml` file will include entries such as the following, showing the routes that are still expected to be configured between networks in the MANAGEMENT network group and the INTERNAL-API network gro

```
RACK1:
-NET:
false

ompute:
tron-server:
elion-ccp-comp0001
-RACK2:
-NET:
false

mpute:
on-server:
ion-ccp-comp0003
```

5.0: Configuring Neutron Provider VLANs

re networks that map directly to an 802.1Q VLAN in the cloud provider's physical network infrastructure. There are four aspects to a provider VLAN configuration:

configuration (e.g. the top-of-rack switch)
iguration (for compute nodes and Neutron network nodes)
ile settings
onding network objects in Neutron

structure must be configured to convey the provider VLAN traffic as tagged VLANs to the cloud compute nodes and Neutron network nodes. Configuration of the physical network infrastructure is outside the sc

automates the server networking configuration and the Neutron configuration based on information in the cloud definition. To configure the system for provider VLANs, specify the `neutron.networks.vl` ribute on one or more **network-groups** as described in the *Network Groups* section. For example (some attributes omitted for brevity):

```
ROUP_A

.networks.vlan:
vider-physical-network: physnet1
```

ated with a server network interface via an **interface-model** as described in the *Interface Models* section. For example (some attributes omitted for brevity):

```
s:
RFACE_SET_X
terfaces:

e: bond0
-groups:
_GROUP_A

e: hed3
-groups:
_GROUP_B
```

provider VLANs may contain only a single HPE Helion OpenStack **network**, because that VLAN must span all compute nodes and any Neutron network nodes/controllers (i.e. it is a single L2 segment). The H
`red-vlan: false`, otherwise a linux VLAN network interface will be created. For example:

```
A
n: false
oup: NET_GROUP_A
B
n: false
oup: NET_GROUP_B
```

d, HPE Helion OpenStack 5.0 will create the appropriate bridges on the servers, and set the appropriate attributes in the Neutron configuration files (e.g. bridge_mappings).

ployed, create Neutron network objects for each provider VLAN using the Neutron CLI:

```
e --provider:network_type vlan --provider:physical_network physnet1 --provider:segmentation_id 101 mynet101
```

```
e --provider:network_type vlan --provider:physical_network physnet2 --provider:segmentation_id 234 mynet234
```

5.0: Standalone Lifecycle Manager

ployer-in-the-cloud" scenario where the first controller is also the deployer/lifecycle manager. If you want to use a standalone lifecycle manager, you will need to add the relevant details in `control_plane.y`

```
licy: strict
x: c0
 1

DEPLOYER-ROLE
nents:
anager
```

```
adm
.20.21
ue
:d4:b5:ce:18
-DL360-4PORT
ROLE
ACK1
```

```
LOYER-600GB-DISKS
: DEPLOYER-INTERFACES
ROLE
```

ml:

```
er
0
sdc
sdd
sde
j
600GB-DISKS
es:
4
```

® 5.0: Configuration Processor Information Files

of the data needed to deploy and configure the cloud, the configuration processor also creates a number of information files that provide details of the resulting configuration.

`~/helion/my_cloud/info` after the first configuration processor run. This directory is also rebuilt each time the Configuration Processor is run.

ML format, allowing them to be used in further automation tasks if required.

File	Provides details of
	IP address assignments on each network

	Routes that need to be configured between networks.
	How servers have been allocated, including their network configuration. Allows details of a server to
	Details of where components of each service are deployed
l	Details the structure of the cloud from the perspective of each control-plane
	Details the structure of the cloud from the perspective of each control-plane
	Details the structure of the cloud from the perspective of each region
	Details the structure of the cloud from the perspective of each service
al	Details the secrets that are generated by the configuration processor – the names of the secrets, along secret and a list of the clusters on which the service that consumes the secret is deployed
	Details the secrets that have been changed by the configuration processor – information for each secr `private_data_metadata.yml`
	An explanation of the decisions the configuration processor has made when allocating servers and ne
	A pictorial representation of the cloud

m the `entry-scale-kvm-vsa` example configuration.

' 5.0: address_info.yml

all the IP addresses allocated by the Configuration Processor:

```
>
works>
ss>
of Aliases>
```

```
-NET:

on-cp1-c1-m1-extapi

on-cp1-c1-m2-extapi

on-cp1-c1-m3-extapi

on-cp1-vip-public-SWF-PRX-extapi
on-cp1-vip-public-FRE-API-extapi
on-cp1-vip-public-GLA-API-extapi
on-cp1-vip-public-HEA-ACW-extapi
on-cp1-vip-public-HEA-ACF-extapi
on-cp1-vip-public-NEU-SVR-extapi
on-cp1-vip-public-KEY-API-extapi
on-cp1-vip-public-MON-API-extapi
on-cp1-vip-public-HEA-API-extapi
```

```
NET: {}


on-cp1-c1-m1-guest

on-cp1-c1-m2-guest

on-cp1-c1-m3-guest

on-cp1-comp0001-guest
```

5.0: firewall_info.yml

all the network ports that will be opened on the deployed cloud. Data is ordered by network. If you want to configure an external firewall in front of the External API network, then you would need to open the p

```
>
IP Addresses>
Components>
```

```
cp



api
'
cp
```

etwork EXTERNAL-API for address 10.0.1.5 because it is used by Horizon

network EXTERNAL-API for address 10.0.1.5 because it is used by Keystone API

5.0: net_info.yml

IP addresses that have been allocated for a service. This data is typically used for service configuration after the initial deployment.

```
tname:  <Hostname of server in the cluster>
address: <IP address of server the cluster>
```

```
: vsa
_ip:
tname: helion-cp1-vsa-VSA-BLK-mgmt
address: 192.168.10.7
_plane: control-plane-1

tname: helion-cp1-vsa0001-VSA-BLK-mgmt
address: 192.168.10.2
tname: helion-cp1-vsa0002-VSA-BLK-mgmt
address: 192.168.10.8
tname: helion-cp1-vsa0003-VSA-BLK-mgmt
address: 192.168.10.12
: MANAGEMENT-NET
```

ontrol-plane-1" has been allocated 192.168.10.7 on network MANAGEMENT-NET as a cluster address and consists of 3 servers with addresses 192.168.10.2, 192.168.192.8, and 192.168.10.12.

? 5.0: route_info.yml

f routes between networks that need to be configured. Available routes are defined in the input model as part of the **network-groups** data; this file shows which routes will actually be used. HPE Helion OpenStac
nfigure the corresponding routes within your physical network. Routes must be configured to be symmetrical -- only the direction in which a connection is initiated is captured in this file.

ay not require any routes, with all servers being attached to common L3 networks. The following example is taken from the `tech-preview/mid-scale-kvm-vsa` example.

```
-Name>
work-Name>
t:   <true if this is this the result of a "default" route rule>
y:
source-service>
    <target-service>
    <list of hosts using this route>
```

```
CK1:
NET:
 false

meter-client:
meter-api:
ion-cp1-mtrmon-m1
one-api:
ion-cp1-mtrmon-m1
ET-RACK2:
 false
```

twork **MANAGEMENT-NET-RACK1** to network **MANAGEMENT-NET-RACK2** so that **cinder-backup** can connect to **rabbitmq** from server **helion-cp1-core-m1**

' 5.0: server_info.yml

f how servers have been allocated by the Configuration Processor. This provides the easiest way to find where a specific physical server (identified by `server-id`) is being used.

```
zone: <failure zone that the server was allocated from>
: <hostname of the server>
: <network configuration>
 "allocated" | "available" >
```

```
zone: AZ1
: helion-cp1-c1-m1-mgmt
:
D0:
  EXTERNAL-API-NET:
      addr: 10.0.1.2
      tagged-vlan: true
      vlan-id: 101
  EXTERNAL-VM-NET:
      addr: null
      tagged-vlan: true
      vlan-id: 102
  GUEST-NET:
      addr: 10.1.1.2
      tagged-vlan: true
      vlan-id: 103
  MANAGEMENT-NET:
      addr: 192.168.10.3
      tagged-vlan: false
      vlan-id: 100
llocated
```

' 5.0: service_info.yml

f how services are distributed across the cloud.

```
ne>

e component>
ist of hosts>
```

```
:
```

```
 helion-cp1-c1-m3-mgmt
ron-l3-agent:
- helion-cp1-comp0001-mgmt
ron-lbaasv2-agent:
- helion-cp1-comp0001-mgmt
```

? 5.0: control_plane_topology.yml

' the topology of the cloud from the perspective of each control plane:

```
name>
ers:
lancer-name>:
ess:  <IP address of VIP>
-file:  <name of cert file>
rnal-name: <name to used for endpoints>
ork: <name of the network this LB is connected to>
ork_group: <name of the network group this LB is connect to
ider: <service component providing the LB>
s:  <list of roles of this LB>
ices:
service-name>:
   <component-name>:
       aliases:
          <role>:  <Name in /etc/hosts>
       host-tls:  <Boolean, true if connection from LB uses TLS>
       hosts:  <List of hosts for this service>
       port:  <port used for this component>
       vip-tls: <Boolean, true if the VIP terminates TLS>

r-name>
lure-zones:
<failure-zone-name>:
   <list of hosts>
vices:
<service name>:
    components:
       <list of service components>
    regions:
       <list of region names>

e-name>:
for clusters above>
```

```
s:
-1:
```

```
    - helion-cp1-c1-m3-mgmt
ervices:
   barbican:
       components:
       - barbican-api
       - barbican-worker
       regions:
       - region1
                              …
ncers:
:
ddress: 10.0.1.5
ert-file: my-public-entry-scale-kvm-vsa-cert
xternal-name: ''
etwork: EXTERNAL-API-NET
etwork-group: EXTERNAL-API
rovider: ip-cluster
oles:
 public
ervices:
   barbican:
       barbican-api:
           aliases:
               public: helion-cp1-vip-public-KEYMGR-API-extapi
           host-tls: true
           hosts:
           - helion-cp1-c1-m1-mgmt
           - helion-cp1-c1-m2-mgmt
           - helion-cp1-c1-m3-mgmt
           port: '9311'
           vip-tls: true
```

' 5.0: network_topology.yml

f the topology of the cloud from the perspective of each network_group:

```
name>:
me>:
-planes:
ntrol-plane-name>:
 clusters:
    <cluster-name>:
        servers:
           <hlm-server-name>: <ip address>
        vips:
           <ip address>: <load balancer name>
 resources:
    <resource-group-name>:
        servers:
           <hlm-server-name>: <ip address>
```

```
    clusters:
        cluster1:
            servers:
                helion-cp1-c1-m1: 10.0.1.2
                helion-cp1-c1-m2: 10.0.1.3
                helion-cp1-c1-m3: 10.0.1.4
            vips:
                10.0.1.5: extlb

VM-NET:
ol_planes:
ontrol-plane-1:
    clusters:
        cluster1:
            servers:
                helion-cp1-c1-m1: null
                helion-cp1-c1-m2: null
                helion-cp1-c1-m3: null
    resources:
        compute:
            servers:
                helion-cp1-comp0001: null
```

? 5.0: region_topology.yml

f the topology of the cloud from the perspective of each region:

```
nes:
l-plane-name>:
vices:
<service-name>:
    <list of service components>
```

```
lanes:
ol-plane-1:
ervices:
    barbican:
    - barbican-api
    - barbican-worker
    ceilometer:
    - ceilometer-common
    - ceilometer-agent-notification
    - ceilometer-api
    - ceilometer-polling
    cinder:
    - cinder-api
    - cinder-volume
```

```
>:
s:
onent-name>:
ontrol-planes:
   <control-plane-name>:
       clusters:
           <cluster-name>:
               <list of servers>
       resources:
           <resource-group-name>:
               <list of servers>
       regions:
           <list of regions>
```

```
s:
er-agent:
ontrol_planes:
   control-plane-1:
       clusters:
           cluster1:
           - helion-cp1-c1-m1-mgmt
           - helion-cp1-c1-m2-mgmt
           - helion-cp1-c1-m3-mgmt
       regions:
       - region1
       resources:
           compute:
           - helion-cp1-comp0001-mgmt
           vsa:
           - helion-cp1-vsa0001-mgmt
           - helion-cp1-vsa0002-mgmt
           - helion-cp1-vsa0003-mgmt
       regions:
       - region1
```

5.0: private_data_metadata.yml

the secrets that are generated by the configuration processor. The details include:

et

cret. This is a list where each element contains details about each `component` service that uses the secret.

ervice that uses the secret, and if applicable the service that this component "consumes" when using the secret

n which the `component` service is deployed

on which the services are deployed

model version number)

```
omponent>
onsumes>
ontrol-plane>
```

```
ssword:

1
: barbican-api

'
ssword:

1
: swift-proxy
 keystone-api

'
ared_secret:

1
: nova-metadata

1

: neutron-metadata-agent

'
```

5.0: password_change.yml

quivalent to those in private_data_metadata.yml for passwords which have been changed from their original values, using the procedure outlined in the HPE Helion OpenStack documentation

5.0: explain.txt

the server allocation and network configuration decisions the configuration processor has made. The sequence of information recorded is:

- s that are automatically added
- clusters and resource groups
- rk configuration for each server
- rk configuration of each load balancer

```
red services to control plane control-plane-1
```

```
cluster1
--------
ed allocation for server 'controller1' (AZ1)
ed allocation for server 'controller2' (AZ2)
ng for server with role ['CONTROLLER-ROLE'] in zones: set(['AZ3'])
ed server 'controller3' (AZ3)

 vsa
----
ed allocation for server 'vsa1' (AZ1)
ed allocation for server 'vsa2' (AZ2)
ed allocation for server 'vsa3' (AZ3)
ng for server with role ['VSA-ROLE'] in zones: set(['AZ1', 'AZ2', 'AZ3'])

 compute
--------
ed allocation for server 'compute1' (AZ1)
ng for server with role ['COMPUTE-ROLE'] in zones: set(['AZ1', 'AZ2', 'AZ3'])

etworks for Servers
===================
elion-cp1-c1-m1
---------------
ERNAL-API for component ip-cluster
AGEMENT for component ip-cluster
AGEMENT for lifecycle-manager (default)
AGEMENT for ntp-server (default)

AGEMENT for swift-rsync (default)
ST for tag neutron.networks.vxlan (neutron-openvswitch-agent)
ERNAL-VM for tag neutron.l3_agent.external_network_bridge (neutron-vpn-agent)
ersisted address 10.0.1.2 for server helion-cp1-c1-m1 on network EXTERNAL-API-NET
ddress 192.168.10.3 for server helion-cp1-c1-m1 on network MANAGEMENT-NET
ersisted address 10.1.1.2 for server helion-cp1-c1-m1 on network GUEST-NET


ad balancers
============

ncer: extlb
-----------
ersisted address 10.0.1.5 for vip extlb helion-cp1-vip-extlb-extapi on network EXTERNAL-API-NET
a-api for roles ['public'] due to 'default'
nce-api for roles ['public'] due to 'default'


balancers to providers
======================

XTERNAL-API-NET
---------------
5: ip-cluster nova-api roles: ['public'] vip-port: 8774 host-port: 8774
5: ip-cluster glance-api roles: ['public'] vip-port: 9292 host-port: 9292
5: ip-cluster keystone-api roles: ['public'] vip-port: 5000 host-port: 5000
5: ip-cluster swift-proxy roles: ['public'] vip-port: 8080 host-port: 8080
5: ip-cluster monasca-api roles: ['public'] vip-port: 8070 host-port: 8070
```

```
5: ip-cluster freezer-api roles: ['public'] vip-port: 9090 host-port: 9090
5: ip-cluster horizon roles: ['public'] vip-port: 443 host-port: 80
5: ip-cluster cinder-api roles: ['public'] vip-port: 8776 host-port: 8776
```

' 5.0: CloudDiagram.txt

l representation of the cloud. Although this file is still produced, it is superseded by the HTML output described in the following section.

```
Plane: region1 (control-plane-1)-------------------------------------------------------------------------------------

er cluster1 ()---------------------------------------------------------------------------------------------------------

ion-cp1-c1-m1 (192.168.10.3)-------------+    +-helion-cp1-c1-m2 (192.168.10.4)-------------+    +-helion-cp1-c1-m3 (192.168.10.5)---------
                                         |    |                                             |    |
lometer                                  |    | ceilometer                                  |    | ceilometer
eilometer-agent-central                  |    |   ceilometer-agent-central                  |    |   ceilometer-agent-central
eilometer-agent-notification             |    |   ceilometer-agent-notification             |    |   ceilometer-agent-notification
eilometer-api                            |    |   ceilometer-api                            |    |   ceilometer-api
eilometer-client                         |    |   ceilometer-client                         |    |   ceilometer-client
eilometer-collector                      |    |   ceilometer-collector                      |    |   ceilometer-collector
eilometer-common                         |    |   ceilometer-common                         |    |   ceilometer-common
eilometer-expirer                        |    |   ceilometer-expirer                        |    |   ceilometer-expirer
der                                      |    | cinder                                      |    | cinder
inder-api                                |    |   cinder-api                                |    |   cinder-api
inder-backup                             |    |   cinder-backup                             |    |   cinder-backup
inder-client                             |    |   cinder-client                             |    |   cinder-client
inder-scheduler                          |    |   cinder-scheduler                          |    |   cinder-scheduler
inder-volume                             |    |   cinder-volume                             |    |   cinder-volume
ndation                                  |    | foundation                                  |    | foundation
pache2                                   |    |   apache2                                   |    |   apache2
p-cluster                                |    |   ip-cluster                                |    |   ip-cluster
afka                                     |    |   kafka                                     |    |   kafka
emcached                                 |    |   memcached                                 |    |   memcached
ysql                                     |    |   mysql                                     |    |   mysql
tp-server                                |    |   ntp-server                                |    |   ntp-server
penstack-client                          |    |   openstack-client                          |    |   openstack-client
abbitmq                                  |    |   rabbitmq                                  |    |   rabbitmq
   | | |
torm                                     |    |   storm                                     |    |   storm
tunnel                                   |    |   stunnel                                   |    |   stunnel
wift-common                              |    |   swift-common                              |    |   swift-common
wift-rsync                               |    |   swift-rsync                               |    |   swift-rsync
ertica                                   |    |   vertica                                   |    |   vertica
ookeeper                                 |    |   zookeeper                                 |    |   zookeeper
ezer                                     |    | freezer                                     |    | freezer
reezer-agent                             |    |   freezer-agent                             |    |   freezer-agent
reezer-api                               |    |   freezer-api                               |    |   freezer-api
nce                                      |    | glance                                      |    | glance
lance-api                                |    |   glance-api                                |    |   glance-api
lance-client                             |    |   glance-client                             |    |   glance-client
lance-registry                           |    |   glance-registry                           |    |   glance-registry
t                                        |    | heat                                        |    | heat
eat-api                                  |    |   heat-api                                  |    |   heat-api
```

```
eystone-api                              |     |   keystone-api                            |     |   keystone-api
eystone-client                           |     |   keystone-client                         |     |   keystone-client
ging                                     |     | logging                                   |     | logging
ogging-producer                          |     |   logging-producer                        |     |   logging-producer
ogging-server                            |     |   logging-server                          |     |   logging-server
asca                                     |     | monasca                                   |     | monasca
onasca-agent                             |     |   monasca-agent                           |     |   monasca-agent
onasca-api                               |     |   monasca-api                             |     |   monasca-api
onasca-client                            |     |   monasca-client                          |     |   monasca-client
onasca-notifier                          |     |   monasca-notifier                        |     |   monasca-notifier
onasca-persister                         |     |   monasca-persister                       |     |   monasca-persister
onasca-threshold                         |     |   monasca-threshold                       |     |   monasca-threshold
tron                                     |     | neutron                                   |     | neutron
eutron-client                            |     |   neutron-client                          |     |   neutron-client
eutron-dhcp-agent                        |     |   neutron-dhcp-agent                      |     |   neutron-dhcp-agent
eutron-metadata-agent                    |     |   neutron-metadata-agent                  |     |   neutron-metadata-agent
eutron-ml2-plugin                        |     |   neutron-ml2-plugin                      |     |   neutron-ml2-plugin
eutron-openvswitch-agent                 |     |   neutron-openvswitch-agent               |     |   neutron-openvswitch-agent
eutron-server                            |     |   neutron-server                          |     |   neutron-server
eutron-vpn-agent                         |     |   neutron-vpn-agent                       |     |   neutron-vpn-agent
a                                        |     | nova                                      |     | nova
ova-api                                  |     |   nova-api                                |     |   nova-api
ova-client                               |     |   nova-client                             |     |   nova-client
ova-conductor                            |     |   nova-conductor                          |     |   nova-conductor
ova-console-auth                         |     |   nova-console-auth                       |     |   nova-console-auth
ova-metadata                             |     |   nova-metadata                           |     |   nova-metadata
ova-novncproxy                           |     |   nova-novncproxy                         |     |   nova-novncproxy
ova-scheduler                            |     |   nova-scheduler                          |     |   nova-scheduler
rations                                  |     | operations                                |     | operations
ifecycle-manager                         |     |   lifecycle-manager                       |     |   lifecycle-manager
ifecycle-manager-target                  |     |   lifecycle-manager-target                |     |   lifecycle-manager-target
ps-console-monitor                       |     |   ops-console-monitor                     |     |   ops-console-monitor
ps-console-web                           |     |   ops-console-web                         |     |   ops-console-web
ft                                       |     | swift                                     |     | swift
wift-account                             |     |   swift-account                           |     |   swift-account
wift-client                              |     |   swift-client                            |     |   swift-client
wift-container                           |     |   swift-container                         |     |   swift-container
wift-object                              |     |   swift-object                            |     |   swift-object
wift-proxy                               |     |   swift-proxy                             |     |   swift-proxy
wift-ring-builder                        |     |   swift-ring-builder                      |     |   swift-ring-builder
-storage                                 |     | vsa-storage                               |     | vsa-storage
mc-service                               |     |   cmc-service                             |     |   cmc-service
                                         |     |                                           |     |
---------------------------------------- |     | ----------------------------------------- |     | -----------------------------------------
                                         |     |                                           |     |
d0 (hed3, hed4)                          |     | bond0 (hed3, hed4)                        |     | bond0 (hed3, hed4)
XTERNAL-API-NET (10.0.1.2)               |     |   EXTERNAL-API-NET (10.0.1.3)             |     |   EXTERNAL-API-NET (10.0.1.4)
XTERNAL-VM-NET                           |     |   EXTERNAL-VM-NET                         |     |   EXTERNAL-VM-NET
UEST-NET (10.1.1.2)                      |     |   GUEST-NET (10.1.1.3)                    |     |   GUEST-NET (10.1.1.4)
ANAGEMENT-NET (192.168.10.3)             |     |   MANAGEMENT-NET (192.168.10.4)           |     |   MANAGEMENT-NET (192.168.10.5)
                                         |     |                                           |     |
-----------------------------------------+     +-------------------------------------------+     +-----------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------

te-------------------------------------------+
```

```
reezer-agent                                 | |
ging                                         | |
ogging-producer                              | |
asca                                         | |
onasca-agent                                 | |
tron                                         | |
eutron-l3-agent                              | |
eutron-lbaasv2-agent                         | |
eutron-metadata-agent                        | |
eutron-openvswitch-agent                     | |
a                                            | |
ova-compute                                  | |
ova-compute-kvm                              | |
rations                                      | |
ifecycle-manager-target                      | |
                                             | |
-----------------------------------------    | |
                                             | |
d0 (hed3, hed4)                              | |
XTERNAL-VM-NET                               | |
UEST-NET (10.1.1.0/24)                       | |
ANAGEMENT-NET (192.168.10.0/24)              | |
                                             | |
---------------------------------------------+ |
-----------------------------------------------+

-------------------------------------------------------------------------------------------------------------------------------------------------

-ROLE (AZ1) (1 servers)--------------------+    +-VSA-ROLE (AZ2) (1 servers)--------------------+    +-VSA-ROLE (AZ3) (1 servers)--------------
                                           |    |                                               |    |
ndation                                    |    | foundation                                    |    | foundation
tp-client                                  |    |   ntp-client                                  |    |   ntp-client
tunnel                                     |    |   stunnel                                     |    |   stunnel
ezer                                       |    | freezer                                       |    | freezer
reezer-agent                               |    |   freezer-agent                               |    |   freezer-agent
ging                                       |    | logging                                       |    | logging
ogging-producer                            |    |   logging-producer                            |    |   logging-producer
asca                                       |    | monasca                                       |    | monasca
onasca-agent                               |    |   monasca-agent                               |    |   monasca-agent
rations                                    |    | operations                                    |    | operations
ifecycle-manager-target                    |    |   lifecycle-manager-target                    |    |   lifecycle-manager-target
-storage                                   |    | vsa-storage                                   |    | vsa-storage
sa                                         |    |   vsa                                         |    |   vsa
                                           |    |                                               |    |
-----------------------------------------  |    |  -------------------------------------------  |    |  ----------------------------------------
                                           |    |                                               |    |
d0 (hed3, hed4)                            |    | bond0 (hed3, hed4)                            |    | bond0 (hed3, hed4)
ANAGEMENT-NET (192.168.10.0/24)            |    |   MANAGEMENT-NET (192.168.10.0/24)            |    |   MANAGEMENT-NET (192.168.10.0/24)
                                           |    |                                               |    |
-------------------------------------------+    +-----------------------------------------------+    +----------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------------
```

