iVolve

Detailed Design Document

Private Cloud Build Plus

|  |  |
| --- | --- |
| Prepared by: | Yoshi Kadokawa <yoshi.kadokawa@canonical.com> |
| Prepared on: | 26 March, 2021 |
| Version: | 1.0 |



**Design Stakeholders**

iVolve

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Title** |
| Bilal Bin Ameer | bilal.ameer@ivolve.io | Manager DevOps |
| Rashid Iqbal | rashid.iqbal@ivolve.io | Senior DevOps Engineer |

Canonical

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Title** |
| Yoshi Kadokawa | yoshi.kadokawa@canonical.com | Field Software Engineer |
| Mark Maglana | mark.maglana@canonical.com | Field Software Engineer |

Revisions

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Notes** |
| 2020-03-26 | 1.0 | Initial DRAFT |
|  |  |  |

# 

Table of Contents

[**1. Introduction**](#_i27fc6426ai0) **7**

[1.1. Intended readers](#_mu18w2q4yt9g) 7

[**2. Design Overview**](#_ic6b5g19j38l) **8**

[2.1. Technical factors affecting design](#_m8v6kn4076o) 8

[2.2. Constraints](#_humiaujhxxbn) 8

[**3. Software Versioning**](#_yqg34mw5i8c4) **9**

[**4. Environment**](#_vlrel2qu56e3) **10**

[4.1. High Level Architecture](#_hroelb5fhm2s) 10

[4.2. Scale](#_hk12qt88uq9a) 10

[4.3. Hardware selection](#_cnqrsxrbr011) 10

[4.3.1. Infrastructure Nodes](#_7rn6ervg6gtn) 10

[4.3.2. Control Nodes](#_mqh2uinzvyx4) 11

[4.3.3. Compute Nodes](#_w8ric3702xg5) 11

[4.3.4. Storage Nodes - All Flash](#_tudl9ubkkkhb) 11

[4.3.5. Storage Nodes - Hybrid](#_7zu6sbs55r6d) 12

[4.4. Data Centres](#_mgtwm95gtj56) 13

[4.4.1. Data Centre 1](#_qa8iqbxw8alq) 13

[4.4.2. Physical Environment](#_ew3ng65ot9tz) 13

[4.5. Implementation](#_m16bghqc8u88) 15

[4.5.1. Data Center Cloud 1](#_u373vv7dlmss) 15

[4.6. Deployment](#_qbqn59ndr2nu) 15

[4.6.1. Logical overview](#_y7aohu8g0h0e) 15

[4.7. Operations](#_sdppvncepivh) 16

[4.7.1. Monitoring Integration](#_o0zvwwqhmanu) 16

[4.7.1.1. Graylog](#_eof06udx6h95) 16

[4.7.1.2. Prometheus](#_cvnqhctciodu) 16

[4.7.1.3. FileBeat](#_4w745i7qwyom) 17

[4.7.1.4. Nagios](#_l8544mir00bi) 17

[4.7.1.5. NRPE](#_xhh4lrslu0hq) 17

[4.7.1.6. Grafana](#_hb9ncgpna5gi) 17

[4.8. Support Services](#_7i16chw9npii) 18

[4.8.1. NTP](#_ain00r42ap5t) 18

[4.8.1. DNS](#_qtv0eijjhs68) 18

[**5. Infrastructure Configuration**](#_rg885bk60d8b) **19**

[5.1. Service Allocation Mapping](#_2njj0c2yygi3) 19

[**6. OpenStack Configuration**](#_u2o2po1ki8j9) **20**

[6.1. Service Allocation Mapping](#_4q5tuv2u3ji5) 20

[6.2. Compute](#_eknildn9e2x3) 22

[6.2.1. Hypervisor](#_wxxfyr3vvxea) 22

[6.2.2. Migration of VMs](#_w2ypoxjspdtl) 22

[6.2.3. Pinning](#_qocnwicqf5f8) 23

[6.2.4. Huge Pages](#_xqbfzx9bkep3) 23

[6.2.5. Filters](#_xfsh7dwwhgut) 24

[6.2.6. Anti-Affinity and Affinity](#_d9qkr5z20kgl) 25

[6.2.7. Overcommit Ratios](#_y4fqfaqa9t7t) 25

[6.2.8. Availability Zones](#_bre10jjbuike) 25

[6.3. Identity](#_4waysghc5n2u) 26

[6.3.1. Keystone LDAP](#_3cqik8iku3w1) 26

[**7. Network**](#_k253aklglrbc) **26**

[7.1. Host Networking](#_9hyjaxivjgtu) 26

[7.1.1. Logical](#_fbk4apy804b5) 26

[7.1.1.1. Default Gateway](#_4jmmtxo52uuj) 29

[7.1.1.2. Floating IP Ranges](#_n86d89cwm4ef) 29

[7.1.2. Physical](#_kin2e5w24w9h) 29

[7.1.2.1. Infrastructure nodes](#_qf6x41w33ig) 29

[7.1.2.2. Control nodes](#_wswh0lm1bdro) 30

[7.1.2.3. Compute nodes](#_b0ydeuuvwphj) 31

[7.1.2.4. Storage nodes](#_mpzzxx6226qp) 31

[7.1.3. Jumbo Frames](#_fubg2ikznksg) 32

[7.1.5. CIDR & VLAN requirements](#_xl5qypy4daag) 32

[7.1.6. IP Addressing](#_yz1rp8qfw1sj) 32

[7.2. Overlay](#_acryj5bxc95e) 33

[7.3. High-availability of API Services](#_20woachx3h85) 33

[7.4. SSL Certificates](#_9ptxwtxhru1j) 35

[7.5. OVN networking](#_18eti2ta5ww) 35

[7.6. Load Balancing as a Service (LBaaS)](#_8509mhkssd9k) 36

[7.6.1. Octavia use cases](#_nwgsppxsjmfd) 36

[7.6.2. Certificates for communication with Amphorae](#_t7wlke9j54zu) 38

[7.6.3. Octavia Amphora image](#_kvxddgocehfa) 39

[**8. Storage**](#_di8so581v7gn) **39**

[8.1. Ceph](#_diuul2r6fg8r) 40

[8.1.1. Capacity](#_yg79khyi58ud) 40

[8.1.2. Crush rules and Services](#_1efb4asfvnrf) 40

[8.2. Disk layout](#_7czvu5h2gf3d) 40

[8.2.1. Infrastructure](#_czkvmiez3n64) 40

[8.2.2. Controller](#_klreq9v2hi9g) 41

[8.2.3. Compute](#_1zo5syx15f3k) 41

[8.2.4. Storage (Hybrid)](#_wimccbvyjnm5) 42

[8.2.5. Storage (All Flash)](#_8revjzm9aa5z) 42

[8.3. Cinder](#_cbwgbml7uejh) 43

[8.3.1. Volume type](#_6qy0f9euh8xr) 43

[8.3.2. Default Volume type](#_ms2nfys3ecxm) 43

[8.4. Encryption at Rest](#_2h3hdvoj8gbg) 43

[8.5. Security Design](#_8v4g6gc8rwmo) 44

# 

# 1. Introduction

This document contains the Low Level Design (LLD) for the project. This document is complemented by the High Level Design document and other artifacts provided by Canonical.