e View | Region View | Service View | Network View | Server View | Server Groups View

-1

sters	Resources		Load Balancers	
ster1	vsa	compute	extlb	lb
bican			barbican	barbican
ometer			ceilometer	ceilometer
nder			cinder	cinder
ignate			designate	designate
eezer	freezer	freezer	freezer	freezer
ance			glance	glanc
neat			heat	heat
rizon			horizon	horizo
ystone			keystone	keysto
gging	logging	logging	logging	loggi
nasca	monasca	monasca	monasca	monas
utron		neutron	neutron	neutron
ova		nova	nova	nova
tavia				octavia
rations			operations	operations
wift			swift	swift
npest				
storage	vsa-storage			
ndation	foundation	foundation		foundation
ients				
hlm	hlm	hlm	hlm	hlm
			10.0.1.5	192.168.10.13
-c1-m1-mgmt	helion-cp1-vsa0001-mgmt	helion-cp1-comp0001-mgmt		
-c1-m2-mgmt	helion-cp1-vsa0002-mgmt			

freezer-api (9090) TLS
helion-cp1-vip-admin-FRE-API-mgmt
helion-cp1-vip-FRE-API-mgmt
hosts:
helion-cp1-c1-m1-mgmt
helion-cp1-c1-m2-mgmt
helion-cp1-c1-m3-mgmt

ology

	control-plane-1		
	Clusters	**Resources**	
	cluster1	**vsa**	**compute**
	EXTERNAL-API-NET		
	MANAGEMENT-NET	MANAGEMENT-NET	MANAGEMENT-NET
	GUEST-NET		GUEST-NET
	EXTERNAL-VM-NET		EXTERNAL-VM-NET
-NET			

ups

I

	Networks	Address	Server	Interface Model
1	EXTERNAL-API-NET vlan id: 101 (tagged) cidr: 10.0.1.0/24 gateway-ip: 10.0.1.1 mtu: 1500	10.0.1.4 10.0.1.3 10.0.1.2 10.0.1.5	helion-cp1-c1-m3 helion-cp1-c1-m2 helion-cp1-c1-m1 extlb	CONTROLLER-INTERFACES

k input model allows a wide variety of configuration parameters that may, at first glance, appear daunting. The example configurations are designed to simplify this process by providing pre-built and pre-qualifie ications to get started.

Example Configurations

es the various example configurations and their capabilities. It also describes in detail, for the entry-scale-kvm-vsa example, how you can adapt the input model to work in your environment.

l examples are shipped with HPE Helion OpenStack 5.0:

Name	Location
A model	`~/helion/examples/entry-scale-kvm-vsa`
A model with Dedicated Cluster for Metering, Monitoring, and Logging	`~/helion/examples/entry-scale-kvm-vsa-mml`
ph model	`~/helion/examples/entry-scale-kvm-ceph`
model	`~/helion/examples/mid-scale-kvm-vsa`
d VSA model	`~/helion/examples/entry-scale-esx-kvm-vsa`
d VSA model with Dedicated Cluster for Metering, Monitoring, and Logging	`~/helion/examples/entry-scale-esx-kvm-vsa-mml`
del	`~/helion/examples/entry-scale-swift`
nic Flat Network	`~/helion/examples/entry-scale-ironic-flat-network`
nic Multi-Tenancy	`~/helion/examples/entry-scale-ironic-multi-tenancy`

e designed to provide an entry-level solution that can be scaled from a small number of nodes to a moderately high node count (approximately 100 compute nodes, for example).

cloud control plane is subdivided into a number of dedicated service clusters to provide more processing power for individual control plane elements. This enables a greater number of resources to be supported nows how a segmented network can be expressed in the HPE Helion OpenStack model.

ale KVM with VSA Model for Your Environment

nges that need to be made to the input model to deploy and run this cloud model in your environment.

- *del*
- *Model*

ons

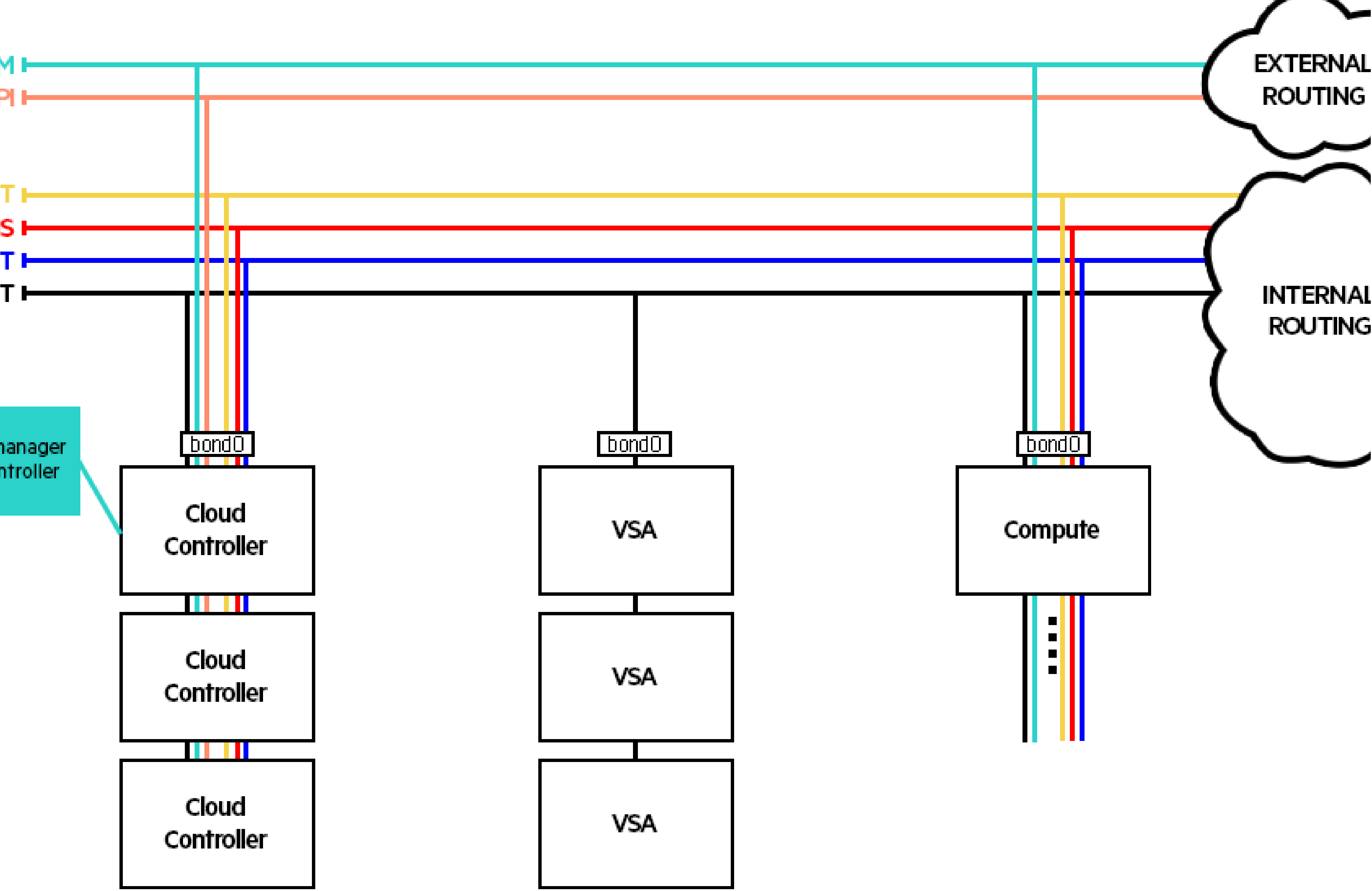
5.0 there are alternative configurations that we recommend for specific purposes and this section we will outline them.

- *Ceph Model with One Network*
- *Ceph Model with Two Networks*
- *ecycle-Manager Node*
- *n OpenStack without DVR*
- *n OpenStack with Provider VLANs and Physical Routers Only*
- *nstalling Two Systems on One Subnet*

® 5.0: KVM Examples

® 5.0: Entry-scale KVM with VSA Model

loyed without the VSA servers and configured to use an external storage device, such as a 3PAR array, which would reduce the minimum server count to four.



IPM/iLO network (not shown) is connected to all controllers.

ollowing networks:

the network that users will use to make requests to the cloud.
the network that will be used to provide access to virtual machines (via floating IP addresses).
 the network that will carry traffic between virtual machines on private networks within the cloud.
- This is the network that will be used for the Octavia load balancing service.
the network that will be used for all internal traffic between the cloud services, including node provisioning. This network must be on an untagged VLAN.

onfigured to be presented via a pair of bonded NICs. The example also enables additional provider VLANs to be configured in Neutron on this interface.

outing" refers to whatever routing you want to provide to allow users to access the External API and External VM networks. Note that the EXTERNAL_API network must be reachable from the EXTERNAL_V
e API calls to the cloud. "Internal Routing" refers to whatever routing you want to provide to allow administrators to access the Management network.

n OpenStack to install the operating system, then an IPMI/iLO network connected to the IPMI/iLO ports of all servers and routable from the lifecycle manager server is also required for BIOS and power manage
n process.

wing disk configurations:

ating system disk and two disks for Swift storage.
stem disk and two disks for VSA storage.
ng system disk and one disk for virtual machine ephemeral storage.

odify this example to match your environment, see *Modifying the Entry-scale KVM with VSA model for your Environment*.

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor
ur hardware.

uirements detailed below can be met with logical drives, logical volumes, or external storage such as a 3PAR array.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	Compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the V Con
	VSA or OSD (Ceph)	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details.	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

supported network requirements, see *Example Configurations*.

5.0: Entry-scale KVM with VSA model with Dedicated Cluster for Metering, Monitoring, and Logging ed processing power for these services, the following configuration changes are made to the control plane in this model:

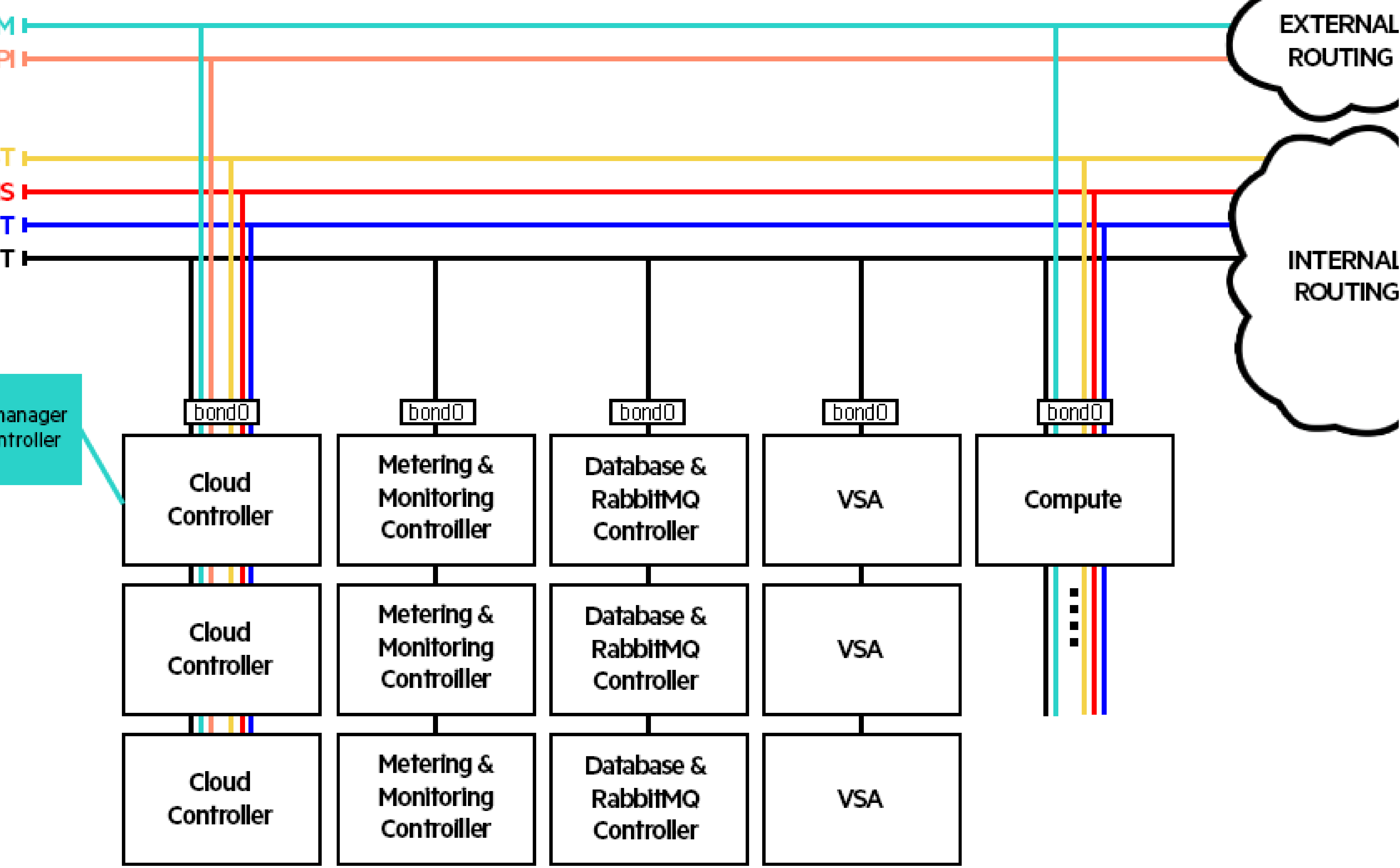
with metering, monitoring, and logging run on a dedicated three-node cluster. Three nodes are required for high availability with quorum.

cluster is used for RabbitMQ message queue and database services. This cluster is also used to provide additional processing for the message queue and database load associated with the additional metering, mo
igh availability with quorum.

reduced to two nodes. These services are stateless and do not require a quorum node for high availability.

tes the physical networking used in this configuration.

edicated Cluster for Metering, Monitoring, and Logging



IPM/iLO network (not shown) is connected to all controllers.