The purpose of this document is to describe the project architecture in detail for the first Data Centre Cloud deployment. This includes overview of data centres, components and technologies used to build the solution, including configuration details and data centre specific details.

This document does not cover operational or change processes.

## 1.1. Intended readers

This document is intended for all readers who wish to get detailed, engineer level information, on the architecture of the Customer Data centre. The principal audience for this report is Customer’s design and engineering teams, operations teams and any other service personnel directly or indirectly involved in this project. This document is also intended to guide Canonical and partner staff in the delivery of the cloud.

# 2. Design Overview

## 2.1. Technical factors affecting design

The two main categories of design decisions are the hardware BOM and the current and proposed security & operational requirements.

Hardware availability, performance, and price drive most of the cost during the deployment phase. It is essential to come up with hardware characteristics that will provide the requested functionality.

While the choice of hardware drives the initial cost of procurement, security & operational procedures take additional effort and time and drive the overall deployment time.

Further, specific operational requirements can also increase complexity for the day to day operations of the cloud. Therefore, it is essential that these requirements are part of the initial scope, and we adequately address during the cloud design.

## 2.2. Constraints

A general-purpose OpenStack environment should provide cloud services to both Network and IT consumers within their business. It should meet a wide range of prospective requirements and offer optimizations and features to support this. This document accommodates the design decisions made during the design workshop sessions.

# 3. Software Versioning

The cloud will be delivered with the following software versions:

|  |  |  |
| --- | --- | --- |
| **Component** | **Version** | **More information** |
| Ubuntu (HostOS) | 20.04 LTS | <https://www.ubuntu.com/download/server> |
| Linux Kernel | V5.4 (distro) |  |
| OpenStack | Ussuri |  |
| MAAS | 2.9 or newer stable release | <https://maas.io> |
| Juju | 2.8 or newer stable release | <https://jujucharms.com> |
| OpenStack Charms | 21.01 | <https://docs.openstack.org/charm-guide/latest/2101.html>  <https://docs.openstack.org/charm-guide/latest/release-schedule.html#id2> |

The official support matrix of the components are available on the [Ubuntu Release End of Life](https://www.ubuntu.com/info/release-end-of-life) page. OpenStack Ussuri version provides extended support for customers until 2030. All Canonical tooling including Juju and MAAS are following the Ubuntu LTS support lifecycle. Juju Charms follow the OpenStack lifecycle but also have smaller updates between OpenStack versions including fixes and improvements on the charms.

# 4. Environment

## 4.1. High Level Architecture

At the core of the architecture is Canonical's OpenStack solution, with Ubuntu as the operating system.

## 4.2. Scale

The initial deployment of the Data Centre Cloud will be 36 nodes comprising:

* 3 Infrastructure nodes hosting MAAS, Juju and monitoring and logging services
* 3 Control nodes hosting OpenStack.
* 10 Compute nodes running nova compute and KVM.
* 11 (6 All Flash + 5 Hybrid) Storage nodes(Ceph Nodes), providing block storage to the Cloud.

It will be possible to scale an individual Data Centre Cloud to between 200-250 nodes depending on control plane load / rate of change without further architectural change. Scaling beyond this point may require re-architecture of the control services.

## 4.3. Hardware selection

For the implementation, network equipment and compute power will be provided by Inspur.

Selected servers are based on Canonical’s BOM for Private Cloud builds. The specification of the chosen hardware nodes in each role is as follows:

### 4.3.1. Infrastructure Nodes

|  |  |
| --- | --- |
| **Server Model** | Inspur NF5180M5 |
| **CPU** | Intel® Xeon® Silver 4210R Processor 8C/16T x2 |
| **RAM** | 32GB DDR4 ECC-RDIMM x4 (128GB in total) |
| **SSD** | SSD 240G SATA 6Gbps 2.5in x2 (RAID 1) |
| **Disk** | 2TB HDD x2 (RAID1) |
| **NVME** | -- |
| **Network** | 1Gbps 2-ports  10Gbps 2-ports x2 |

### 4.3.2. Control Nodes

|  |  |
| --- | --- |
| **Server Model** | Inspur NF5180M5 |
| **CPU** | Intel® Xeon® Silver 4214R Processor 12C/24T x2 |
| **RAM** | 32GB DDR4 ECC-RDIMM x8 (256GB in total) |
| **SSD** | SSD 240G SATA 6Gbps 2.5in x2 (RAID 1)  SSD 1.92T SATA 6Gbps 2.5in x2 (RAID 1) |
| **Disk** | -- |
| **NVME** | -- |
| **Network** | 1Gbps 2-ports  25Gbps 2-ports x2 |