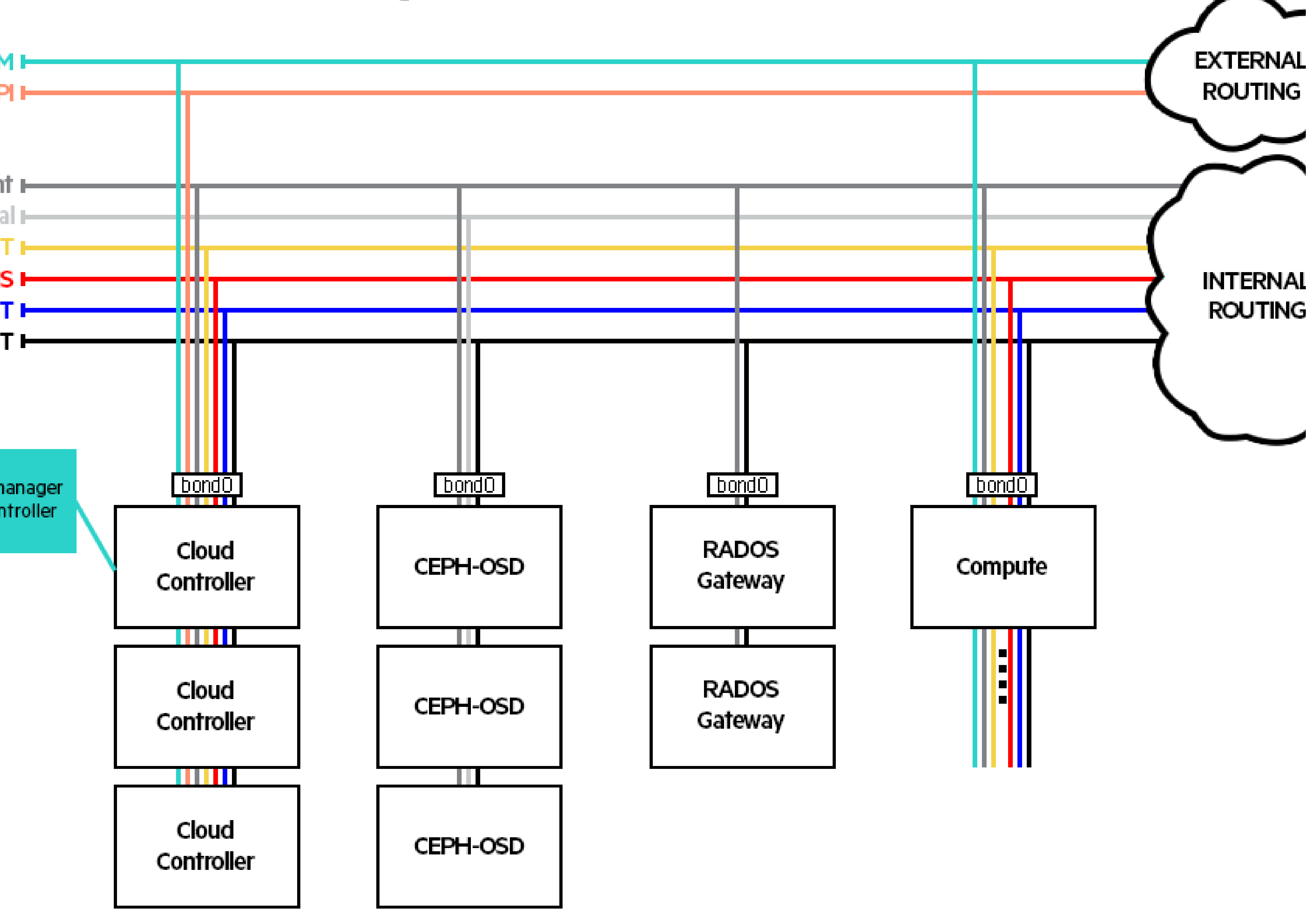
5.0: Entry-scale KVM with Ceph Model

VM-based cloud using Ceph for both block and object storage.

egated into the following VLANs:

This is the network that will be used for all internal traffic between the cloud services.
the network that will be used for internal traffic of cluster among Ceph OSD servers. Only Ceph OSD servers will need connectivity to this network.
e network that Ceph clients will use to talk to Ceph Monitor and OSDs. Cloud controllers, Nova Compute, Ceph Monitor, OSD and Rados Gateway servers will need connectivity to this network.

tes the physical networking used in this configuration. Click any network name in the diagram to see that network isolated.



M
PI
nt
al
T
S
T
T
EXTERNAL
ROUTING
INTERNAL
ROUTING
anager
ntroller
bond0
bond0
bond0
bond0
Cloud Controller
Cloud Controller
Cloud Controller
CEPH-OSD
CEPH-OSD
CEPH-OSD
RADOS Gateway
RADOS Gateway
Compute

on the entry-scale-kvm-ceph cloud input model which is included with the HPE Helion OpenStack distro. You will need to make the changes outlined below prior to the deployment of your Ceph cluste

e key characteristics needed per server role for this configuration.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
or)	Compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the Con
	ceph-osd	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum)	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	radosgw	2	2 x 600 GB (minimum)	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

server NIC interfaces are correctly specified in the ~/helion/my_cloud/definition/data/nic_mappings.yml file and that they meet the server requirements.

tes in-line:

```
ion for controller nodes.  A bonded interface is used for the management
a separate interface is used to connect to the Ceph nodes.
LER-NIC-MAPPING
s:
name: hed1
mple-port
ess: "0000:07:00.0"

name: hed2
mple-port
ess: "0000:08:00.0"

name: hed3
mple-port
ess: "0000:09:00.0"

name: hed4
mple-port
```

```
ess: "0000:04:00.0"

name: hed4
mple-port
ess: "0000:04:00.1"

ion for OSD nodes. The first interface is used for management network
econd interface is used for client or public traffic. The third
sed for internal OSD traffic.
-MAPPING
s:
name: hed1
mple-port
ess: "0000:06:00.0"

name: hed2
mple-port
ess: "0000:06:00.1"

name: hed3
mple-port
ess: "0000:06:00.2"

ion for RADOS Gateway nodes. The first interface is used for management network
econd interface is used for client or public traffic.
-MAPPING
s:
name: hed1
mple-port
ess: "0000:07:00.0"

name: hed2
mple-port
ess: "0000:07:00.1"
```

the `~/helion/my_cloud/definition/data/servers.yml` file are mapped to the correct NIC interface.

d line for `nic-mapping` illustrating this:

```
s
r1
3.111.138
 RACK1
LER-ROLE
CONTROLLER-NIC-MAPPING
:92:1c:05:69:10"
.8.214
 password
in



3.111.139
```

```
3.111.140
 RACK1
E
OSD-NIC-MAPPING
:92:1c:25:69:e0"
.8.216
 password
in



168.10.12
E
 RACK1
RGW-NIC-MAPPING
:f6:9e:ca:3b:62"
68.9.12
 password
in


168.10.13
E
 RACK2
RGW-NIC-MAPPING
:f6:9e:ca:3b:63"
68.9.13
 password
in
```

or your OSD interfaces in the `~/helion/my_cloud/definition/data/net_interfaces.yml` file.

tes in-line:

```
R-INTERFACES
ces:
terface is used by the controller
d management traffic.


nd0


 active-backup
n: 200
ry: hed1
: linux

e: hed1
e: hed2
```

```
 interface.


3
ups:
ENT

NTERFACES
ces:



d3
ork-groups:
AL-VM

MENT
 is used to connect the compute node
uster so that a workload VM can route
o the Ceph cluster over this interface.



ed4
work-groups:
LIENT

FACES
ces:
he interface used for management
ogging, monitoring, etc.



ed1
ups:
ENT
he interface used for client
c.



ed2
ups:
ENT
he interface used for internal
ication among OSD nodes.



ed3
ups:
ERNAL

TERFACES
rfaces:
ND0

 bond0
a:
ns:
```

```
groups:
GEMENT
CLIENT
```

oup in the ~/helion/my_cloud/definition/data/network_groups.yml file:

```
work group that will be used for
c of cluster among OSDs.

T
: osdc

ints
r

w




work group that will be used for
c of cluster among OSDs.

NAL
: osdi

ints:
ternal
```

he ~/helion/my_cloud/definition/data/networks.yml file.

two separate network VLANs:

```

T-NET

ue
87.0/24
.168.187.1
OSD-CLIENT

NAL-NET

ue
00.0/24
.168.200.1
OSD-INTERNAL
```

```
-NET
NET

ET
ET- OSD-INTERNAL-NET
```

n the ~/helion/my_cloud/definition/data/firewall_rules.yml file to allow OSD nodes to be pingable via the OSD network, indicated by the bold portion below:

ng for OSD-CLIENT and OSD-INTERNAL is optional. Enabling ping on these networks might make debugging connectivity issues on these networks easier.

```
o request (ping)

fix:  0.0.0.0/0

n: 8

x: 0
p
```

and README.md Files

n/my_cloud/definition/README.html and ~/helion/my_cloud/definition/README.md files to reflect the OSD network group information if you wish. This change does not have any se
of your model.

ploying Ceph Monitor Services on Dedicated Resource Nodes

ck 5.0 example configurations, the Ceph monitor service is installed on the controller nodes by default. If you wish to break these out into their own cluster then you can do so by modifying the input model to f

want to deploy the monitor service as a dedicated resource node, then you must decide prior to the deployment of Ceph. HPE Helion OpenStack 5.0 does not support deployment transition. Once Ceph is deploye
m controller to dedicated resource nodes.

t be set up before starting Ceph deployment. For more details on the installation of the lifecycle manager, see *HPE Helion OpenStack 5.0: Installing Mid-scale and Entry-scale KVM*.

nanager.

y-scale-kvm-ceph example configuration as the base for these steps. Copy the example configuration files into the required setup directory before beginning the edit process:

```
examples/entry-scale-kvm-ceph/* ~/helion/my_cloud/definition/
```

ts to the ~/helion/my_cloud/definition/data/control_plane.yml file:

ce to - ceph-monitor under the service-components section for your control plane cluster.

our Ceph monitoring cluster. It is shown as the bolded portion in the example below, we added the rest to show the proper positioning:

```
uster1
orefix: c1
ole: CONTROLLER-ROLE
ount: 3
on-policy: strict
components:
cycle-manager
server


oh-mon
orefix: ceph-mon
ole: CEP-MON-ROLE
t: 3
on-policy: strict
components:
client
-monitor

w
orefix: rgw
ole: RGW-ROLE
```

ndentation in the file is important to review the file to ensure it matches before continuing on.

y_cloud/definition/data/servers.yml file to define all of the Ceph monitor nodes in the cluster. Here is an example, you will want to edit the values to match your environment:

```
Nodes

3.111.141
 RACK1
-ROLE
MY-4PORT-SERVER
:92:1c:05:69:10"
.8.217
 password
in


3.111.142
 RACK2
-ROLE
MY-4PORT-SERVER
```

```
 RACK3
-ROLE
MY-4PORT-SERVER
:92:1c:25:69:e0"
.8.219
 password
in

s
```

y_cloud/definition/data/net_interfaces.yml file to define a new network interface set for your Ceph monitors. You can copy the RGW-INTERFACES model as a base and then edit it to match

example:

```
s:
e device names and bond options
h your environment

TROLLER-INTERFACES
nterfaces:
ded interface is used by the controller
r cloud management traffic.
 BOND0
e:
me: bond0
data:
tions:
 mode: active-backup
 miimon: 200
 primary: hed1
ovider: linux
vices:
 - name: hed1
 - name: hed2
work-groups:
XTERNAL-API
XTERNAL-VM
UEST
ANAGEMENT
erface is used to connect the controller
the Ceph nodes so that any Ceph client
der-volume can route data directly to
r thisinterface.
 HETH3
e:
e: hed3
d-network-groups:
SD-CLIENT

PUTE-INTERFACES
nterfaces:
 HETH3
e:
ame: hed3
```

```
e:
ame: hed4
d-network-groups:
SD-CLIENT

-MON-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
 BOND0
e:
ame: bond0
data:
ptions:
   mode: active-backup
   miimon: 200
   primary: hed1
rovider: linux
evices:
 - name: hed1
 - name: hed2
rk-groups:
ANAGEMENT
nterface is used to connect the client
o the Ceph nodes so that any Ceph client
inder-volume can route data directly to
ver thisinterface.
 HETH3
e:
ame: hed3
d-network-groups:
SD-CLIENT

-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
 BOND0
e:
ame: bond0
data:
ptions:
   mode: active-backup
   miimon: 200
   primary: hed1
rovider: linux
evices:
 - name: hed1
 - name: hed2
rk-groups:
ANAGEMENT
efines the interface used for client
a traffic.
 HETH3
e:
ame: hed3
```

```
rk-groups:
SD-INTERNAL
```

ample:

```
s:
e device names and bond options
h your environment

TROLLER-INTERFACES
nterfaces:
ded interface is used by the controller
r cloud management traffic.
 interface is also used to connect the client
the Ceph nodes so that any Ceph client
der-volume can route data directly to
r thisinterface.
 BOND0
e:
me: bond0
data:
tions:
 mode: active-backup
 miimon: 200
 primary: hed1
ovider: linux
vices:
 - name: hed1
 - name: hed2
work-groups:
XTERNAL-API
XTERNAL-VM
UEST
ANAGEMENT

PUTE-INTERFACES
nterfaces:
me interface is also used to connect the compute node
 Ceph cluster so that a workload VM can route
raffic to the Ceph cluster over thisinterface.
 HETH3
e:
ame: hed3
rk-groups:
XTERNAL-VM
UEST
ANAGEMENT

-MON-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
me interface is also used to connect the client
o the Ceph nodes so that any Ceph client
inder-volume can route data directly to
```

```
   primary: hed1
rovider: linux
evices:
 - name: hed1
 - name: hed2
rk-groups:
ANAGEMENT

-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
me interface is also used for client
a traffic.
 BOND0
e:
ame: bond0
data:
ptions:
   mode: active-backup
   miimon: 200
   primary: hed1
rovider: linux
evices:
 - name: hed1
 - name: hed2
rk-groups:
ANAGEMENT
efines the interface used for internal
r communication among OSD nodes.
 HETH4
e:
ame: hed4
rk-groups:
SD-INTERNAL
```

example:

```
s:
e device names and bond options
h your environment

TROLLER-INTERFACES
nterfaces:
ded interface is used by the controller
r cloud management traffic.
 interface is also used to connect the client
the Ceph nodes so that any Ceph client
der-volume can route data directly to
r thisinterface.
 BOND0
e:
me: bond0
data:
tions:
 mode: active-backup
```

```
XTERNAL-API
XTERNAL-VM
UEST
ANAGEMENT

PUTE-INTERFACES
erface is also used to connect the compute node
eph cluster so that a workload VM can route
ffic to the Ceph cluster over thisinterface.
nterfaces:
 HETH3
e:
ame: hed3
rk-groups:
XTERNAL-VM
UEST
ANAGEMENT

-MON-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
me interface is also used to connect the client
o the Ceph nodes so that any Ceph client
inder-volume can route data directly to
ver thisinterface.
 BOND0
e:
ame: bond0
data:
ptions:
   mode: active-backup
   miimon: 200
   primary: hed1
rovider: linux
evices:
 - name: hed1
 - name: hed2
rk-groups:
ANAGEMENT

-INTERFACES
nterfaces:
efines the interface used for management
c like logging, monitoring, etc.
me interface is also used for internal cluster
ication among the OSD nodes.
me interface is also used for internal
r communication among OSD nodes.
 BOND0
e:
ame: bond0
data:
ptions:
   mode: active-backup
   miimon: 200
```

d disks_ceph_monitor.yml in the ~/helion/my_cloud/definition/data/ directory which will define the disk model for your Ceph monitors. You can use the disks_rgw.yml file as a base

```
-DISKS
 to be used for Ceph monitor nodes
ot is used as a volume group for /, /var/log and /var/crash
 a templated value to align with whatever partition is really used
is checked in os config and replaced by the partition actually used
 sda1 or sda5

:
-vg
volumes:
sda_root

olumes:
icy is not to consume 100% of the space of each volume group.
ld be left free for snapshots and to allow for some flexibility.
 root
 30%
e: ext4
: /
 log
 45%
: /var/log
e: ext4
opts: -O large_file
 crash
 20%
: /var/crash
e: ext4
opts: -O large_file

os
```

y_cloud/definition/data/server_roles.yml file to define a new server role for your Ceph monitors:

```
-ROLE
el: CEP-MON-INTERFACES
EP-MON-DISKS
```

tion:

```
/ansible

adding dedicated Ceph monitor cluster"
```

ook to add your nodes into Cobbler:

```
/ansible/
k -i hosts/localhost cobbler-deploy.yml
```

s using PXE, run the following playbook:

```
k -i hosts/localhost config-processor-run.yml
```

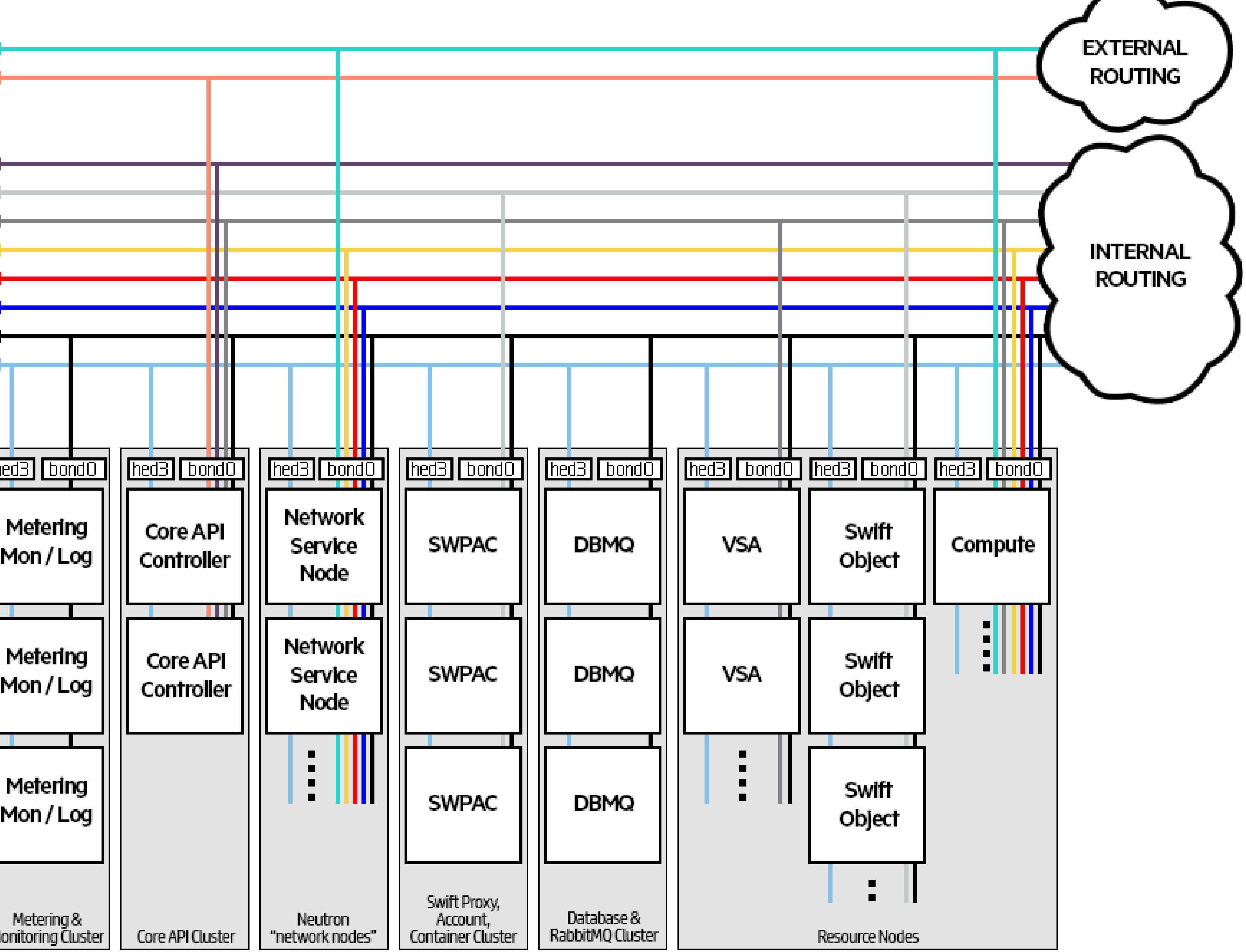
t directory with this playbook:

```
/ansible/
k -i hosts/localhost ready-deployment.yml
```

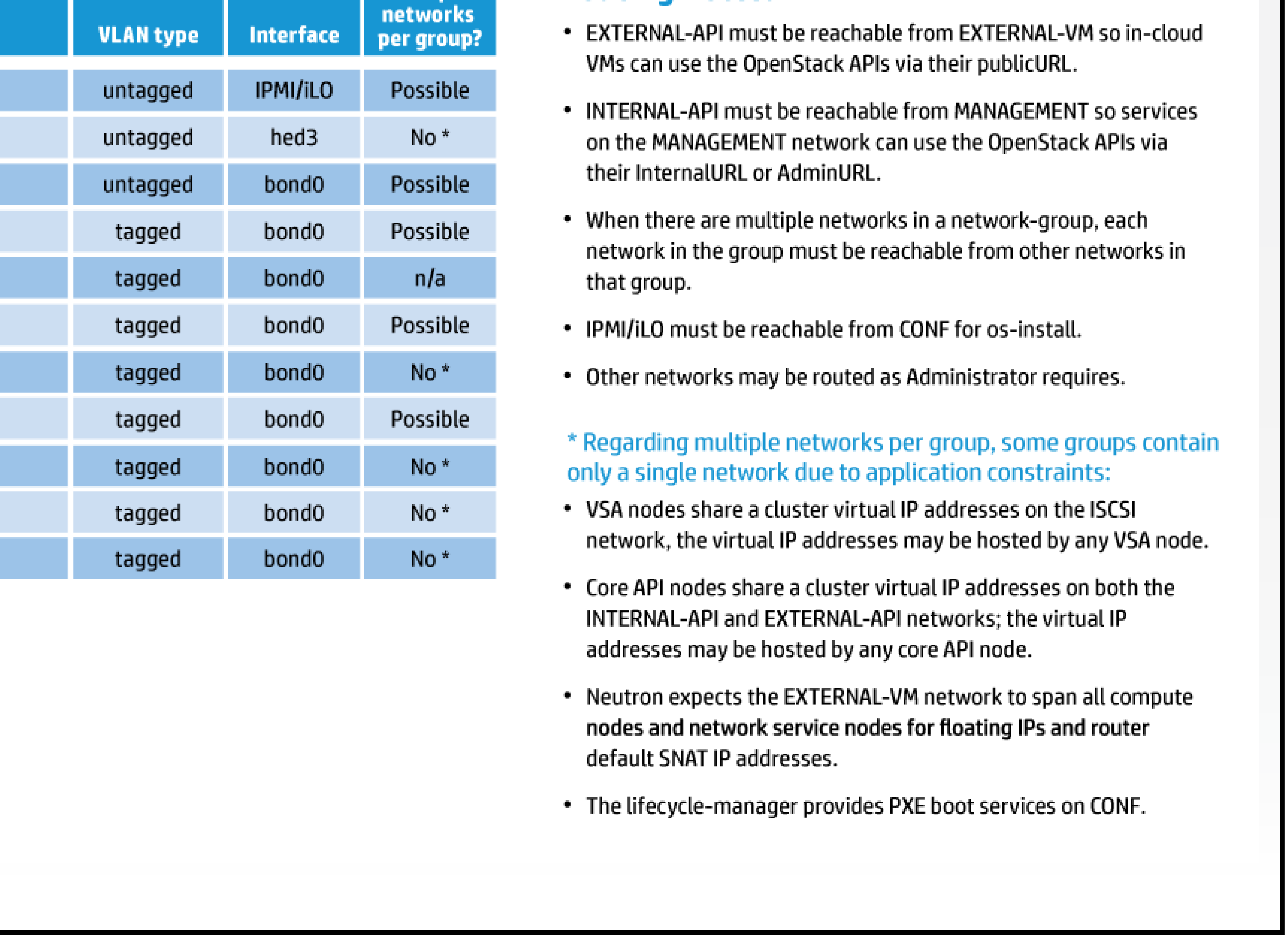
```
sible/next/hos/ansible
k -i hosts/verb_hosts site.yml
```

® 5.0: Mid-scale KVM with VSA Model

rates two important aspects of configuring HPE Helion OpenStack for increased scale. The controller services are distributed across a greater number of controllers and a number of the networks are configured a
working).



M/ILO network (not shown) is connected to all controllers



	VLAN type	Interface	networks per group?
	untagged	IPMI/iLO	Possible
	untagged	hed3	No *
	untagged	bond0	Possible
	tagged	bond0	Possible
	tagged	bond0	n/a
	tagged	bond0	Possible
	tagged	bond0	No *
	tagged	bond0	Possible
	tagged	bond0	No *
	tagged	bond0	No *
	tagged	bond0	No *

- EXTERNAL-API must be reachable from EXTERNAL-VM so in-cloud VMs can use the OpenStack APIs via their publicURL.
- INTERNAL-API must be reachable from MANAGEMENT so services on the MANAGEMENT network can use the OpenStack APIs via their InternalURL or AdminURL.
- When there are multiple networks in a network-group, each network in the group must be reachable from other networks in that group.
- IPMI/iLO must be reachable from CONF for os-install.
- Other networks may be routed as Administrator requires.

* Regarding multiple networks per group, some groups contain only a single network due to application constraints:

- VSA nodes share a cluster virtual IP addresses on the ISCSI network, the virtual IP addresses may be hosted by any VSA node.
- Core API nodes share a cluster virtual IP addresses on both the INTERNAL-API and EXTERNAL-API networks; the virtual IP addresses may be hosted by any core API node.
- Neutron expects the EXTERNAL-VM network to span all compute nodes and network service nodes for floating IPs and router default SNAT IP addresses.
- The lifecycle-manager provides PXE boot services on CONF.

image

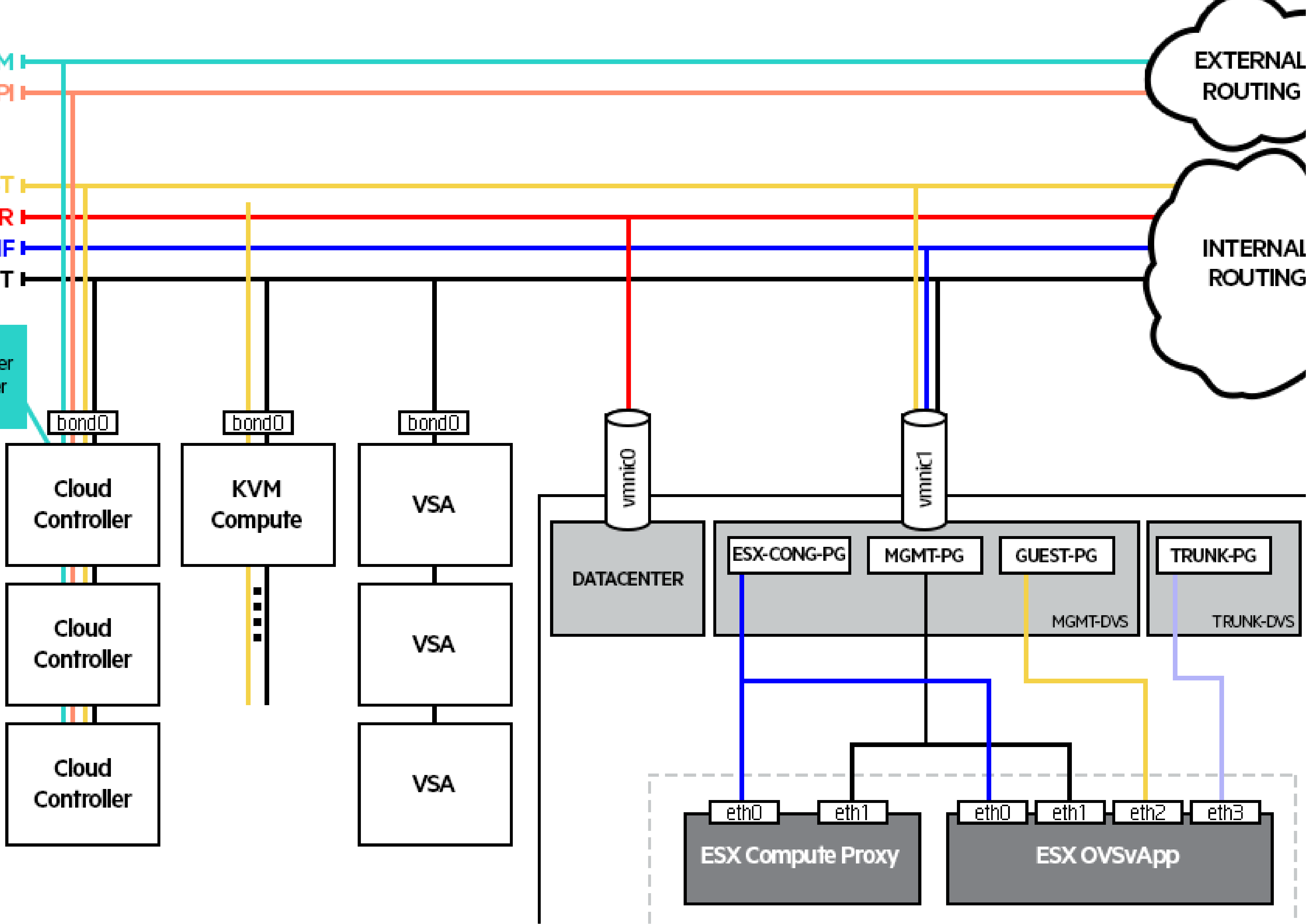
etwork Diagram Template

across controllers is only one possible configuration, and other combinations can also be expressed.

® 5.0: ESX Examples

® 5.0: Entry-scale ESX, KVM with VSA Model

o integrate HPE Helion OpenStack with ESX, KVM with VSA in the same Cloud. The controller configuration is essentially the same as in the Entry-scale KVM with VSA Model example, but the resource node



M
PI
T
R
IF
T
EXTERNAL ROUTING
INTERNAL ROUTING
er
r
bond0
bond0
bond0
Cloud Controller
KVM Compute
VSA
Cloud Controller
VSA
Cloud Controller
VSA
vmnic0
vmnic1
DATACENTER
ESX-CONG-PG
MGMT-PG
GUEST-PG
MGMT-DVS
TRUNK-PG
TRUNK-DVS
eth0
eth1
eth0
eth1
eth2
eth3
ESX Compute Proxy
ESX OVSvApp

onfiguration is also largely the same as the KVM example, with the default GUEST network VXLAN as the Neutron networking model.

etwork (CONF) is required for configuration access from the lifecycle manager. This network must be reachable from the Management network.

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor
ır hardware.

currently supports the following ESXi versions:

te 3)

te 1b)

rements for your vCenter server:

3 and above (It is recommended to run the same server version as the ESXi hosts)

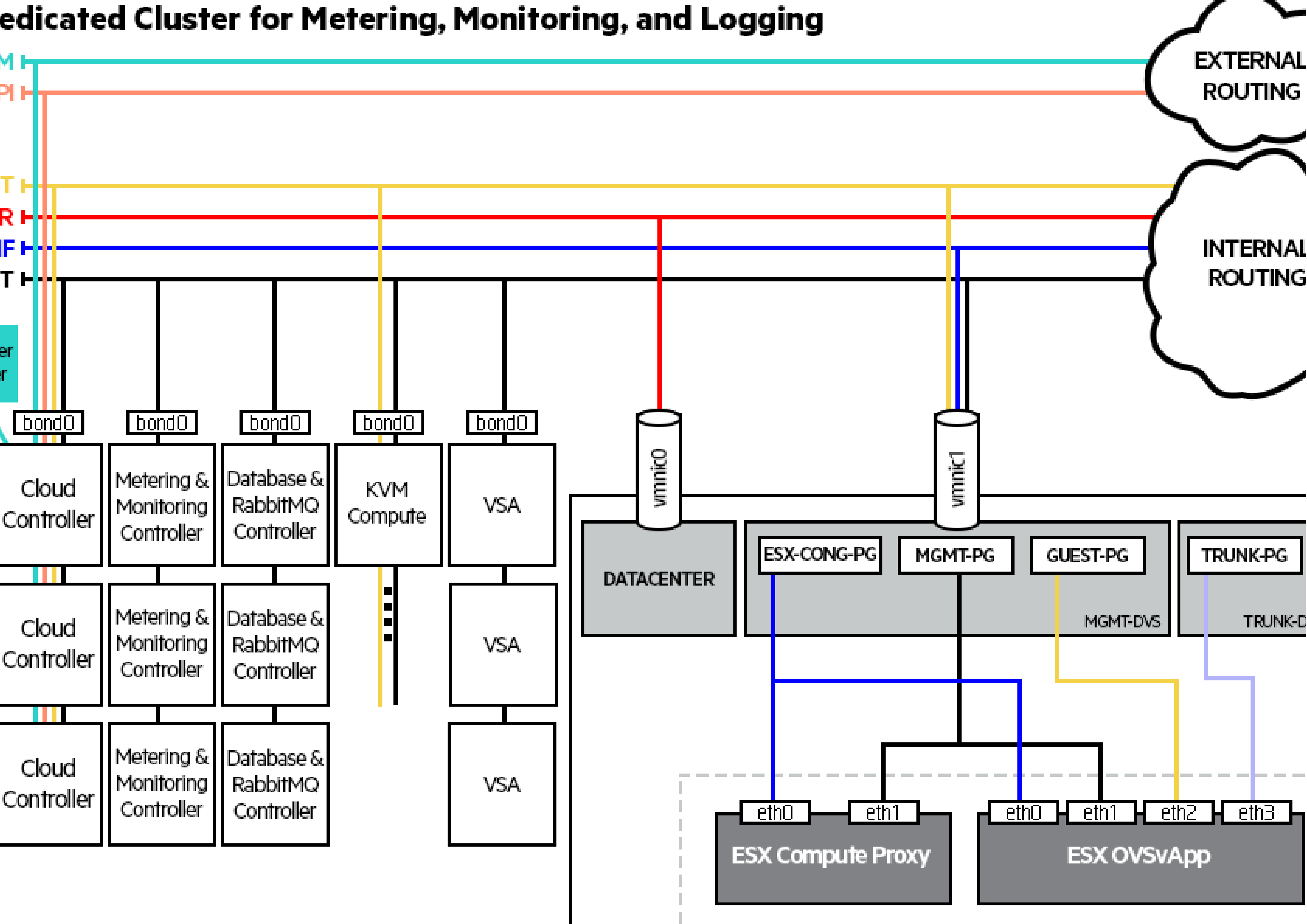
Plus license

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
r)		2	2 X 1 TB (minimum, shared across all nodes)	128 GB (minimum)	2 x 10 Gbit/s +1 NIC (for DC access)	16 C x86_
or)	kvm-compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the V Con
	VSA	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details.	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

5.0: Entry-scale ESX, KVM with VSA Model with Dedicated Cluster for Metering, Monitoring, and Logging

ne Entry-scale ESX KVM with VSA model. It is designed to support greater levels of metering, monitoring, and logging.

equired to support charge-back/show-back for core Infrastructure as a Service (IaaS) elements.
ces at INFO level with the ability to change the settings to DEBUG in order to triage specific error conditions. Minimum retention for logs is 30 days to satisfy audit and compliance requirements.
rmance metrics and health checks for all services.



edicated Cluster for Metering, Monitoring, and Logging
M
PI
T
R
IF
T
EXTERNAL
ROUTING
INTERNAL
ROUTING
bond0
bond0
bond0
bond0
bond0
Cloud Controller
Metering & Monitoring Controller
Database & RabbitMQ Controller
KVM Compute
VSA
Cloud Controller
Metering & Monitoring Controller
Database & RabbitMQ Controller
VSA
Cloud Controller
Metering & Monitoring Controller
Database & RabbitMQ Controller
VSA
vmnic0
vmnic1
DATACENTER
ESX-CONG-PG
MGMT-PG
GUEST-PG
MGMT-DVS
TRUNK-PG
eth0
eth1
eth0
eth1
eth2
eth3
ESX Compute Proxy
ESX OVSvApp

nfiguration is also largely the same as the KVM example, with the default GUEST network VxLAN as the Neutron networking model.

etwork (CONF) is required for configuration access from the lifecycle manager. This network must be reachable from the Management network.

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfor
ur hardware.

currently supports the following ESXi versions:

te 3)

te 1b)

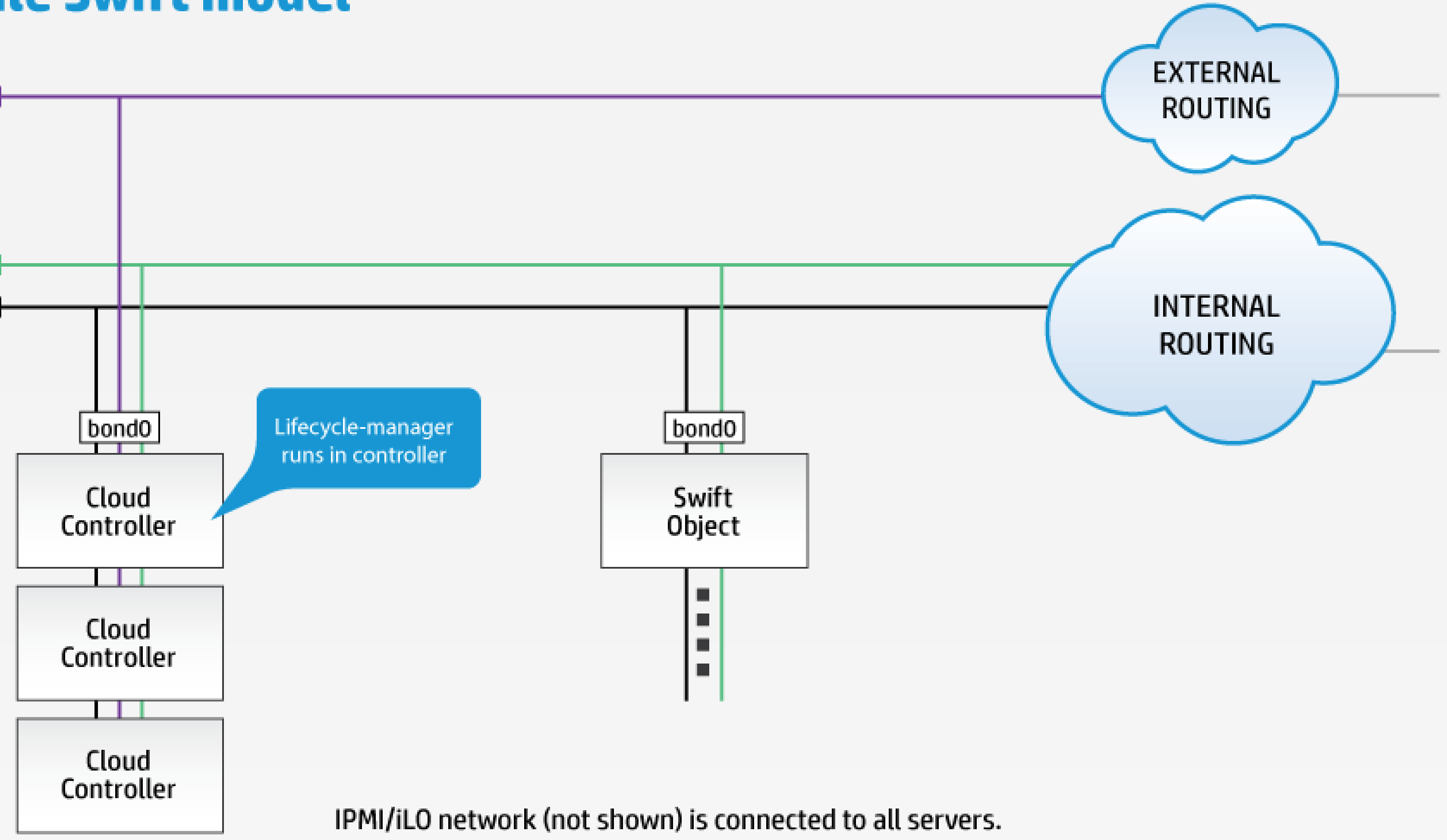
rements for your vCenter server:

3 and above (It is recommended to run the same server version as the ESXi hosts)

Plus license

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Core-API Controller	2	• 1 x 600 GB (minimum) - operating system drive • 2 x 300 GB (minimum) - Swift drive	128 GB	2 x 10 Gbit/s with PXE Support	24 C x86_
	DBMQ Cluster	3	• 1 x 600 GB (minimum) - operating system drive • 1 x 300 GB (minimum) - MySQL drive	96 GB	2 x 10 Gbit/s with PXE Support	24 C x86_
	Metering Mon/Log Cluster	3	• 1 x 600 GB (minimum) - operating system drive	128 GB	2 x 10 Gbit/s with one PXE enabled port	24 C x86_
r)		2 (minimum)	2 X 1 TB (minimum, shared across all nodes)	64 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s +1 NIC (for Data Center access)	16 C x86_
or)	kvm-compute	1-3	2 X 600 GB (minimum)	32 GB (memory must be sized based on the virtual machine instances hosted on the Compute node)	2 x 10 Gbit/s with one PXE enabled port	8 Cl (Inte virtu core the Con
	VSA	0 or 3 (which will provide the recommended redundancy)	3 X 600 GB (minimum) See *Pre-Install Checklist - VSA* for more details	32 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

le Swift model



	VLAN type	Interface
	untagged	IPMI/iLO
	untagged	bond0
	tagged	bond0
	tagged	bond0

Routing Notes:

- IPMI/iLO must be reachable from the lifecycle-manager for operating system install.
- Other networks may be routed as Administrator requires.

the network that will be used for all internal traffic between the cloud services, including node provisioning. This network must be on an untagged VLAN.

onfigured to be presented via a pair of bonded NICs. The example also enables provider VLANs to be configured in Neutron on this interface.

outing" refers to whatever routing you want to provide to allow users to access the External API. "Internal Routing" refers to whatever routing you want to provide to allow administrators to access the Managem

n OpenStack to install the operating system, then an IPMI/iLO network connected to the IPMI/iLO ports of all servers and routable from the lifecycle manager is also required for BIOS and power management c

ers use one disk for the operating system and two disks for Swift proxy and account storage. The Swift object servers use one disk for the operating system and four disks for Swift storage. These values can be m

nums are based on the included *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacity and perfo
ur hardware.

`ft` example runs the Swift proxy, account and container services on the three controller servers. However, it is possible to extend the model to include the Swift proxy, account and container services on dedicate
you are using this model, we have included the recommended Swift proxy servers specs in the table below.