### 

### 4.3.3. Compute Nodes

|  |  |
| --- | --- |
| **Server Model** | Inspur NF5180M5 |
| **CPU** | Intel® Xeon® Silver 4214R Processor 12C/24T x2 |
| **RAM** | 32GB DDR4 ECC-RDIMM x24 (768GB in total) |
| **SSD** | SSD 240G SATA 6Gbps 2.5in x2 (RAID 1) |
| **Disk** | -- |
| **NVME** | -- |
| **Network** | 1Gbps 2-ports  10Gbps 2-ports x2 |

### 4.3.4. Storage Nodes - All Flash

|  |  |
| --- | --- |
| **Server Model** | Inspur NF5280M5 |
| **CPU** | Intel® Xeon® Gold 5220R Processor 24C/48T x2 |
| **RAM** | 32GB DDR4 ECC-RDIMM x12 (384GB in total) |
| **SSD** | SSD 240G SATA 6Gbps 2.5in x2 (RAID 1) |
| **Disk** | -- |
| **NVME** | 4TB x24 |
| **Network** | 1Gbps 2-ports  25Gbps 2-ports x2 |

### 4.3.5. Storage Nodes - Hybrid

|  |  |
| --- | --- |
| **Server Model** | Inspur NF5266M5 |
| **CPU** | Intel® Xeon® Silver 4210R Processor 10C/20T x2 |
| **RAM** | 32GB DDR4 ECC-RDIMM x12 (384GB in total) |
| **SSD** | SSD 240G SATA 6Gbps 2.5in x2 (RAID 1) |
| **Disk** | 12TB SAS 12Gbps 7.2Krpm 3.5in\_Enterprise x20 |
| **NVME** | 3.84TB U.2 8GTps x4 |
| **Network** | 1Gbps 2-ports  25Gbps 2-ports x2 |

## 

## 4.4. Data Centres

### 4.4.1. Data Centre 1

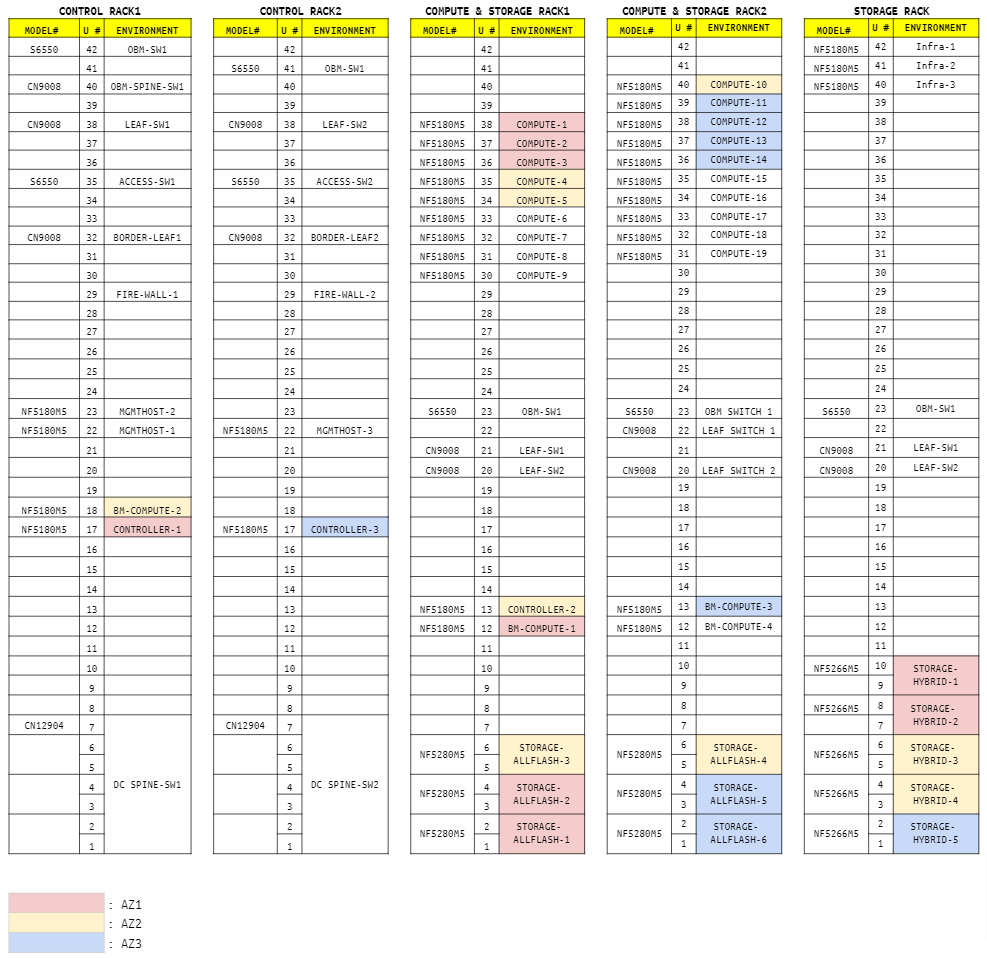
This will be the first of three Data Centre Clouds deployed for Customer. The location of the first cloud will be Karachi.

### 4.4.2. Physical Environment

The cloud will be deployed in a model which provides three availability zones, initially comprising 5 racks.

Diagram Fig.1 shows the initial rack layout.

Fig.1



NOTE: Although the cloud is configured with three availability zones, storage nodes are not evenly distributed through 3 racks. Therefore, losing one whole rack may result in data loss.

## 

## 4.5. Implementation

### 4.5.1. Data Center Cloud 1

Data Centre Cloud 1 is the first implementation phase of the project. It includes design sessions and deployment of the first production cloud.

Implementation follows Canonical’s Private Cloud Build process, which is based on an iterative process. Solution is re-deployed multiple times, whenever a new feature is added. This ensures redeployment of provided solutions, which then helps in eliminating deployment challenges in other environments.

On top of standard Canonical’s Private Cloud Build features, Customer’s Data Centre Cloud will include these features:

* None

The Data Centre Cloud will host production instances both for IT and Network workloads.

## 4.6. Deployment

Once hardware is delivered and integrated in the customer’s data centre, Canonical will be given remote access to IPMI network via a Fortinet VPN. Access is provided to IPMI network and OAM network. This will allow installation of Infrastructure nodes using BMC KVM functionality. Once the Infrastructure nodes are installed, the rest of the deployment is done using Juju and MAAS controlled by the Foundation Engine tooling.

The deployment will be done in an automated fashion to build confidence in the repeatability of OpenStack deployment in future.

### 4.6.1. Logical overview

The deployment is of a disaggregated architecture, with specific node profiles for different roles in the cloud. Infra nodes hold supporting functions such as logging and monitoring. Compute nodes provide compute services to workloads. Control nodes host control plane for OpenStack. Storage nodes host Ceph storage services. This architecture provides a number of benefits:

* Elimination of concerns over ‘noisy neighbor’ services on converged nodes
* Separation of scaling points for compute and storage resources
* Separation of scaling points for block and object storage

Isolation between control services is done using LXD technology.

## 4.7. Operations

### 4.7.1. Monitoring Integration

#### 4.7.1.1. Graylog

Graylog is a centralised log aggregator and analyzer. All normal logging data is collected as standard and is available for analysis as required. The creation of specific filters highlighting specific data such as security related events is also supported.

#### 4.7.1.2. Prometheus

Prometheus is an open-source monitoring and alerting toolkit. Telegraf will be used to collect metrics for Prometheus, and Grafana will be used to visualise the collected metrics.

Prometheus's main features are:

* a multi-dimensional data model (time series identified by metric name and key/value pairs)
* a flexible query language to leverage this dimensionality
* no reliance on distributed storage; single server nodes are autonomous
* time series collection happens via a pull model over HTTP
* pushing time series is supported via an intermediary gateway
* targets are discovered via service discovery or static configuration
* multiple modes of graphing and dashboarding support

High level OpenStack metrics are gathered and exposed by prometheus-openstack-exporter[[1]](#footnote-0). The following metrics are exposed:

* Neutron
  + Neutron networks size, Neutron floating IP and router IP usage statistics.
* Cinder
  + Cinder volume metric (GB), Cinder volume metric (number of volumes).
* Nova
  + Number of running VMs, Total number of vCPUs, Number of used vCPUs, Total amount of memory in MBs, Used memory in MBs, Total amount of disk space in GBs, Used disk space in GBs, Number of schedulable instances, Number of schedulable instances we have capacity for, Nova instances, Nova RAM usage metric, Nova vCPU usage, Nova disk usage, OpenStack overcommit ratios, Nova cores, Nova floating IP addresses (number), Nova instances (number), Nova RAM (MB).

#### 4.7.1.3. FileBeat

Filebeat can forward logs to Graylog.

#### 4.7.1.4. Nagios

Nagios has become an industry standard. Its main characteristics are robustness, extensibility and configurability. In the Private Cloud Build, Nagios is running within a virtual machine located on one of the infra nodes.

Nagios provides a web interface on TCP port 80 with an optional TLS version on port 443, and charms support configuring Nagios to send alerts over email, pager, or pagerduty. There are both readwrite and readonly roles that simplify operational tasks.

Data collected by Nagios is stored locally and Nagios can be queried to show availability of a service over a specific period of time. Severity levels by default are ‘warning’ and ‘critical’.

While monitoring a remote service from Nagios itself is possible, the data collected that way might not be very useful. In general, monitoring a service or a machine from a remote host depends on the service itself. Nagios Remote Plugin Executor is used for running checks on remote nodes instead.

#### 4.7.1.5. NRPE

NRPE is a core component of Nagios and it allows Nagios to execute monitoring commands on remote hosts. While Nagios is running on a separate machine, NRPE daemon is running on the host that is being monitored. This daemon is authenticated with Nagios and runs checks requested by Nagios.

As it is with logging, whenever a new unit of a service is added, monitoring scales automatically and all checks are in place from the start. This ensures both the consistency and coverage of monitoring across the cloud.

Since these checks are encoded in charms, adding checks for a service requires adding those checks to the charms and updating the charms within the environment. This provides revision control of monitoring in addition to consistency and coverage benefits.

#### 4.7.1.6. Grafana

Grafana is a popular visualization tool with support for several data sources including Prometheus. As everything else in this document, it also comes out of the box, fully integrated into the solution. By default it uses TCP port 3000.

Grafana is included for visualization of data from Prometheus. Dashboards are included for OpenStack and Ceph as part of the Private Cloud deployments. Additionally, custom dashboards can be configured using any of the data from Prometheus to allow for dashboards for customer specific use cases.

## 4.8. Support Services

### 4.8.1. NTP

All units will use MAAS with the iVolve provided upstream servers as a reference.

The upstream iVolve NTP servers are:

* TBD by iVolve

### 4.8.1. DNS

MAAS will maintain its local zone and will act as forwarding server to a DNS server running on infrastructure systems. Each infrastructure node will contain a Bind9 DNS server managed by MAAS with identical state maintained by region controllers.  
MAAS will be configured with the following DNS servers:

* TBD by iVolve

# 5. Infrastructure Configuration

## 5.1. Service Allocation Mapping

The below diagram illustrates how the infrastructure services are mapped to the three infrastructure nodes in the deployment.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Infra #1** | **Infra #2** | **Infra #3** |
| **Rack** | COMPUTE & STORAGE RACK1 | CONTROL RACK1 | COMPUTE & STORAGE RACK2 |
| **AZ** | 1 | 2 | 3 |
| **Bare Metal** | MAAS Region API | MAAS Region API | MAAS Region API |
| MAAS Rack | MAAS Rack | MAAS Rack |
| MAAS PostgreSQL | MAAS PostgreSQL | MAAS PostgreSQL |
| **KVM** | Juju Controller | Juju Controller | Juju Controller |
| Landscape Server | Landscape Server | Landscape Server |
| Landscape RabbitMQ | Landscape RabbitMQ | Landscape RabbitMQ |
| Landscape PostgreSQL | Landscape PostgreSQL | Landscape PostgreSQL |
| Grafana | Nagios | Prometheus |
| Elasticsearch | Elasticsearch | Elasticsearch |
| Graylog | Graylog | Graylog |
| Vault | Vault | Vault |

# 

# 6. OpenStack Configuration

The OpenStack services comprise the core components of OpenStack spanning over the three pillars of compute, storage and networking. These services (including identity, image management and a web interface) integrate the OpenStack components with each other as well as external systems to provide a unified experience for users as they interact with different cloud resources. All the OpenStack services will be deployed in HA.