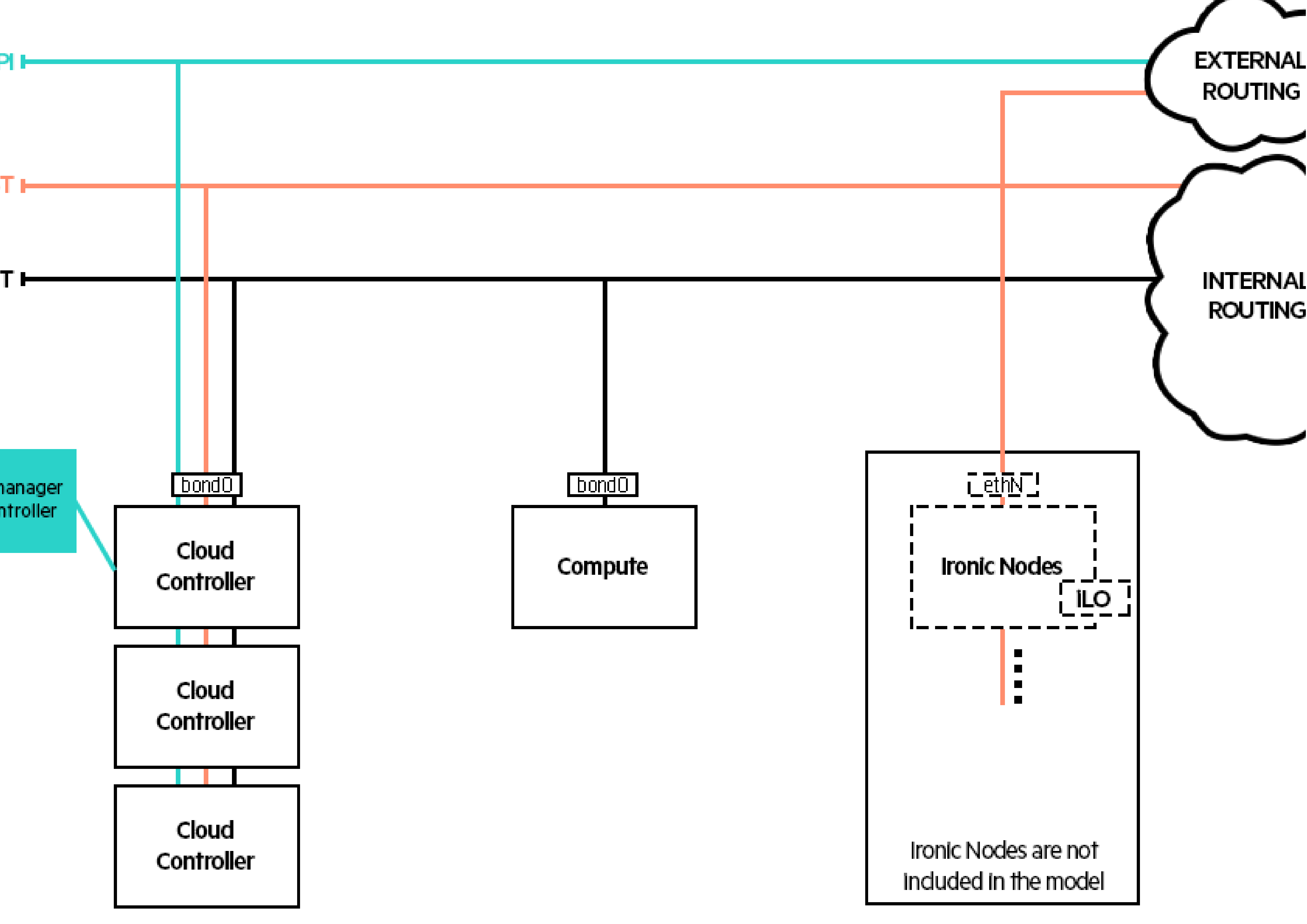
	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Swift account/container data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	swobj	3	If using x3 replication only: • 1 x 600 GB (minimum, see considerations at bottom of page for more details) If using Erasure Codes only or a mix of x3 replication and Erasure Codes: • 6 x 600 GB (minimum, see considerations at bottom of page for more details) **Note:** The disk speeds (RPM) chosen should be consistent within the same ring or storage policy. It's best to not use disks with mixed disk speeds within the same Swift ring.	32 GB (see considerations at bottom of page for more details)	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	swpac	3	2 x 600 GB (minimum, see considerations at bottom of page for more details)	64 GB (see considerations at bottom of page for more details)	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_

wift object and proxy, account, container servers RAM and disk capacity needs

mb is that if you are expecting to have more than a million objects in a container then you should consider using SSDs on the Swift PAC servers rather than HDDs.

® 5.0: Ironic Examples

® 5.0: Entry-scale Cloud with Ironic Flat Network


PI
T
T
EXTERNAL
ROUTING
INTERNAL
ROUTING
anager
ntroller
bond0
Cloud
Controller
Cloud
Controller
Cloud
Controller
bond0
Compute
ethN
Ironic Nodes
iLO
Ironic Nodes are not
included in the model

lO driver, you should ensure that the most recent iLO controller firmware is installed. A recommended minimum for the iLO4 controller is version 2.30.

m hardware requirements are based on the *example configurations* included with the base installation and are suitable only for demo environments. For production systems you will want to consider your capacit
out your hardware.

	Role Name	Required Number	Server Hardware - Minimum Requirements and Recommendations			
			Disk	Memory	Network	
er	Lifecycle-manager	1	300 GB	8 GB	1 x 10 Gbit/s with PXE Support	8 Cl x86_
	Controller	3	• 1 x 600 GB (minimum) - operating system drive • 2 x 600 GB (minimum) - Data drive	64 GB	2 x 10 Gbit/s with one PXE enabled port	8 Cl x86_
	Compute	1	1 X 600 GB (minimum)	16 GB	2 x 10 Gbit/s with one PXE enabled port	16 C x86_

supported network requirements, see *Example Configurations*.

ples of the configuration files for the Entry-scale Cloud with Ironic Flat Network.

```
ol-plane-1
ne-prefix: cp1
: region1
es:



ice-components:
-producer
-agent
-agent

le-manager-target

luster1
-prefix: c1
role: CONTROLLER-ROLE
count: 3
ion-policy: strict
-components:
ecycle-manager
```

```
oitmq
nce-api
nce-registry
nce-client
a-api
a-scheduler-ironic
a-scheduler
a-conductor
a-console-auth
a-novncproxy
a-client
tron-server
tron-ml2-plugin
tron-dhcp-agent
tron-metadata-agent
tron-openvswitch-agent
tron-client
izon
ft-proxy
cached
ft-account
ft-container
ft-object
ft-client
t-api
t-api-cfn
t-api-cloudwatch
t-engine
t-client
nic-api
nic-conductor
nic-client
nstack-client
lometer-api
lometer-polling
lometer-agent-notification
lometer-common
lometer-client
keeper
ka
tica
rm
asca-api
asca-persister
asca-notifier
asca-threshold
asca-client
ging-server
-console-web
-console-monitor
ezer-api
oican-api
oican-client
oican-worker
```

```
a-compute
-client
```

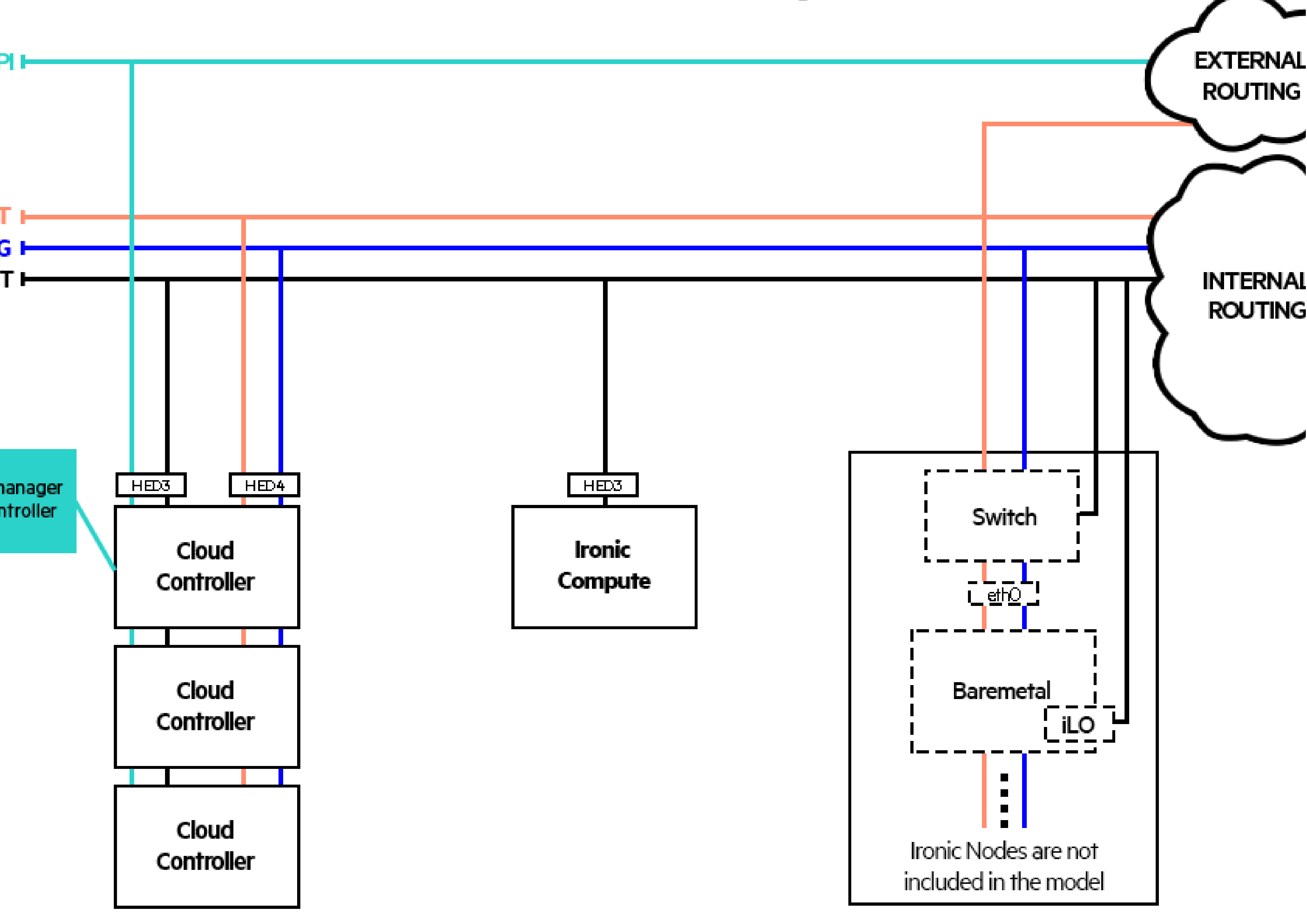
```
e uses the following networks

    CIDR             VLAN
    ----             ----
I   10.0.1.0/24      101 (tagged)
                     102 (tagged)
    192.168.10.0/24  100 (untagged)


as part of Neutron configuration

e values to match your environment

NAL-API-NET
: true

1.0/24
 10.0.1.1
up: EXTERNAL-API
ess: 10.0.1.10
s: 10.0.1.250

-NET
: true

up: GUEST

EMENT-NET
: false

68.10.0/24
 192.168.10.1
up: MANAGEMENT
ess: 192.168.10.10
s: 192.168.10.250
```

```
 network group that users will use to
public API endpoints of your cloud

NAL-API
ffix: extapi

ers:
r: ip-cluster
xtlb
ternal-name is set then public urls in keystone
use this name instead of the IP address.
ust either set this to a name that can be resolved in your network
mment out this line to use IP addresses
l-name:

ponents:
ault

lic
le: my-public-entryscale-ironic-cert
is the name of the certificate that will be used on load balancer.
ce this with name of file in "~helion/my_cloud/config/tls/certs/".
is the certificate that matches your setting for external-name

that it is also possible to have per service certificates:

file:
lt: my-public-entryscale-ironic-cert
on: my-horizon-cert
api: my-nova-cert






 network group that will be used to provide
works to Baremetals


ffix: guest

.networks.flat:
der-physical-network: physnet1



 network group that will be used to for
traffic within the cloud.

ce used by this group will be presented
as physnet1, and used by provider VLANS

EMENT
```

```
ers:
r: ip-cluster
o
nts:
ault

ernal
in
```

```
s:
mples uses hed3 and hed4 as a bonded
all networks on all three server roles

device names and bond options
your environment

OLLER-INTERFACES
erfaces:
OND0

e: bond0
ta:
ions:
 mode: active-backup
 miimon: 200
 primary: hed3
vider: linux
ices:
 name: hed3
 name: hed4
-groups:
ERNAL-API
ST
AGEMENT

TE-IRONIC-INTERFACES
erfaces:
OND0

e: bond0
ta:
ions:
 mode: active-backup
 miimon: 200
 primary: hed3
```

® 5.0: Entry-scale Cloud with Ironic Multi-Tenancy



PI
T
G
T
EXTERNAL
ROUTING
INTERNAL
ROUTING
anager
ntroller
HED3
HED4
HED3
Cloud
Controller
Cloud
Controller
Cloud
Controller
Ironic
Compute
Switch
eth0
Baremetal
iLO
Ironic Nodes are not
included in the model

	Untagged for controllers and compute, needs subnet with IP address range	untagged	• hed3 on controllers and comp
	Tagged for controllers, needs subnet with IP address range. For ironic baremetal nodes, switch config will be set dynamically by Neutron.	neutron provider VLAN (untagged)	• hed4 on controllers • eth0 on baremetal nodes
	Tagged range of VLANs. Number of VLANs in range may be up to number of baremetal nodes (for each node have it's own network). For ironic baremetal nodes, switch config will be set dynamically by Neutron.	neutron provider VLAN (untagged)	• hed4 on controllers • eth0 on baremetal nodes

s to be reachable from TENANT VLANs if ironic instances need to access the cloud APIs
nodes IPMI/iLO must be reachable form lifecycle-manager for operating system install
LO must be reachable from controllers via MANAGEMENT network for operating system install
s must be reachable from controllers MANAGEMENT network for VLAN configuration setting
ld be configured to allow inbound/outbound external access, if external access is needed for Ironic instances

® 5.0: Modifying the Entry-scale KVM with VSA Model for Your Environment

nges that need to be made to the input model to deploy and run this cloud model in your environment.

the perspective of the `entry-scale-kvm-vsa` example, although the same principles apply to all of the examples.

modifications that we will look at:

re the minimum set of changes that you need to make to adapt the examples to run in your environment. These are mostly concerned with networking.
describe more general changes that you can make to your model, e.g. changing disk storage layouts.

the examples use upper case for the object names, but these strings are only used to define the relationships between objects and have no specific significance to the configuration processor. You can change the n
ou do so consistently across the input model.

® 5.0: Localizing the Input Model

imum set of changes needed to localize the cloud for your environment. This assumes you are using other features of the example unchanged:

`l` to specify the network addresses (VLAN IDs and CIDR values) for your cloud.
`s.yml` to specify the PCI bus information for your servers' Ethernet devices.
`ces.yml` to provide network interface configurations, such as bond settings and bond devices.
`ups.yml` to provide the public URL for your cloud and to provide security certificates.
to provide information about your servers.

te specific CIDRs and VLANs for these networks and update these values in the `networks.yml` file. The example models define the following networks:

Network	CIDR	VLAN ID	Tagged / U
	10.0.1.0/24	101	Tagged
	Addresses configured by Neutron, leave blank in the file.	102	Tagged

s shown as untagged. This is required if you are using this network to PXE install the operating system on the cloud nodes.

yml file is shown below. Modify the bolded fields to reflect your site values.

```
 uses the following networks

  CIDR              VLAN
  ----              ----
  10.0.1.0/24       101 (tagged)
  see note 1        102 (tagged)
  10.1.1.0/24       103 (tagged)
  192.168.10.0/24   100 (untagged)


s part of Neutron configuration

 values to match your environment

AL-API-NET

 true
.0/24
10.0.1.1
p: EXTERNAL-API

AL-VM-NET

 true
p: EXTERNAL-VM

NET

 true
.0/24
10.1.1.1
p: GUEST

MENT-NET

 false
8.10.0/24
192.168.10.1
p: MANAGEMENT
```

t names to specific bus slots. Due to inherent race conditions associated with multiple PCI device discovery there is no guarantee that Ethernet devices will be named as expected by the operating system, and it i ent servers with the same physical configuration.

naming pattern, the input model supports an explicit mapping from PCI bus address to a user specified name. HPE Helion OpenStack uses the prefix **hed** (Helion Ethernet Device) to name such devices to avoid rating system.

ngs.yml file is shown below.

```
-name: hed2
imple-port
ress: "0000:08:00.0"

-name: hed3
imple-port
ress: "0000:09:00.0"

-name: hed4
imple-port
ress: "0000:0a:00.0"

ORT-SERVER
rts:
-name: hed3
imple-port
ress: "0000:04:00.0"

-name: hed4
imple-port
ress: "0000:04:00.1"
```

C mappings, representing two different physical server types. The name of each mapping is used as a value in the `servers.yml` file to associate each server with its required mapping. This enables the use of d ware.

ports with the following information:

elion OpenStack uses the form `hedN`.
t types are supported in HPE Helion OpenStack 5.0.
bus address of the port.

e found using the `lspci` command on one of the servers. This command can produce a lot of output, so you can use the following command which will limit the output to list Ethernet class devices only:

```
ep -i net
```

```
grep -i net
rnet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
rnet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
rnet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
rnet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
rnet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
rnet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
```

e the mapping names with the names of your choice and enumerate the ports as required.

ow the network interfaces are to be configured. The example reflects the slightly different configuration of controller, compute nodes, and VSA nodes.

used, this file specifies how bonding is to be set up. It also specifies which networks are to be associated with each interface.

f interfaces `hed3` and `hed4`. You only need to modify this file if you have mapped your physical ports to different names, or if you need to modify the bond options.

```
e: bond0
ta:
ions:
 mode: active-backup
 miimon: 200
 primary: hed3
r: linux
:
ame: hed3
ame: hed4
-groups:
ERNAL-API
ERNAL-VM
ST
AGEMENT
```

ort bonding, then you can modify this specification to specify a non-bonded interface, for example using device `hed3`:

```
LLER-INTERFACES
rfaces:
ed3

: hed3
-groups:
TERNAL-API
TERNAL-VM
EST
NAGEMENT
```

ks groups used in your cloud. A network-group defines the traffic separation model, and all of the properties that are common to the set of L3 networks that carry each type of traffic. They define where services he routing within that model.

ng network groups are defined:

is network group is used for external IP traffic to the cloud. In addition, it defines:

f the load balancer to be used for the external API.
 Security (TLS) attributes.
ating IPs for virtual machines are created on this network group. This is identified by the tag value `neutron.l3_agent.external_network_bridge`.
N traffic is carried on this network group. This is identified by the tag value `neutron.networks.vxlan`.
is is the default network group for traffic between service components in the cloud. In addition, it defines:

ancer is defined on this network group for managing internal and administrative API requests.

le should be left unmodified if you are using the network model defined by the example. More complex modifications are supported but are outside the scope of this document.

l to the external API network are site-specific and need to be modified:

L for the cloud.
 security certificate to use.

roups.yml file is shown below, modify the bolded fields to reflect your site values.

```
fix: extapi

rs:
: ip-cluster
tlb
ernal-name is set then public urls in keystone
se this name instead of the IP address
st either set this to a name that can be resolved
r network
ment out this line to use IP addresses
-name:

onents:
ault

lic
e: my-public-kvm-vsa-cert
```

as follows:

nal name defines how the public URLs will be registered in Keystone. Users of your cloud will need to be able to resolve this URL to access the cloud APIs, and if you are using the TLS, the name must match th

ult to change after initial deployment, this value is left blank in the supplied example which prevents the configuration processor from running until a value has been supplied. If you want to register the public UI nt out this line.

e of the file located in `~/helion/my_cloud/config/tls/certs/` that will be used for your cloud endpoints. As shown above, this can be either a single certificate for all endpoints or a default certificat

not want to use a TLS for the public URLs then change the entry that says `tls-components` to `components`.

ide the details of the physical servers that make up your cloud. There are two sections to this file: `baremetal` and `servers`:

```
se values need to be changed to match your environment.
e network range that contains the ip-addr values for
idual servers listed below.
.168.10.0
5.255.255.0
```

on are used to configure cobbler for operating system installation and must match the network values for the addresses given for the servers.

provides the details of each individual server. For example, here are the details for the first controller:

```
rs
oller1
192.168.10.3
TROLLER-ROLE
oup: RACK1
ng: HP-DL360-4PORT
 b2:72:8d:ac:7c:6f
```

at the system will use for SSH connections to the server for deployment and configuration changes. This address must be in the IP range of one of the networks in the model. In the example, the servers are prov

o an entry in `server_roles.yml` that tells the system how to configure the disks and network interfaces for this server. Roles are also used to define which servers can be used for specific purposes. Adding a for more information, see *HPE Helion OpenStack 5.0 Input Model*.

stem how this server is physically related to networks and other servers. Server groups are used to ensure that servers in a cluster are selected from different physical groups. The example provides a set of server K1, **RACK2**, and **RACK3**. Modifying the server group structure is beyond the scope of this walkthrough - for more information, see *HPE Helion OpenStack 5.0 Input Model*.

f a network port mapping definition (for more information, see *nic_mappings.yml*). You need to set this to the mapping that corresponds to this server.

ress of the interface associated with this server that will be used for PXE boot.

he iLO or IPMI port for this server.