## 6.1. Service Allocation Mapping

The following table illustrates the allocation of services to nodes in the OpenStack Cloud, including Control, Compute, Storage and Networking elements in line with the chosen architecture.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Rack** | **AZ** | **Bare Metal** | **LXD** | **LXD** | **LXD** | **LXD** | **LXD** | **LXD** | **LXD** |
| **Control #1** | **1** | 1 |  | ceph-mon | ceph-radosgw | ceph-fs | cinder | keystone | nova-cloud -controller | etcd |
| barbican | Img Sync[[2]](#footnote-1) | ovn-central | manila | aodh | openstack-dashboard | designate |
| glance | manila-ganesha | placement | gnocchi | octavia | openstack- service-check | rabbitmq- server |
| heat | neutron-api | ceilometer | mysql | EasyRSA |  |  |
| **Control #2** | **2** | 2 |  | ceph-mon | ceph-radosgw | ceph-fs | cinder | keystone | nova-cloud -controller | etcd |
| barbican | designate -bind | ovn-central | manila | aodh | openstack-dashboard | designate |
| glance | manila-ganesha | placement | gnocchi | octavia | prometheus -ceph-exporter | rabbitmq- server |
| heat | neutron-api | ceilometer | mysql |  |  |  |
| **Control #3** | **3** | 3 |  | ceph-mon | ceph-radosgw | ceph-fs | cinder | keystone | nova-cloud -controller | etcd |
| barbican | designate -bind | ovn-central | manila | aodh | openstack-dashboard | designate |
| glance | manila-ganesha | placement | gnocchi | octavia | prometheus- openstack -exporter | rabbitmq- server |
| heat | neutron-api | ceilometer | mysql |  |  |  |
| **Storage #1 - #11** | **3,4,5** | 1,2,3 | Ceph OSD |  |  |  |  |  |  |  |
| **Compute #1 - #10** | **3,4,5** | 1,2,3 | Nova KVM |  |  |  |  |  |  |  |

This service allocation is based on an evaluation of the characteristics of each service, including the compute and storage load the service requires. The general analysis of OpenStack services is shown below

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **High impact** | **High disk IO** | **Containerized** | **Scale-out** | **HA type** | **Units Needed** |
| **Mysql** | **Y** | **Y** | Y | N | Group replication | 3 |
| **Keystone** | **Y** | N | Y | Y | Pacemaker | 3 |
| **Nova Cloud Controller** | **Y** | N | Y | Y | Pacemaker | 3 |
| **RabbitMQ** | **Y** | N | Y | N | Built-in | 3 |
| **RADOSGW** | **Y** | N | Y | Y | Pacemaker | 3 |
| **Designate** | **Y** | N | Y | Y | Pacemaker | 3 |
| **Designate-bind, memcache** | N | N | Y | Y | Round-robin | 2 |
| **Ceph Mon** | N | N | Y | Y | Built-in | 3 |
| **Glance** | N | N | Y | Y | Pacemaker | 3 |
| **Cinder** | N | N | Y | Y | Pacemaker | 3 |
| **Neutron API** | N | N | Y | Y | Pacemaker | 3 |
| **MongoDB** | N | N | Y | Y | Built-in | 3 |
| **Gnocchi** | N | N | Y | Y | Pacemaker | 3 |
| **Ceilometer** | N | N | Y | Y | Pacemaker | 3 |
| **Heat** | N | N | Y | Y | Pacemaker | 3 |
| **OpenStack Dashboard** | N | N | Y | Y | Pacemaker | 3 |
| **Glance Simplestreams Sync** | N | N | Y | N | N/A | 1 |
| **Aodh** | N | N | Y | Y | Pacemaker | 3 |
| **Ceph OSD** | - | **Y** | N | Y | Built-in | - |
| **Nova KVM** | - | - | N | Y | N/A | - |
| **Barbican** | N | N | Y | Y | Pacemaker | 3 |
| **Octavia** | N | N | Y | Y | Pacemaker | 3 |
| **Manila** | N | N | Y | Y | Pacemaker | 3 |
| **Manila Ganesha** | N | N | Y | Y | Pacemaker | 3 |
| **CephFS** | N | N | Y | Y | Built-in | 3 |

This analysis allows us to ensure high impact services are not colocated and appropriate constraints are applied in the LXD containers if required. Similarly, by allocating networking services appropriately we avoid hotspots in the control plane nodes.

## 6.2. Compute

### 6.2.1. Hypervisor

The KVM Hypervisor will be configured as the compute service hypervisor. KVM belongs to the Group A set of hypervisors in OpenStack, which means that KVM is fully tested and supported (other hypervisors may have limited testing and/or support).

### 6.2.2. Migration of VMs

Migration of VMs enables an administrator to move a virtual-machine instance from one compute host to another. This feature is useful when a compute host requires maintenance. Migration can also be useful to redistribute the load when many VM instances are running on a specific physical machine.