- The login details used to access the iLO or IPMI port of this server. The iLO password value can be provided as an OpenSSL encrypted string. (For instructions on how to generate encrypted passwords, see *Co*

® 5.0: Customizing the Input Model

nal changes that you can make to further adapt the example to your environment:

roller.yml to add additional disk capacity to your controllers.
ml to add additional disk capacity to your VSA servers.
te.yml to add additional disk capacity to your compute servers.

he controllers consists of two sections: a definition of a volume group that provides a number of file-systems for various subsystems, and device-group that provides disk capacity for Swift.

n-vg) is divided into a number of logical volumes that provide separate file systems for the various services that are co-hosted on the controllers in the entry-scale examples. The capacity of each file system is ex city. Because not all file system usage scales linearly, two different disk configurations are provided:

KS - Based on a 512 GB root volume group.
DISKS - Provides a higher percentage of space for the logging service.

ses the smaller disk model. To use the larger disk model you need to modify the `disk-models` parameter in the `server_roles.yml` file, as shown below:

```
NTROLLER-ROLE
e-model: CONTROLLER-INTERFACES
el: CONTROLLER-1TB-DISKS
```

he root volume group, you need to modify the volume group definition in whichever disk model you are using. The following example shows adding an additional disk, `/dev/sdd` to the `disks_controlle`

```
TROLLER-DISKS

oups:
hlm-vg
al-volumes:

OTE: 'sda_root' is a templated value. This value is checked in
```

as a device-group and has a syntax that allows disks to be allocated to specific rings. In the example, two disks are allocated to Swift to be shared by the account, container, and object-0 rings.

```
iftobj

e: /dev/sdb
e: /dev/sdc
 any additional disks for swift here
me: /dev/sdd
me: /dev/sde
:
swift

gs:
- account
- container
      - object-0
```

additional Swift storage, see *HPE Helion OpenStack 5.0: Allocating Disk Drives for Object Storage*.

as a device-group and has a syntax that allows disks to be allocated for data storage or for adaptive optimization (caching). As a best practice, you should use solid state drives for adaptive optimization. The exar
r data and one of adaptive optimization. (For more information, see *VSA with AO or without AO*.)

```
a-data
:
vsa
 data

: /dev/sdc
a-cache
:
vsa
 adaptive-optimization

e: /dev/sdb
```

added by adding more disks to the `vsa-data` device group. Similarly, caching capacity can be increased by adding more high speed storage devices to the `vsa-cache` device group.

ation for compute nodes consists of two volume groups: one for the operating system and one for the ephemeral storage for virtual machines, with one disk allocated to each.

ephemeral storage capacity can be configured by adding additional disks to the `vg-comp` volume group. The following example shows the addition of two more disks, `/dev/sdc` and `/dev/sdd`, to the `disk`

```
p
volumes:
sdb
```

® 5.0: VSA with or without Adaptive Optimization (AO)

n adaptive optimization (AO) or without AO. AO allows built-in storage tiering for VSA. While deploying VSA with or without AO you must ensure to use the appropriate disk input model.

AO, you will have an extra device group section where the usage is identified as adaptive-optimization as described in the following example:

```
can be added if available
groups:
e: vsa-data
sumer:
ame: vsa
sage: data
ices:
 name: /dev/sdc
 /dev/sdd
 /dev/sde
 /dev/sdf

e: vsa-cache
sumer:
ame: vsa
sage: adaptive-optimization
ices:
 name: /dev/sdb
```

of only data disks as described in the following example:

```
can be added if available
ice_groups:
 name: vsa-data
 consumer:
   name: vsa
   usage: data
 devices:
   - name: /dev/sdc
   - name: /dev/sdd
   - name: /dev/sde
   - name: /dev/sdf
```

SD disk for AO.

A node can have a maximum of seven raw disks (excluding the operating system disks) attached to it, which is defined in the disk input model for your VSA nodes. It is expected that no more than seven disks ar
) per VSA node. For example, if you want to deploy VSA with two disks for Adaptive Optimization then your disk input model should not specify more than five raw disks for data and two raw disks for Adaptiv
SA deployment failure.

® 5.0: Creating Multiple VSA Clusters

k 5.0 input model comes with one cluster and three VSA nodes. This is the default configuration available in the input model, but the input model allows you to create multiple VSA clusters of same or different t

pdate in the document means editing the respective YAML files to add or update the configurations/values.

ervers.yml file with a unique name and node_id for each cluster.

ng example we are adding one more cluster. Similarly, you can keep adding clusters based on your requirements.

s.yml file lists six nodes for two clusters:

```
192.168.61.15
-ROLE
oup: RACK1
ng: HP-BL460c-4PORT
0.1.192.232
ord: gone2far
 Administrator
 5C:B9:01:78:8C:B0


192.168.61.16
-ROLE
oup: RACK2
ng: HP-BL460c-4PORT
0.1.192.233
ord: gone2far
 Administrator
 5C:B9:01:78:0E:30


192.168.61.17
-ROLE
oup: RACK3
ng: HP-BL460c-4PORT
0.1.192.234
ord: gone2far
 Administrator
 5C:B9:01:78:2D:00


192.168.62.18
-ROLE-1
oup: RACK1
ng: HP-BL460c-4PORT
0.1.193.232
ord: gone2far
 Administrator
 5C:B9:01:78:8C:B0


192.168.63.19
-ROLE-1
oup: RACK2
ng: HP-BL460c-4PORT
0.1.194.233
ord: gone2far
 Administrator
 5C:B9:01:78:0E:30
```

```
5C:B9:01:78:2D:00
```

he control_plane.yml file with the name, resource-prefix, and server-role.

g control_plane.yml file contains the information of the newly added resource nodes:

```
a
-prefix: vsa
ole: ROLE-VSA
on-policy: strict
t: 0
components:
-client


sa1
e-prefix: vsa1
role: ROLE-VSA-1
ion-policy: strict
nt: 0
-components:
p-client
a
```

e following fields:

	The name assigned for the cluster. In the above example **vsa** and **vsa1**.
	The prefix of that resource cluster.
	The role must be unique for each cluster.

es.yml with new VSA nodes.

ng server_roles.yml file, new VSA nodes are added/updated:

```


E-VSA
-model: INTERFACE_SET_VSA
l: DISK_SET_VSA

E-VSA-1
-model: INTERFACE_SET_VSA
l: DISK_SET_VSA
```

e following fields:

	The name assigned to the cluster. In the above example **vsa** and **vsa1**.
	The type of disk available for the clusters. It can be the same set of disks or a different set of disks. set of disk models is shown (for example:**DISK_SET_VSA**).

hanges to Create Two Cluster with Different Set of Disks

) disk, refer to *VSA with or without Adaptive Optimization (AO)*.

es.yml with new VSA nodes and appropriate disk_set used for that node.

ng servers_roles.yml file you can see both AO and without AO assigned for the node:

```
E-CONTROLLER
-model: INTERFACE_SET_CONTROLLER
l: DISK_SET_CONTROLLER

E-COMPUTE
-model: INTERFACE_SET_COMPUTE
l: DISK_SET_COMPUTE

E-VSA
-model: INTERFACE_SET_VSA
l: DISK_SET_VSA

E-VSA-1
-model: INTERFACE_SET_VSA
l: DISK_SET_VSA_AO
```

ve configured your cloud to have more than one cluster or n-clusters, remember to note down all the cluster IPs.

® 5.0: Configuring a Separate iSCSI Network to use with VSA

cedure to assign a separate iSCSI network to use with VSA nodes. You must configure controller and compute nodes along with VSA to use a separate iSCSI network.

edure to assign a separate iSCSI network:

nanager.

at ~/helion/my_cloud/definition/data to assign a separate iSCSI network to controller nodes, compute nodes, and VSA nodes:

t YAML files need to be changed during the cloud deployment.

Enter the name of the network-group as shown in the example below. In the following example, the name of the network-group is "ISCSI" and this name should remain consistent in other files too.

```
ISCSI
37
n: true
16.13.0/24
: 172.16.13.1
oup: ISCSI
```

s.yml: A new field (forced-network-groups) is added in this file, as shown in the sample below.

```
odels
INTERFACE_SET_CONTROLLER
```

```
 provider: linux
 devices:
   - name: Port0_10G1
   - name: Port1_10G1
twork-groups:
- MGMT
- TENANT
rced-network-groups:
- ISCSI

INTERFACE_SET_COMPUTE
k-interfaces:
me: BOND0
vice:
  name: bond0
nd-data:
 options:
     mode: "802.3ad"
     miinon: 200
 provider: linux
 devices:
   - name: Port0_10G1
   - name: Port1_10G1
twork-groups:
- MGMT
- TENANT
rced-network-groups:
- ISCSI

INTERFACE_SET_VSA
k-interfaces:
me: BOND0
vice:
  name: bond0
nd-data:
 options:
     mode: "802.3ad"
     miinon: 200
 provider: linux
 devices:
   - name: Port0_10G1
   - name: Port1_10G1
twork-groups:
- MGMT
- TENANT
rced-network-groups:
- ISCSI
```

s.yml

```
es
PING
k-groups:

NT
RNAL_API
```

```
ne-suffix: iscsi
ent-endpoints:
a
```

```
.yml
```

```
s.yml
ks:
 the Global networks shared across all the Racks
T_EXTERNAL_API
T_EXTERNAL_VM
T_TENANT
T_MGMT
T_SWIFT
T_ISCSI
```

```
Add Node <name>"
```

rocessor:

```
/ansible
k -i hosts/localhost config-processor-run.yml
```

mand to create a deployment directory.

```
/ansible
k -i hosts/localhost ready-deployment.yml
```

aybook using the command below.

```
sible/next/hos/ansible
k -i hosts/verb_hosts site.yml
```

CSI network is not explicitly configured on the controller nodes then boot from cinder volumes would fail.

® 5.0: Modifying Example Configurations for Object Storage using Swift

ed descriptions about the Swift-specific parts of the input model. For example input models, see *Example Configurations*. For general descriptions of the input model, see *HPE Helion OpenStack 5.0: Networks*. in the `~/helion/my_cloud/definition/data/swift/rings.yml` file.

ls provide most of the data that is required to create a valid input model. However, before you start to deploy, you must do the following:

sed by your nodes and that all disk drives are correctly named and used as described in *Swift Requirements for Device Group Drives*.
artition power for your rings. For more information, see *Ring Specifications*.

d these related pages:

® 5.0: Object Storage using Swift Overview

age (Swift) Service?

Services

d of a number of services:

e API for all requests to the Swift system.
services provide storage management of the accounts and containers.
storage management for object storage.

cated in a number of ways. The following general pattern exists in the example cloud models distributed in HPE Helion OpenStack:

nt, container, and object services run on the same (PACO) node type in the control plane. This is used for smaller clouds or where Swift is a minor element in a larger cloud. This is the model seen in most of the
nt, and container services run on one (PAC) node type in a cluster in a control plane and the object services run on another (OBJ) node type in a resource pool. This deployment model, known as the Entry-Scale
Swift system is in use or planned. See *HPE Helion OpenStack 5.0: Entry-scale Swift Model* for more details.

an be scaled both vertically (nodes with larger or more disks) and horizontally (more Swift storage nodes) to handle an increased number of simultaneous user connections and provide larger storage space.

a number of YAML files in the HPE Helion implementation of the OpenStack Object Storage (Swift) service. For more details on the configuration of the YAML files, see *HPE Helion OpenStack 5.0: Modifyi*

® 5.0: Allocating Proxy, Account, and Container (PAC) Servers for Object Storage

d container (PAC) server is a node that runs the swift-proxy, swift-account and swift-container services. It is used to respond to API requests and to store account and container data. The PAC node does not store

procedure to allocate PAC servers during the **initial** deployment of the system.

servers

s to allocate PAC servers:

put model already contains a suitable server role. The server roles are usually described in the `data/server_roles.yml` file. If the server role is not described, you must add a suitable server role and alloc
g Roles for Swift Nodes and *Allocating Disk Drives.*
put model has assigned a cluster to Swift proxy, account, container servers. It is usually mentioned in the `data/control_plane.yml` file. If the cluster is not assigned, then add a suitable cluster. For instru
(PAC) Cluster.
vers and their IP address and other detailed information.

s to the servers list (usually in the `data/servers.yml` file).
you must also verify and/or modify the server-groups information (usually in `data/server_groups.yml`)

s that is unique to Swift is the allocation of disk drives for use by the account and container rings. For instructions, see *Allocating Disk Drives.*

® 5.0: Allocating Object Servers

ode that runs the swift-object service (**only**) and is used to store object data. It does not run the swift-proxy, swift-account, or swift-container services.

procedure to allocate a Swift object server during the **initial** deployment of the system.

ect Server

s to allocate one or more Swift object servers:

put model already contains a suitable server role. The server roles are usually described in the `data/server_roles.yml` file. If the server role is not described, you must add a suitable server role. For instru
ing a server role for the Swift object server, you will also allocate drives to store object data. For instructions, see *Allocating Disk Drives.*
put model has a resource node assigned to Swift object servers. The resource nodes are usually assigned in the `data/control_plane.yml` file. If it is not assigned, you must add a suitable resource node. F
Nodes.
vers and their IP address and other detailed information. Add the details for the servers in either of the following YAML files and verify the server-groups information:

rvers list (usually in the `data/servers.yml` file).

```
ame>
: <specify-a-name>
ecify-a-name>
```

re defined as follows:

	Specifies a name assigned for the role. In the following example, **SWOBJ-ROLE** is the role name.
	You can either select an existing interface model or create one specifically for Swift object servers. I **SWOBJ-INTERFACES** is used. For more information, see *Swift Network and Service Requirement*
	You can either select an existing model or create one specifically for Swift object servers. In the follo is used. For more information, see *Allocating Disk Drives*.

```
s:



BJ-ROLE
-model: SWOBJ-INTERFACES
l: SWOBJ-DISKS
```

® 5.0: Allocating Disk Drives for Object Storage

he configuration of disk drives and their usage. The examples include several disk models. You must always review the disk devices before making any changes to the existing the disk model. For more informati

owing sections:

wift Disk Model

wift Disk Model

or changing the disk model:

rives available, you can add them to the devices list.
l in the example disk model have different names on your servers. This may be due to different hardware drives. Edit the disk model and change the device names to the correct names.
disk drive than the one listed in the model. For example, if `/dev/sdb` and `/dev/sdc` are slow hard drives and you have SDD drives available in `/dev/sdd` and `/dev/sde`. In this case, delete `/dev/sdb` `dev/sde`.

ves must not contain labels or file systems from a prior usage. For more information, see *Swift Requirements for Device Group Drives*.

```
ring-name>
ring-name>
```

e defined as follows:

	Specifies the service that uses the device group. A `name` field containing **swift** indicates that the driv by Swift.
	Lists the rings that the devices are allocated to. It must contain a `rings` item.
	Contains a list of ring names. In the `rings` list, the `name` field is optional.

rent configurations (patterns) of the proxy, account, container, and object services:

er, and object (PACO) run on same node type.
tainer run on a node type (PAC) and the object services run on a dedicated object server (OBJ).

ervice does not have any rings associated with it.

- proxy, account, container, and object run on the same node type.

```
wift

gs:
ame: account
ame: container
ame: object-0
```

proxy, account, and container run on the same node type.

```
wift

gs:
ame: account
ame: container
```

Dedicated object server. The following example shows two Storage Policies (object-0 and object-1). For more information, see *Designing Storage Policies*.

```
wift

gs:
```

ws a configuration where one drive is used for account and container rings and the other drives are used by the object-0 ring.

```
o



ount
tainer
j

sdc
sde
sdf

t


ame: object-0
```

ul while using logical volumes to store Swift data. The data remains intact during an upgrade, but will be lost if the server is reimaged. If you use logical volumes you must ensure that you only reimage one serv
replicas to be replicated back to the logical volume once the reimage is complete.

ume. To do this, ensure you meet the requirements listed in the table below:

	Do not specify these attributes.
	Specify both of these attributes.
	This attribute must have a `name` field set to **swift**.

Swift logical volumes:

```




ift

s:
me: object-0
me: object-1
```

ust not contain a file system label. For instructions, see *HPE Helion OpenStack 5.0: Verifying a Swift File System Label*.

dy labeled as described above, the `swiftlm-drive-provision` process will assume that the drive has valuable data and will not use or modify the drive.

® 5.0: Creating a Swift Proxy, Account, and Container (PAC) Cluster

r with the server-role `SWPAC-ROLE` there is no need to proceed through these steps.

Proxy, Account, and Container (PAC) Cluster

t proxy, account, and container (PAC) servers, you must identify the control plane and node type/role:

_cloud/definition/data/control_plane.yml file, identify the control plane that the PAC servers are associated with.

type/role used by the Swift PAC servers. In the following example, `server-role` is set to **SWPAC-ROLE**.

usters item in the `control-plane` section.

```
trol-plane-1
lane-prefix: cp1




pac1
prefix: c2
ole: SWPAC-ROLE
ount: 3
on-policy: strict
components:
client
t-ring-builder
t-proxy
t-account
t-container
t-client
```

ease do not change the name of the cluster `swpac` as it may conflict with an existing cluster. A name such as `swpac1`, `swpac2` or `swpac3` would be advisable.

hree servers available that have the `SWPAC-ROLE` assigned to them, you must change `member-count` to match the number of servers.

e four servers with a role of `SWPAC-ROLE`, then the `member-count` should be 4.

s the following service components:

_plane.yml file, identify the control plane that the object servers are associated with.
type/role used by the Swift object servers. In the following example, server-role is set to **SWOBJ-ROLE**:

sources item in the **control-plane**:

```
trol-plane-1
lane-prefix: cp1
me: region1




efix: swobj
: SWOBJ-ROLE
policy: strict
0
ponents:
t
ect
```

res the following service components:

tional; installs the python-swiftclient package on the server.

e a member count attribute. So the number of servers allocated with the **SWOBJ-ROLE** is the number of servers in the data/servers.yml file with a server role of **SWOBJ-ROLE**.

® 5.0: Understanding Swift Network and Service Requirements

s requirements for which service components must exist in the input model and how these relate to the network model. This information is useful if you are creating a cluster or resource node, or when defining th
y options and configurations. For smooth Swift operation, the following must be **true**:

must have a **direct** connection to the same network:

er

lder
rvice must have a **direct** connection to the same network as the cluster-ip service.
e must be configured on a cluster of the control plane. In small deployments, it is convenient to run it on the same cluster as the horizon service. For larger deployments, with many nodes running the swift-pr
oxy and memcached services. The swift-proxy and swift-container services must have a **direct** connection to the same network as the memcached service.
nd swift-ring-builder service must be **co-located** in the same cluster of the control plane.
vice must be **present** on all Swift nodes.

® 5.0: Understanding Swift Ring Specifications

utility as part of the deploy process. (Normally, you will not run the `swift-ring-builder` utility directly.)

ne input model using the **configuration-data** key. The `configuration-data` in the `control-planes` definition is given a name that you will then use in the `swift_config.yml` file. If you have seve
ions can use a shared `configuration-data` object, however it is considered best practice to give each Swift instance its own `configuration-data` object.

E Helion OpenStack 2.x and 3.x

2.x and 3.x, ring specifications were mentioned in the `~/helion/my_cloud/definition/data/swift/rings.yml` file. HPE Helion OpenStack 4.x continues to support ring specifications in that fil
need to make any changes.

he Input Model

ecification is mentioned in the `~/helion/my_cloud/definition/data/swift/swift_config.yml` file. For example:

```
a:
ONFIG-CP1




ne_rings:
es:

r-groups:
Z1

r-groups:
Z2

r-groups:
Z3

 account
ay-name: Account Ring
art-hours: 16
tion-power: 12
cation-policy:
lica-count: 3

 container
ay-name: Container Ring
art-hours: 16
tion-power: 12
cation-policy:
lica-count: 3

 object-0
ay-name: General
lt: yes
art-hours: 16
tion-power: 12
cation-policy:
lica-count: 3
```

s that the rings are specified using the `configuration-data` object **SWIFT-CONFIG-CP1** and has three rings as follows:

st always specify a ring called **account**. The account ring is used by Swift to store metadata about the projects in your system. In Swift, a Keystone project maps to a Swift account. The `display-name` is infor

ion-power, replication-policy and **replica-count** are described in the following section.

neters

ditional replication rings are defined as follows:

Parameter	Description
	Defines the number of copies of object created. Use this to control the degree of resiliency or availability. The `replica-count` is normally set to copies of accounts, containers, or objects). As a best practice, you should not decrease the value to lo higher resiliency, you can increase the value.
	Changes the value used to decide when a given partition can be moved. This is the number of hours that the `swift-ring-builder` tool will enforce between ring rebuil be as low as **1** (one hour). The value can be different for each ring. In the example above, the `swift-ring-builder` will enforce a minimum of 16 hours between ri time is system-dependent so you will be unable to determine the appropriate value for `min-part-h` experience with your system. A value of 0 (zero) is not allowed. In prior releases, this parameter was called `min-part-time`. The older name is still supported, ho `part-hours` and `min-part-time` in the same files.
	The optimal value for this parameter is related to the number of disk drives that you allocate to Swift should use the same drives for both the account and container rings. In this case, the `partition-p` For more information, see *Selecting a Partition Power*.
y	Specifies that a ring uses replicated storage. The duplicate copies of the object are created and stored replicas are identical. If one is lost or corrupted, the system automatically copies one of the remainin replica.
	The default value in the above sample file of ring-specification is set to **yes**, which means that the sto objects. For more information, see *HPE Helion OpenStack 5.0: Designing Storage Policies*.

5.0, Swift supports erasure coded object rings as well as traditional replication rings. Erasure coded rings can be useful for large objects, like backup, video, biotech, i.e. data that is typically written once but read

n-core feature, and as such we recommend working with Professional Services to enable the feature, the use cases have been tested, and are suitable for use with erasure coding.

`-specification` is mentioned in the `~/helion/my_cloud/definition/data/swift/rings.yml` file. A typical erasure coded ring in this file looks like this:

```
C_ring

 16
: 12
policy:
sure_rs_vand
ragments: 10
-fragments: 4
```

	• `jerasure_rs_vand` => Vandermonde Reed-Solomon encoding, based on Jerasure
	This line indicates that the object ring will be of type "erasure coding"
	This indicated the number of data fragments for an object in the ring.
	This indicated the number of parity fragments for an object in the ring.
	The amount of data that will be buffered up before feeding a segment into the encoder/decoder. The d

ed ring, the number of devices in the ring must be greater than or equal to the total number of fragments of an object. For example, if you define an erasure coded ring with 10 data fragments and 4 parity fragmen e ring.

for a PUT object to be successful it must store `ec_ndata + 1` fragment to achieve quorum. Where the number of data fragments (`ec_ndata`) is 10 then at least 11 fragments must be saved for the object PU different drives. To tolerate a single object server going down, say in a system with 3 object servers, each object server must have at least 6 drives assigned to the erasure coded storage policy. So with a single o ining object servers. This allows an object PUT to save 12 fragments, one more than the minimum to achieve quorum.

ne of the erasure coded parameters may be edited after the initial creation. Otherwise there is potential for permanent loss of access to the data.

d expect that an erasure coded configuration that uses a data to parity ratio of 10:4, that the data consumed storing the object is 1.4 times the size of the object just like the x3 replication takes x3 times the size of g, this 10:4 ratio is not correct. The efficiency (ie. how much storage is needed to store the object) is very poor for small objects and improves as the object size grows. However, the improvement is not linear. If will take more space to store than the x3 replication.

ower

object storage system hashes the name. This hash results in a hit on a partition (so a number of different object names result in the same partition number). Generally, the partition is mapped to available disk dri three different disk drives. The hashing algorithm used hashes over a fixed number of partitions. The partition-power attribute determines the number of partitions you have.

istribute the data uniformly across drives in a Swift nodes. It also defines the storage cluster capacity. You must set the partition power value based on the total amount of storage you expect your entire ring to us

n power for a given ring that is appropriate to the number of disk drives you allocate to the ring for the following reasons:

on power and have a few disk drives, each disk drive will have thousands of partitions. With too many partitions, audit and other processes in the Object Storage system cannot walk the partitions in a reasonable

n power and have many disk drives, you will have tens (or maybe only one) partition on a drive. The Object Storage system does not use size when hashing to a partition - it hashes the name.

n a drive, a large partition is cancelled out by a smaller partition so the overall drive usage is similar. However, with very small numbers of partitions, the uneven distribution of sizes can be reflected in uneven di ighboring drive is empty).

ns per drive is 100. If you know the number of drives, select a partition power that will give you approximately 100 partitions per drive. Usually, you install a system with a specific number of drives and add dri the partition power. Hence you must select a value that is a compromise between current and planned capacity.

are installing a small capacity system and you need to grow to a very large capacity but you cannot fit within any of the ranges in the table, please seek help from Professional Services to plan your system.

that can help mitigate the fixed nature of the partition power:

storage represents a small fraction (typically 1 percent) of your object storage needs. Hence, you can select a smaller partition power (relative to object ring partition power) for the account and container rings.

can add additional storage policies (i.e., another object ring). When you have reached capacity in an existing storage policy, you can add a new storage policy with a higher partition power (because you now hav t you can install your system using a small partition power appropriate to a small number of initial disk drives. Later, when you have many disk drives, the new storage policy can have a higher value appropriate

ue to add storage capacity, existing containers will continue to use their original storage policy. Hence, the additional objects must be added to new containers to take advantage of the new storage policy.

elect an appropriate partition power for each ring. The partition power of a ring cannot be changed, so it is important to select an appropriate value. This table is based on a replica count of 3. If your replica coun e, then see *Calculating Numbers of Partitions* for information of selecting a partition power.

n you first deploy Swift, you have a small number of drives (the minimum column in the table), and later you add drives.

r example, if you determine that the maximum number of drives the system will grow to is 40,000, then use a partition power of 17 as listed in the table below. In addition, a minimum of 36 drives is required to
wer.
es that disk drives are the same size. The actual size of a drive is not significant.

f drives during deployment (minimum)	Number of drives in largest anticipated system (maximum)	Recommended partition
	5,000	14
	10,000	15
	40,000	17
	80,000	18
	160,000	19
	300,000	20
	600,000	21
	1,200,00	22
	2,500,000	23
	5,000,000	24

artitions

hashes a given name into a specific partition. For each partition, for a replica count of 3, there are three partition directories. The partition directories are then evenly scattered over all drives. If you are using an
y adding the data fragments and the parity fragments. Using the erasure coded values in the section above this, you would have a replica count of 14 (10 + 4). You can calculate the number of partition directorie

```
directories-per-drive = ( (2 ** partition-power) * replica-count ) / number-of-drives
```

n directories per drive is 100. However, the system can operate normally with a wide range of number of partition directories per drive. The table *Partition Power Matrix* is based on the following:

wer, the minimum number of drives results in approximately 10,000 partition directories per drive. More directories on a drive results in performance issues.
wer, using the maximum number of drives results in approximately 10 partition directories per drive. Using fewer directories per drive results in an uneven distribution of space usage.

priate partition power if your system is a fixed size. Select a value that gives the closest value to 100 partition directories per drive. If your system starts smaller and then grows, the issue is more complicated. Th

is closer to your final anticipated system size - this means that you can use a high partition power that suits your final system.

storage policies as the system grows. These storage policies can have a higher partition power because there will be more drives in a larger system. Note that this does not help account and container rings - stora

® 5.0: Designing Storage Policies

to differentiate the way objects are stored.

cies include the following:

s of disk drive

rives to store various type of data. For example, you can use 7.5K RPM high-capacity drives for one type of data and fast SSD drives for another type of data.
availability needs

nented on a per-container basis. If you want a non-default storage policy to be used for a new container, you can explicitly specify the storage policy to use when you create the container. You can change which s
ct existing containers. Once the storage policy of a container is set, the policy for that container cannot be changed.

rage policies can overlap or be distinct. If the storage policies overlap (i.e., have disks in common between two storage policies), it is recommended to use the same set of disk drives for both policies. But in the c
use one storage policy receives many objects, the drives that are common to both policies must store more objects than drives that are only allocated to one storage policy. This can be appropriate for a situation w
rlapped drives.

icies

storage policies are specified in the input model:

- age policy is specified in ring-specification in the `data/swift/rings.yml` file for a given region.
- drives with specific rings in a disk model. This specifies which drives and nodes use the storage policy. In other word words, where data associated with a storage policy is stored.

d similar to other rings. However, the following features are unique to storage policies:

- licable to object rings only. The account or container rings cannot have storage policies.
- e ring name: object-<index>, where index is a number in the range 0 to 9 (in this release). For example: object-0.
- always be specified.
- deployed, it should never be deleted. You can remove all disk drives for the storage policy, however the ring specification itself cannot be deleted.
- `ay-name` attribute when creating a container to indicate which storage policy you want to use for that container.
- ies can be the default policy. If you do not specify the storage policy then the object created in new container uses the default storage policy.
- lt, only containers created later will have that changed default policy.

ws three storage policies in use. Note that the third storage policy example is an erasure coded ring.

```



eneral

 16
: 12
icy:
nt: 3

ata

 16
: 20
icy:
nt: 3

rchive

 16
: 20
olicy:
sure_rs_vand
ragments: 10
-fragments: 4
ment-size: 1048576
```

allows you to control the placement of replicas on different groups of servers. When constructing rings and allocating replicas to specific disk drives, Swift will, where possible, allocate replicas using the follow
eved by avoiding single points of failure:

- plica on a different disk drive within the same server.
- plica on a different server.
- plica in a different Swift zone.

d a replica count of three, it is easy for Swift to place each replica on a different server. If you only have two servers though, Swift will place two replicas on one server (different drives on the server) and one co

re is no need to use the Swift zone concept. However, if you have more servers than your replica count, the Swift zone concept can be used to control the degree of resiliency. The following table shows how data
re scenarios. In all cases, a replica count of three is assumed and that there are a total of six servers.

er of Swift Zones	Replica Placement	Failure Scenarios	Det
e zone)	Replicas are placed on different servers. For any given object, you have no control over which servers the replicas are placed on.	One server fails	You are guaranteed that there are
		Two servers fail	You are guaranteed that there is o
		Three servers fail	1/3 of the objects cannot be acces three replicas.
Swift zone)	Half the objects have two replicas in Swift zone 1 with one replica in Swift zone 2. The other objects are reversed, with one replica in Swift zone 1 and two replicas in Swift zone 2.	One Swift zone fails	You are guaranteed to have at lea have two remaining replicas and replica.
Swift zone)	Each zone contains a replica. For any given object, there is a replica in each Swift zone.	One Swift zone fails	You are guaranteed to have two r
		Two Swift zones fail	You are guaranteed to have one r

w examples of how to specify the Swift zones in your input model.

o Specify Swift Zones

the ring specifications using the server group concept. To define a Swift zone, you specify:

- zone number
- er groups

n your input model. The example input models typically define a number of server groups. You can use these pre-defined server groups or create your own.

three models use the example server groups `CLOUD`, `AZ1`, `AZ2` and `AZ3`. Each of these examples achieves the same effect – creating a single Swift zone.

```
ons:
on1


roups:
D
```

```
ring-specifications:
  - region: region1
    swift-zones:
     - id: 1
       server-groups:
         - AZ1
         - AZ2
```

```
server-groups:
  - name: ZONE_ONE
    server-groups:
      - AZ1
      - AZ2
      - AZ3
ring-specifications:
```

he swift-zones specification, a single Swift zone is used by default for all servers.

hree Swift zones are specified and mapped to the same availability zones that Nova uses (assuming you are using one of the example input models):

```
ns:
n1


oups:


oups:


oups:
```

datacenter with four availability zones which are mapped to two Swift zones. This type of setup may be used if you had two buildings where each building has a duplicated network infrastructure:

```
ns:
n1


oups:



oups:
```

at Ring Level

same Swift zone layout for all rings in your system. However, it is possible to specify a different layout for a given ring. The following example shows that the account, container and object-0 rings have two zo

```
ns:
n1


oups:


oups:


ount

tainer

ect-0

ect-1
```

5.6. Customizing Swift Service Configuration Files

enables you to modify various Swift service configuration files. The following Swift service configuration files are located on the lifecycle manager in the `~/helion/my_cloud/config/swift/` director

- onf.j2
- iler.conf.j2
- .conf.j2
- ealms.conf.j2
- onf.j2
- nf.j2
- f.j2

on options that can be set or changed, including **container rate limit** and **logging level**:

ainer Rate Limit

nit allows you to limit the number of **PUT** and **DELETE** requests of an object based on the number of objects in a container. For example, suppose the **container_ratelimit_x = r** . It means that for containers of

niting:

nanager.

ction of `~/helion/my_cloud/config/swift/proxy-server.conf.j2`:

```
imit_0 = 100
imit_1000000 = 100
imit_5000000 = 50
```

nd **DELETE** object rate limit to 100 requests per second for containers with up to 1,000,000 objects. Also, the **PUT** and **DELETE** rate for containers with between 1,000,000 and 5,000,000 objects will vary line he container object count increases.

o git:

```
/ansible

<commit message>"
```

rocessor:

```
/ansible
k -i hosts/localhost config-processor-run.yml
```

rectory:

```
/ansible
k -i hosts/localhost ready-deployment.yml
```

`nfigure.yml` playbook to reconfigure the Swift servers:

actice, do not set the log level to DEBUG for a long period of time. Use it for troubleshooting issues and then change it back to INFO.

s to set the logging level of the `account-server` to **DEBUG**:

nanager.

ction of `~/helion/my_cloud/config/swift/account-server.conf.j2`:

```
UG
```

o git:

```
/ansible

 <commit message>"
```

rocessor:

```
/ansible
k -i hosts/localhost config-processor-run.yml
```

rectory:

```
/ansible
k -i hosts/localhost ready-deployment.yml
```

`nfigure.yml` playbook to reconfigure the Swift servers:

```
sible/next/hos/ansible
k -i hosts/verb_hosts swift-reconfigure.yml
```

Centralized Logging Service.

® 5.0: Alternative Configurations

5.0 there are alternative configurations that we recommend for specific purposes

- *Ceph Model with One Network*
- *Ceph Model with Two Networks*
- *ecycle-Manager Node*
- *n OpenStack without DVR*
- *n OpenStack with Provider VLANs and Physical Routers Only*
- *nstalling Two Systems on One Subnet*

® 5.0: SLES Compute Nodes

```
UTE-INTERFACES
es:
```

```
ary: hed1
: linux

me: hed1
me: hed2
ps:
-VM

NT
```

```
1
.13.111.15
COMPUTE-ROLE
p: RACK1
: DL360p_G8_2Port
c:b1:d7:77:d0:b0
12.13.14
d: *********
dministrator
sles12sp2-x86_64
```

```
UTE-ROLE
: SLES-COMPUTE-INTERFACES
S-COMPUTE-DISKS
```

```
MPUTE-DISKS
:
-vg
volumes:
da_root

olumes:
icy is not to consume 100% of the space of each volume group.
ld be left free for snapshots and to allow for some flexibility.
 root
 35%
e: ext4
: /
 log
 50%
: /var/log
e: ext4
opts: -O large_file
 crash
 10%
: /var/crash
```

```
olumes:
 compute
 95%
: /var/lib/nova
e: ext4
opts: -O large_file
```

```
ol-plane-1
ne-prefix: cp1
: region1


les-compute
e-prefix: sles-comp
role: SLES-COMPUTE-ROLE
ion-policy: any
nt: 1
-components:
-client
a-compute
a-compute-kvm
tron-l3-agent
tron-metadata-agent
tron-openvswitch-agent
tron-lbaasv2-agent
```

® 5.0: RHEL Compute Nodes

```
UTE-INTERFACES
es:


nd0


: active-backup
on: 200
ary: hed1
: linux

me: hed1
me: hed2
ps:
-VM

NT
```

```
c:b1:d7:77:d0:b0
12.13.14
d: *********
dministrator
rhel72-x86_64
```

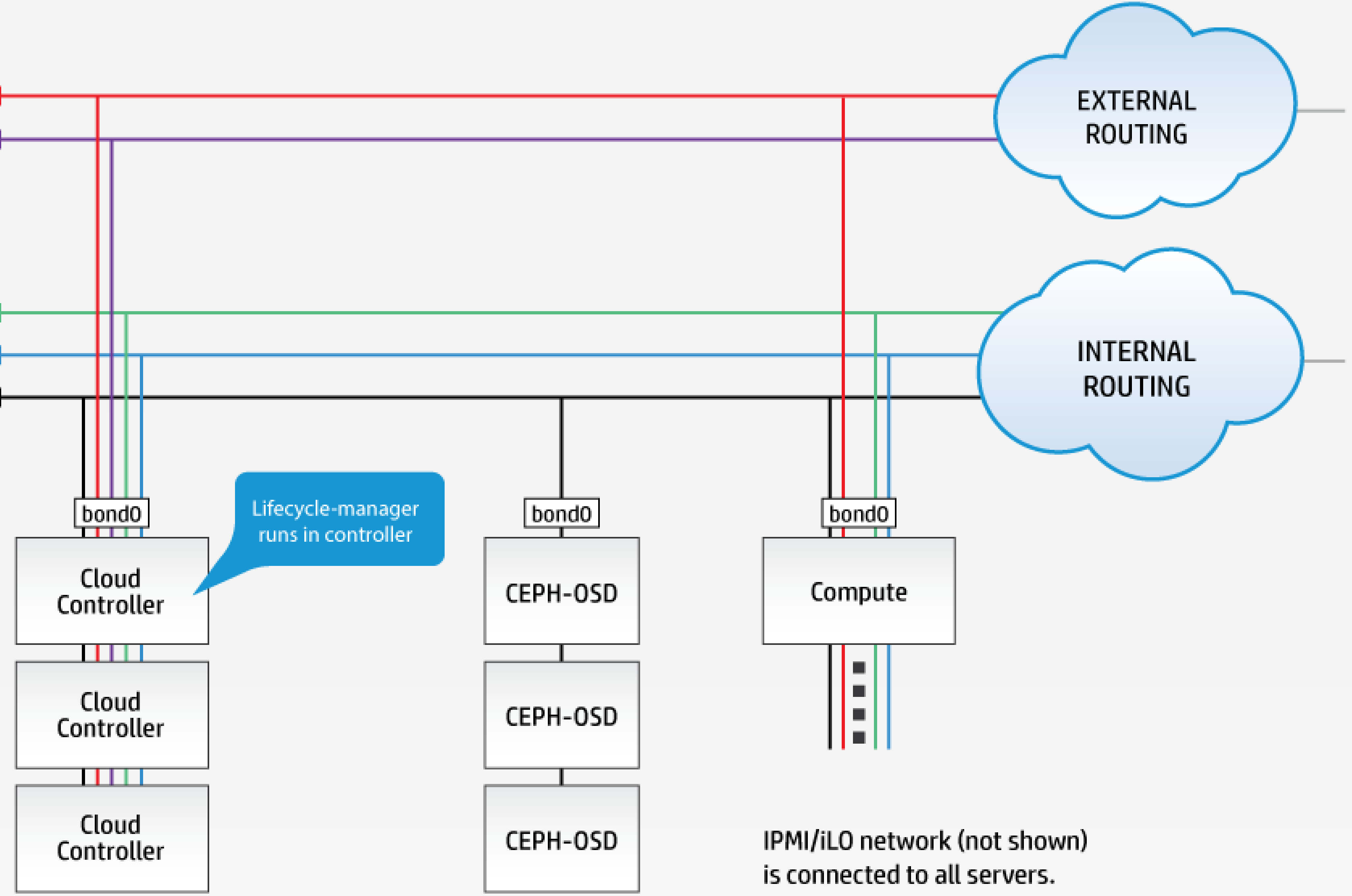
```
UTE-ROLE
: RHEL-COMPUTE-INTERFACES
L-COMPUTE-DISKS
```

```
MPUTE-DISKS
:
-vg
volumes:
da_root

olumes:
icy is not to consume 100% of the space of each volume group.
ld be left free for snapshots and to allow for some flexibility.
 root
 35%
e: ext4
: /
 log
 50%
: /var/log
e: ext4
opts: -O large_file
 crash
 10%
: /var/crash
e: ext4
opts: -O large_file

comp
 is dedicated to Nova Compute to keep VM IOPS off the OS disk
volumes:
sdb
olumes:
 compute
 95%
: /var/lib/nova
e: ext4
opts: -O large_file
```

```
role: RHEL-COMPUTE-ROLE
ion-policy: any
nt: 1
-components:
-client
a-compute
a-compute-kvm
tron-l3-agent
tron-metadata-agent
tron-openvswitch-agent
tron-lbaasv2-agent
```

® 5.0: Entry-scale KVM with Ceph Model

Ceph can be altered to use a single-network model:



	VLAN type	Interface
	untagged	IPMI/iLO
	untagged	bond0
	tagged	bond0
	tagged	bond0
	tagged	bond0

Routing Notes:

- EXTERNAL-API must be reachable from EXTERNAL-VM.
- IPMI/iLO must be reachable from the lifecycle-manager for operating system install.
- Other networks may be routed as Administrator requires.

ph is a unified storage system for various storage use cases for an OpenStack-based cloud. It is highly reliable, easy to manage, and horizontally scalable as demand grows.