The migration types are:

* Non-live migration (sometimes referred to simply as 'migration'). The instance is shut down for a period of time to be moved to another hypervisor. In this case, the instance recognizes that it was rebooted.
* Live migration (or 'true live migration'). Almost no instance downtime. Useful when the instances must be kept running during the migration. The different types of live migration are:
  + Shared storage-based live migration. Both hypervisors have access to shared storage. Several shared file systems options are supported, although some of them are not production ready (e.g. cephFS) and therefore not supported
  + Block live migration. No shared storage is required. Incompatible with read-only devices such as CD-ROMs and Configuration Drive (config\_drive).
  + Volume-backed live migration. Instances are backed by volumes rather than ephemeral disk, no shared storage is required, and migration is supported (currently only available for libvirt-based hypervisors).

For this design, the Data Centre Cloud deployment will have Live Migration enabled. Shared storage-based live migration will be enabled due to the use of Ceph as a shared storage backend.

### 6.2.3. Pinning

CPU Pinning may be configured on a subset of hosts to dedicate the first two cores of each NUMA cell (physical CPU) to the host OS and OpenStack services. Two options to guarantee that are passing the isolcpus kernel parameter or setting CPUAffinity configuration for systemd. Isolcpus is considered the classic approach and CPUAffinity is considered the modern approach and therefore all new cloud deployments will use it.

OpenStack Nova vcpu\_pin\_set is set through the charm configuration in order to ensure the spawned VMs are scheduled only on the dedicated CPU cores. It pairs well with both host process pinning options.

|  |  |  |
| --- | --- | --- |
| **Nodes to configure CPU pinning\*** | **Cores reserved for Host** | **Cores dedicated for OpenStack Nova** |
|  | ex.) 8 Cores | ex.) 40 Cores |

\*NOTE: Due to [this](https://bugs.launchpad.net/charm-nova-cloud-controller/+bug/1468871), there is a limitation that live-migration is not supported between Nova Compute nodes with different parameters, such as Nova compute nodes with vcpu\_pin\_set configured, and the ones without vcpu\_pin\_set configured.

### 6.2.4. Huge Pages

The standard page size in x86 systems is 4 kB. This is optimal for general purpose computing but larger page sizes - 2 MB and 1 GB - are also available. These larger page sizes are known as huge pages. Huge pages result in less efficient memory usage as a process will not generally use all memory available in each page. However, use of huge pages will result in fewer overall pages and a reduced risk of Translation Lookaside Buffer misses. For processes that have significant memory requirements or are memory intensive, the benefits of huge pages frequently outweigh the drawbacks. For more information, refer to the [official OpenStack documentation](https://docs.openstack.org/nova/stein/admin/huge-pages.html).

Note that huge pages can only be used by KVM instances, and KVM instances can only use huge pages. The “regular” processes running on the compute node will have to use whatever is left of non-hugepages memory.

|  |  |
| --- | --- |
| Nodes to configure Hugepages |  |
| Pagesize | 2MB or 1GB |
| Allocation | ex.) 50% |
| Reserved memory for Host OS | 16GB |

### 6.2.5. Filters

The following filters will be set in parameter scheduler\_default\_filters located in config file nova.conf.

|  |  |
| --- | --- |
| Filter | Description |
| AvailabilityZoneFilter | Filters hosts by availability zone. It passes hosts matching the availability zone specified in the instance properties. |
| ComputeFilter | Filters hosts based upon service availability. Only hosts that have a running compute service are passed |
| ComputeCapabilitiesFilter | Filters hosts based upon extra specs. This is eg. used for CPU pinning and huge pages |
| ImagePropertiesFilter | Filter hosts based upon image properties |
| ServerGroupAntiAffinityFilter | Filter hosts based upon placement of other instances. This ensures that no two instances in one servergroup share a host. To be used with a named server group |
| ServerGroupAffinityFilter | Filter hosts based upon placement of other instances. Ensures instances are placed on one of a set of group hosts. To be used with a named server group |
| DifferentHostFilter | Allows the instance on a different host from a set of instances |
| SameHostFilter | Puts the instance on the same host as another instance in a set of instances |
| NUMATopologyFilter | Filters hosts based on the NUMA topology requested by the instance, if any. |

See <http://docs.openstack.org/developer/nova/filter_scheduler.html> for more details on specific filters.

### 6.2.6. Anti-Affinity and Affinity

Affinity is the policy that forces Nova to host the concerned VMs in the same hypervisor. Anti-Affinity is the policy that forces Nova to host the concerned VMs each in a different hypervisor.

Enabling Affinity and Anti-Affinity is simply done by adding ServerGroupAffinityFilter and ServerGroupAntiAffinityFilter into scheduler\_default\_filters. By default they are already included.

### 6.2.7. Overcommit Ratios

Overcommit ratios define the maximum ratio at which virtual CPUs to physical CPUs or threads are assigned. This can be assigned at a number of levels with the most granular being per physical node and the least granular being for the cloud as a whole.

|  |  |
| --- | --- |
| Overcommit | Ratio |
| CPU | 4:1 |
| RAM | 1:1 |
| Disk | T.B.D |

### **6**.2.8. Availability Zones

Nova:

|  |  |
| --- | --- |
| AZ | Nodes |
| 1 | compute-1, compute-2, compute-3 |
| 2 | compute-4, compute-5, compute-10 |
| 3 | compute-11, compute-12, compute-13  compute-14 |

\*NOTE: At the moment, Availability Zone is not supported on Neutron with OVN.