OSD daemons for storage operations instead of client routing the request to a specific gateway as is commonly found in other storage solutions. OSD daemons perform data replication and participate in recover count of three, causing daemons to transact three times the amount of client data over the cluster network. So, every 4 MB of write data is likely to result in 12 MB of data movement across Ceph clusters. Consi n data traffic, which can be primarily categorized into three segments:

primarily includes all admin related operations such as pool creation, crush map modification, user creation, etc.

primarily includes client requests sent to OSD daemons.

raffic - primarily includes replication and recovery data traffic among OSD daemons.

er, the network configuration is important. Segregating the data traffic using multiple networks allows for this. For medium-size production environments we recommend to have a cluster with at least two netwo de) network. For larger production environments we recommend that you segregate all three network traffic types by utilizing three networks. This particular document shows you how to setup two networks but

provides additional security as well because your cluster network does not need to be connected to the internet directly. This helps in preventing spoof attacks and allows the OSD daemons to keep communicati ought to active + clean state whenever required.

he Entry-scale KVM with Ceph model. It is designed with two VLANs: a public (front-side) network and a cluster (back-side) network. This enables more options in regards to scaling.

ing components:

one KVM compute node, and three Ceph OSD nodes.

ponent of the Ceph cluster is deployed on the controller nodes along with other OpenStack service components. This limits your cloud to three monitor nodes which should be suitable for most production enviro

. management VLAN and OSD VLAN) which segregates Ceph client traffic from Ceph cluster traffic. The management network will be used to carry cloud management data, such as RabbitMQ, HOPS, and da on, as well as client data traffic, such as cinder-volume writing blocks to Ceph storage pools. The Ceph cluster network will be dedicated for OSD daemons and will be used to carry replication traffic.

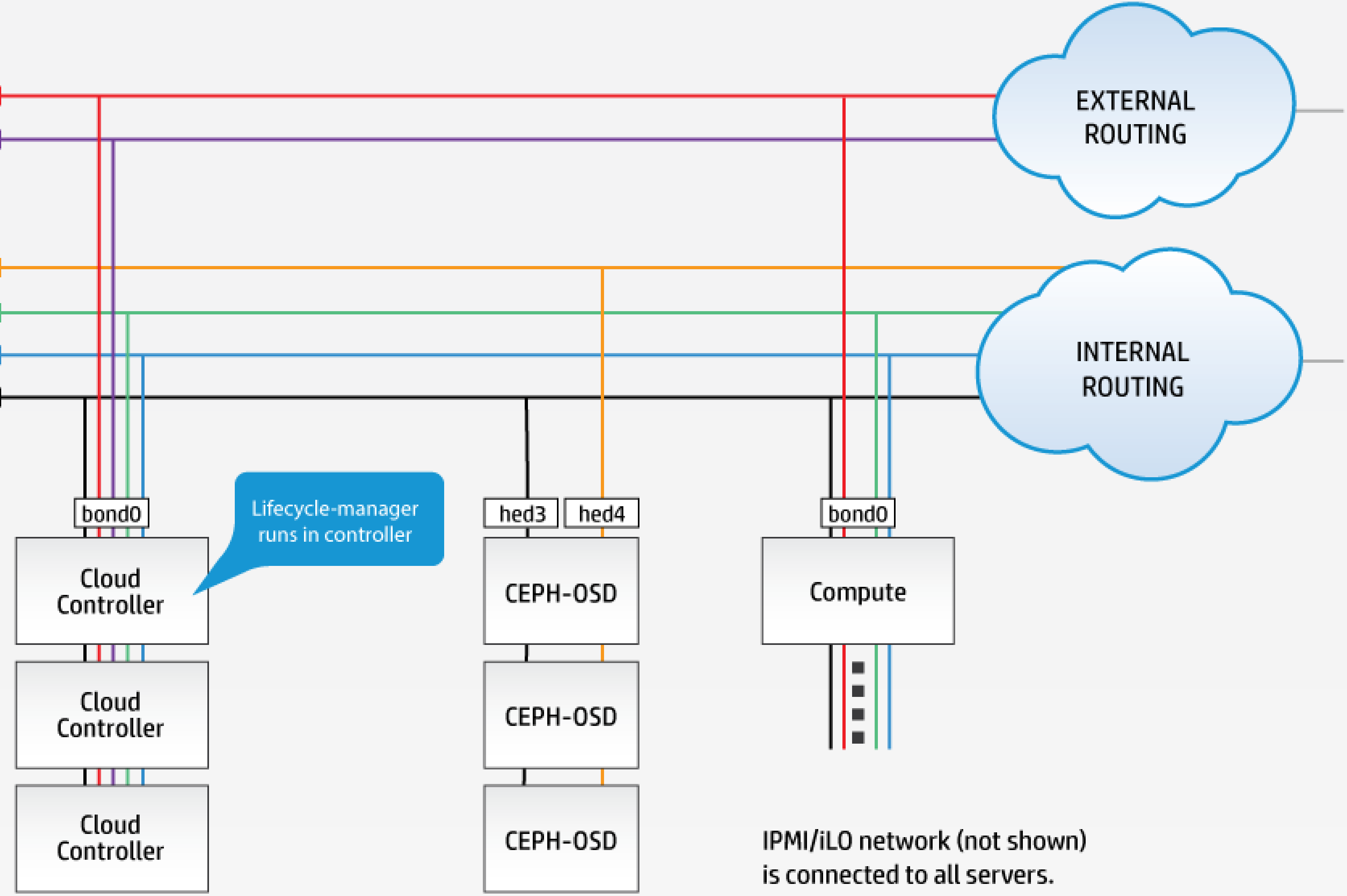
is initially provided with this example configuration. If additional compute capacity is required then further compute nodes can be added to the configuration by adding more nodes to the compute resource plane

les are initially provided with this example configuration. If additional OSD capacity is required then further OSD nodes can be added to the configuration by adding more nodes to the OSD resource plane.

e key characteristics needed per server role for this configuration.

Server role	Quantity	Compute Requirement	Network Re
	3	2x 10 core 2.66 GHz 96 - 128 GB RAM	2x 10Gb Dual Port NIC
or)	1 (minimum)	2x 12 core 2.66 GHz (ES-2690v3) Intel Xeon 256 GB RAM	1x 10Gb Dual Port NIC
	3 (minimum)	RAM is dependent upon the number of disks. 1 GB per TB of disk capacity is recommended.	1x 10Gb Dual Port NIC

tes the physical networking used in this configuration.



	VLAN type	Interface
	untagged	IPMI/iLO
	untagged	bond0
	untagged	hed4
	tagged	bond0
	tagged	bond0

Routing Notes:

- EXTERNAL-API must be reachable from EXTERNAL-VM.
- IPMI/iLO must be reachable from the lifecycle-manager for operating system install.
- Other networks may be routed as Administrator requires.

ked up beforehand.

server NIC interfaces are correctly specified in the ~/helion/my_cloud/definition/data/nic_mappings.yml file and that they meet the server requirements.

tes in-line:

```
ion for controller nodes.  A bonded interface is
anagement network.
4PORT
s:
ame: hed1
ple-port
ss: "0000:07:00.0"

ame: hed2
ple-port
ss: "0000:08:00.0"

ame: hed3
ple-port
ss: "0000:09:00.0"

ame: hed4
ple-port
ss: "0000:0a:00.0"

ion for compute and OSD nodes should be

T-SERVER
s:
ame: hed3
ple-port
ss: "0000:04:00.0"

ame: hed4
ple-port
ss: "0000:04:00.1"
```

or your OSD interfaces in the ~/helion/my_cloud/definition/data/net_interfaces.yml file.

NIC is configured to both the Management and OSD network groups, indicated below:

```
FACES
es:



3
ps:
ENT
```

oup in the ~/helion/my_cloud/definition/data/network_groups.yml file:

```
work group that will be used for
c of cluster among OSDs.


: osd

ints:
ternal
```

he ~/helion/my_cloud/definition/data/networks.yml file:

```
lse
24
0.1.1
OSD
```

e server groups in the ~/helion/my_cloud/definition/data/server_groups.yml file, indicated by the bold portion below:

```
-NET
NET

ET
```

the ~/helion/my_cloud/definition/data/firewall_rules.yml file to allow OSD nodes to be pingable via the OSD network, indicated by the bold portion below:

```
```

```
n: 8

x: 0
p
```

and README.md Files

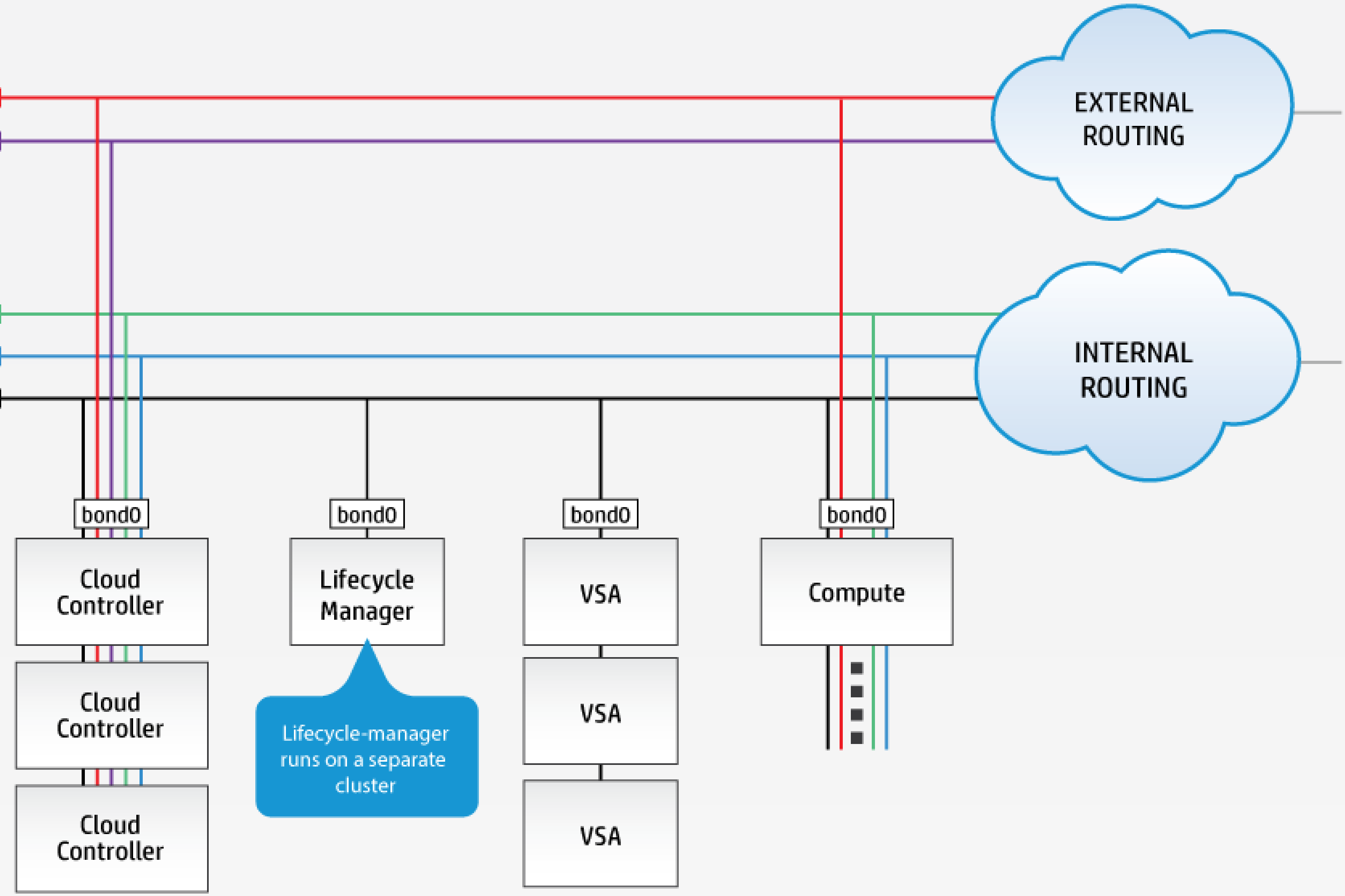
n/my_cloud/definition/README.html and ~/helion/my_cloud/definition/README.md files to reflect the OSD network group information if you wish. This change does not have any se
of your model.

® 5.0: Using a Dedicated Lifecycle Manager Node

rations included host the lifecycle manager on the first controller nodes. It is also possible to deploy this service on a dedicated node. A typical use case for wanting to run the dedicated lifecycle manager is to be
hout having to re-install the first server. Some administrators might also prefer the additional security of keeping all of the configuration data on a separate server from those that users of the cloud connect to (alt
n be password protected).

ntation of what this setup would look like:

with dedicated lifecycle-manager and shared os-install network



IPMI/iLO network (not shown) is connected to all servers.

	VLAN type	Interface
	untagged	IPMI/iLO
	untagged	bond0
	tagged	bond0

Routing Notes:

- EXTERNAL-API must be reachable from EXTERNAL-VM.
- IPMI/iLO must be reachable from the lifecycle-manager for operating system install.

ted lifecycle manager in your input model, make the following edits to your configuration files.

dentation of each of the input files is important and will cause errors if not done correctly. Use the existing content in each of these files as a reference when adding additional content for your lifecycle manager.

yml to add the lifecycle manager.
nl to add the lifecycle manager role.
yml to add the interface definition for the lifecycle manager.
_manager.yml file to define the disk layout for the lifecycle manager.
add the dedicated lifecycle manager node.

he addition of a single node cluster into the control plane to host the lifecycle manager service. Note that, in addition to adding the new cluster, you also have to remove the lifecycle manager component from the

```
ter0
efix: c0
e: LIFECYCLE-MANAGER-ROLE
nt: 1
-policy: strict
mponents:
cle-manager
ient
ter1
efix: c1
e: CONTROLLER-ROLE
nt: 3
-policy: strict
mponents:
rver
```

e of role `LIFECYCLE-MANAGER-ROLE` hosting the lifecycle manager.

he insertion of the new server roles definition:

```
ECYCLE-MANAGER-ROLE
-model: LIFECYCLE-MANAGER-INTERFACES
l: LIFECYCLE-MANAGER-DISKS

TROLLER-ROLE
```

ole which references a new interface-model and disk-model to be used when configuring the server.

he insertion of the network-interface info:

```
mode: active-backup
miimon: 200
primary: hed3
ider: linux
ces:
- name: hed3
- name: hed4
-groups:
NAGEMENT
```

r uses the same physical networking layout as the other servers in the example. For details on how to modify this to match your configuration, see *net_interfaces.yml*.

er.yml

ls are provided as separate files (this is just a convention, not a limitation) so the following should be added as a new file named `disks_lifecycle_manager.yml`:

```
CLE-MANAGER-DISKS
 to be used for Lifecycle Managers nodes
oot is used as a volume group for /, /var/log and /var/crash
s a templated value to align with whatever partition is really used
 is checked in os config and replaced by the partition actually used
. sda1 or sda5

s:
m-vg
-volumes:
/sda_root

lumes:
cy is not to consume 100% of the space of each volume group.
d be left free for snapshots and to allow for some flexibility.
 root
 80%
e: ext4
: /
 crash
 15%
: /var/crash
e: ext4
opts: -O large_file

e: os
```

e insertion of an additional server used for hosting the lifecycle manager. Provide the address information here for the server you are running on, i.e., the node where you have installed the HPE Helion OpenSta

```
CYCLE-MANAGER-ROLE
up: RACK1
g: HP-SL230-4PORT
8c:dc:d4:b5:c9:e0
ormation is not needed

s
ller1
92.168.10.3
ROLLER-ROLE
```

® 5.0: Configuring HPE Helion OpenStack without DVR

n OpenStack without DVR

del, the Neutron service utilizes distributed routing (DVR). This is the recommended setup because it allows for high availability. However, if you would like to disable this feature, here are the steps to achieve t

make the following changes:

_cloud/config/neutron/neutron.conf.j2 file, change the line below from:

```
ted = {{ router_distributed }}
```

```
ted = False
```

_cloud/config/neutron/ml2_conf.ini.j2 file, change the line below from:

```
ted_routing = True
```

```
ted_routing = False
```

_cloud/config/neutron/l3_agent.ini.j2 file, change the line below from:

```
 neutron_l3_agent_mode }}
```

```
gacy
```

_cloud/definition/data/control_plane.yml file, remove the following values from the Compute resource service-components list:

```

-agent
tadata-agent
```

ou fail to remove the above values from the Compute resource service-components list from file ~/helion/my_cloud/definition/data/control_plane.yml, you will end up with routers (non_D
st, even though the lifecycle manager is configured for non_distributed routers.

o your local git repository:

```
/ansible
k -i hosts/localhost config-processor-run.yml
```

ent playbook:

```
/ansible
k -i hosts/localhost ready-deployment.yml
```

Iore information on cloud deployments are available in the *HPE Helion OpenStack 5.0: Cloud Installation Overview*

® 5.0: Configuring HPE Helion OpenStack with Provider VLANs and Physical Routers Only

ring Neutron is to use provider VLANs and physical routers only, here are the steps to achieve this.

make the following changes:

_cloud/config/neutron/neutron.conf.j2 file, change the line below from:

```
ted = {{ router_distributed }}
```

```
ted = False
```

_cloud/config/neutron/ml2_conf.ini.j2 file, change the line below from:

```
ted_routing = True
```

```
ted_routing = False
```

_cloud/config/neutron/dhcp_agent.ini.j2 file, change the line below from:

```
_metadata = {{ neutron_enable_isolated_metadata }}
```

```
_metadata = True
```

_cloud/definition/data/control_plane.yml file, remove the following values from the Compute resource `service-components` list:

```

agent
adata-agent
```

® 5.0: Considerations When Installing Two Systems on One Subnet

eparate HPE Helion OpenStack 5.0 systems using a single subnet, you will need to consider the following notes.

includes the `keepalived` daemon which maintains virtual IPs (VIPs) on cluster nodes. In order to maintain VIPs, it communicates between cluster nodes over the VRRP protocol.

entifies a particular VRRP cluster and must be unique for a subnet. If you have two VRRP clusters with the same virtual routerid, causing a clash of VRRP traffic, the VIPs are unlikely to be up or pingable and y
eepalived/keepalived.log:

ecommendation is to install your separate HPE Helion OpenStack 5.0 systems with VRRP traffic on different subnets.

may also assign a unique routerid to your separate HPE Helion OpenStack 5.0 system by changing the `keepalived_vrrp_offset` service configurable. The routerid is currently derived using the `keepali`
processor variable and the `keepalived_vrrp_offset`.

manager.

`my_cloud/config/keepalived/defaults.yml` file and change the value of the following line:

```
_offset: 0
```

a number that uniquely identifies a separate vrrp cluster. For example:

`offset: 0` for the 1st vrrp cluster on this subnet.

`offset: 1` for the 2nd vrrp cluster on this subnet.

`offset: 2` for the 3rd vrrp cluster on this subnet.

ou should be aware that the files in the `~/helion/my_cloud/config/` directory are symlinks to the `~/helion/hos/ansible/` directory. For example, `~/helion/my_cloud/config/keepal`
o `~/helion/hos/ansible/roles/keepalived/defaults/main.yml`

```
helion/my_cloud/config/keepalived/defaults.yml
k 1 stack stack 55 May 24 20:38 /home/stack/helion/my_cloud/config/keepalived/defaults.yml -> ../../../hos/ansible/roles/keepalived/defaults/
```

a tool like `sed` to make edits to files in this directory, you might break the symbolic link and create a new copy of the file. To maintain the link, you will need to force `sed` to follow the link:

```
follow-symlinks 's$keepalived_vrrp_offset: 0$keepalived_vrrp_offset: 2$' ~/helion/my_cloud/config/keepalived/defaults.yml
```

ou could directly edit the target of the link `~/helion/hos/ansible/roles/keepalived/defaults/main.yml`.

tion to the *local git repo*, as follows:

```
/ansible

changing Admin password"
```

rocessor with this command:

```
/ansible
k -i hosts/localhost config-processor-run.yml
```

to create a deployment directory:

```
/ansible
k -i hosts/localhost ready-deployment.yml
```

hange after your initial install, run the following reconfigure playbook to make this change in your environment:

```
sible/next/hos/ansible/
k -i hosts/verb_hosts FND-CLU-reconfigure.yml
```