## 6.3. Identity

### 6.3.1. Keystone LDAP

Keystone v3 deployments support the use of domain specific identity drivers, allowing different types of authentication backend to be deployed in a single Keystone deployment. The keystone-ldap charm used in this deployment supports use of an LDAP domain backend, with configuration details provided by charm configuration options.

|  |  |
| --- | --- |
| **Key** | **Value** |
| LDAPS Servers |  |
| Certificates |  |
| OpenStack domain name |  |
| LDAP Bind Account |  |
| User DN |  |
| Group DN |  |
| User filter |  |

# 7. Network

The network layer is provided by two types of switches:

* Management fabric: Inspur S6550 switches
* Data fabric: Inspur CN9008 switches

## 7.1. Host Networking

### 7.1.1. Logical

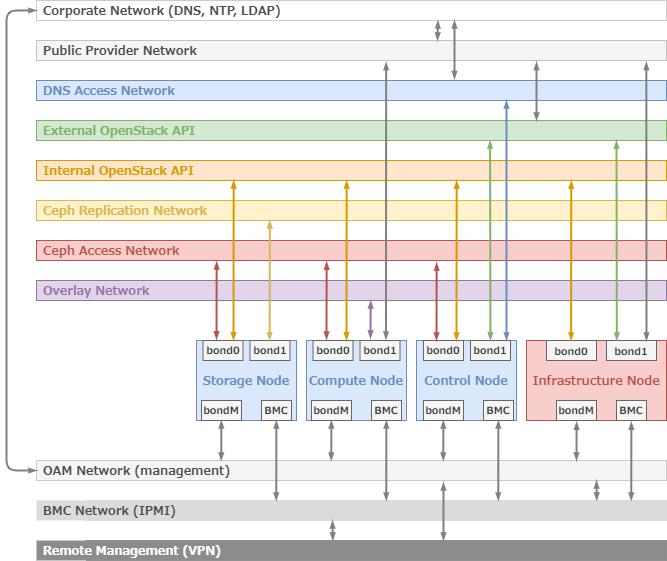
The logical networks configured in the Data Centre Cloud will be

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Network** | **Purpose** | **Fabric** | **Connectivity** |
| OAM | DHCP/PXE/Juju agents/Monitoring | Management | This network is used to manage the cluster and should be routable for administrators |
| BMC | IPMI | Management | No external connectivity required |
| Storage Access | Ceph Storage Access network used internally for communication between storage consuming services and ceph | Data | No external connectivity required |
| Storage Replication | Ceph Storage Replication Network | Data | No external connectivity required |
| Internal API | Internally accessible OpenStack API network | Data | No external connectivity required |
| External API | Network to provide API endpoints for OpenStack services | Data | Accessible for any potential OpenStack users. |
| Underlay network for overlay network traffic | Tenant network (Geneve) | Data | No external connectivity required |
| DNS Access network | Network to provide DNS services from Designate | Data | Accessible for any potential network |
| Provider network | Publicly accessible space for OpenStack tenants | Data | Accessible for any potential application consumers |

These are mapped to nodes consuming them as follows:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Machines** | **BMC** | **OAM** | **Storage Access** | **Replication** | **Internal API** | **External API** | **Overlay** | **DNS** | **Provider** |
| Infra | Required + Access (routed via OAM) | Required |  |  | Required | Required **for testing** |  |  | Required **for testing** |
| Control | Required | Required | Required |  | Required | Required |  | Required |  |
| Compute | Required | Required | Required |  | Required |  | Required |  | Required |
| Storage | Required | Required | Required | Required | Required |  |  |  |  |

Each node will have three bonds configured, these are referred to as bondM, bond0 and bond1. They are allocated to logical networks as follows:



|  |  |  |
| --- | --- | --- |
| **Network** | **Interface** | **Tagged** |
| BMC | Dedicated BMC Interface + bondM on Infra Nodes(routed via OAM) | Untagged |
| OAM | bondM | Untagged |
| Storage Access | bond0 | Tagged |
| Storage Replication | bond1 | Tagged |
| Overlay | bond1 | Tagged |
| Internal API | bond0 | Tagged |
| External API | bond1 | Tagged |
| Provider | bond1 | Tagged |
| DNS Access | bond1 | Tagged |

For all of the machines and containers deployed for this environment, the default gateway will be configured to the following address.

|  |  |
| --- | --- |
| **Network** | **Gateway Address** |
| OAM network |  |

#### 7.1.1.2. Floating IP Ranges

Floating IPs will be assigned as blocks and can be assigned as required, there is no impediment to the use of multiple small blocks assigned as required as opposed to a single large network.

### 7.1.2. Physical

#### 7.1.2.1. Infrastructure nodes

Machines have 3 PCIE devices, these are:

* <CARD VENDOR> card with two 1Gb ports
* 2x <CARD VENDOR> cards each with two 10Gb ports

As such, the port allocations on the hosts are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NIC** | **Port** | **Bond** | **Bond Purpose** | **Leaf Switch** |
| A (Riser Card 1) | 1 | bondM | OAM | MGMT A |
| A (Riser Card 1) | 2 | bondM | OAM | MGMT B |
| B (Riser Card 2) | 1 | bond0 | Internal API | DATA A |
| B (Riser Card 2) | 2 | bond0 | Internal API | DATA B |
| C (Riser Card 3) | 1 | bond1 | External API/Provider | DATA A |
| C (Riser Card 3) | 2 | bond1 | External API/Provider | DATA B |
| Onboard BMC | 1 | N/A | BMC port | MGMT |

<Photo of the rear with labels of each port of the server>

#### 7.1.2.2. Control nodes

Machines have 3 PCIE devices, these are:

* <CARD VENDOR> card with two 1Gb ports
* 2x <CARD VENDOR> cards each with two 25Gb ports

As such, the port allocations on the hosts are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NIC** | **Port** | **Bond** | **Bond Purpose** | **Leaf Switch** |
| A (Riser Card 1) | 1 | bondM | OAM | MGMT A |
| A (Riser Card 1) | 2 | bondM | OAM | MGMT B |
| B (Riser Card 2) | 1 | bond0 | Storage Access/Internal API | DATA A |
| B (Riser Card 2) | 2 | bond0 | Storage Access/Internal API | DATA B |
| C (Riser Card 3) | 1 | bond1 | External API/DNS Access | DATA A |
| C (Riser Card 3) | 2 | bond1 | External API/DNS Access | DATA B |
| Onboard BMC | 1 | N/A | BMC port | MGMT |

<Photo of the rear with labels of each port of the server>

#### 7.1.2.3. Compute nodes

Machines have 3 PCIE devices, these are:

* <CARD VENDOR> card with two 1Gb ports
* 2x <CARD VENDOR> cards each with two 10Gb ports

As such, the port allocations on the hosts are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NIC** | **Port** | **Bond** | **Bond Purpose** | **Leaf Switch** |
| A (Riser Card 1) | 1 | bondM | OAM | MGMT A |
| A (Riser Card 1) | 2 | bondM | OAM | MGMT B |
| B (Riser Card 2) | 1 | bond0 | Storage Access/Internal API | DATA A |
| B (Riser Card 2) | 2 | bond0 | Storage Access/Internal API | DATA B |
| C (Riser Card 3) | 1 | bond1 | Overlay/Provider | DATA A |
| C (Riser Card 3) | 2 | bond1 | Overlay/Provider | DATA B |
| Onboard BMC | 1 | N/A | BMC port | MGMT |

<Photo of the rear with labels of each port of the server>

#### 7.1.2.4. Storage nodes

Machines have 3 PCIE devices, these are:

* <CARD VENDOR> card with two 1Gb ports
* 2x <CARD VENDOR> cards each with two 25Gb ports

As such, the port allocations on the hosts are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NIC** | **Port** | **Bond** | **Bond Purpose** | **Leaf Switch** |
| A (Riser Card 1) | 1 | bondM | OAM | MGMT A |
| A (Riser Card 1) | 2 | bondM | OAM | MGMT B |
| B (Riser Card 2) | 1 | bond0 | Storage Access/Internal API | DATA A |
| B (Riser Card 2) | 2 | bond0 | Storage Access/Internal API | DATA B |
| C (Riser Card 3) | 1 | bond1 | Storage Replication | DATA A |
| C (Riser Card 3) | 2 | bond1 | Storage Replication | DATA B |
| Onboard BMC | 1 | N/A | BMC port | MGMT |

<Photo of the rear with labels of each port of the server>

### 7.1.3. Jumbo Frames

Jumbo Frames will be enabled and configured on both the switches and all machine classes. A minimum of 9000 MTU will be set

### 7.1.5. CIDR & VLAN requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Network | VLAN ID | MTU | CIDR |
| OAM | (untagged) | 1500 | TBD by iVolve (Minimum /24) |
| Internal API | TBD by iVolve | 9000 | TBD by iVolve (Minimum /24) |
| External API | TBD by iVolve | 1500 | TBD by iVolve (Minimum /24) |
| Storage Replication | TBD by iVolve | 9000 | TBD by iVolve (Minimum /25) |
| Storage Access | TBD by iVolve | 9000 | TBD by iVolve (Minimum /24) |
| Overlay | TBD by iVolve | 9000 | TBD by iVolve (Minimum /25) |
| DNS Access | TBD by iVolve | 1500 | TBD by iVolve (Minimum /28) |
| Provider | TBD by iVolve | 1500 | TBD by iVolve |

### 7.1.6. IP Addressing

### 

|  |  |  |
| --- | --- | --- |
| Network | Reservation Range | Purpose |
| OAM | TBD by iVolve | GW addresses |
|  | Infra nodes addresses |
|  | VIP addresses |
|  | MAAS DHCP (PXE boot) |
| Internal API | TBD by iVolve | GW addresses |
|  | Infra nodes addresses |
|  | VIP addresses |
| External API | TBD by iVolve | GW addresses |
|  | Infra nodes addresses |
|  | VIP addresses |
| Storage Replication | TBD by iVolve | GW addresses |
| Storage Access | TBD by iVolve | GW addresses |
| Overlay | TBD by iVolve | GW addresses |
| DNS Access | TBD by iVolve | GW addresses |
| Provider | TBD by iVolve | GW addresses |

## 7.2. Overlay

Overlay network will be provided by OVN.

## 7.3. High-availability of API Services

In order to provide a single endpoint to API clients but implement load-balancing between different API endpoints, virtual IPs (VIPs) are used on a per-service basis. Virtual IP usage is possible as the target environment provides a single broadcast domain for the public API network.

The allocation of VIPs to services is summarized in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Service | FQDN | VIP | VLAN |
| MAAS | N/A | TBD by iVolve |  |
| PostgreSQL | N/A | TBD by iVolve |  |
| MySQL | N/A | TBD by iVolve |  |
| Vault | TBD by iVolve | TBD by iVolve |  |
| OpenStack Internal API endpoints | | | |
| Keystone | TBD by iVolve | TBD by iVolve |  |
| Cinder | TBD by iVolve | TBD by iVolve |
| Glance | TBD by iVolve | TBD by iVolve |
| Nova | TBD by iVolve | TBD by iVolve |
| Neutron | TBD by iVolve | TBD by iVolve |
| Designate | TBD by iVolve | TBD by iVolve |
| RADOSGW | TBD by iVolve | TBD by iVolve |
| Aodh | TBD by iVolve | TBD by iVolve |
| Heat | TBD by iVolve | TBD by iVolve |
| Gnocchi | TBD by iVolve | TBD by iVolve |
| Octavia | TBD by iVolve | TBD by iVolve |
| Barbican | TBD by iVolve | TBD by iVolve |
| Ceilometer | TBD by iVolve | TBD by iVolve |
| Placement | TBD by iVolve | TBD by iVolve |
| Manila | TBD by iVolve | TBD by iVolve |
| OpenStack External API endpoints | | | |
| Keystone | TBD by iVolve | TBD by iVolve |  |
| Cinder | TBD by iVolve | TBD by iVolve |
| Glance | TBD by iVolve | TBD by iVolve |
| Nova | TBD by iVolve | TBD by iVolve |
| Neutron | TBD by iVolve | TBD by iVolve |
| Designate | TBD by iVolve | TBD by iVolve |
| RADOSGW | TBD by iVolve | TBD by iVolve |
| Aodh | TBD by iVolve | TBD by iVolve |
| Heat | TBD by iVolve | TBD by iVolve |
| Gnocchi | TBD by iVolve | TBD by iVolve |
| Horizon | TBD by iVolve | TBD by iVolve |
| Octavia | TBD by iVolve | TBD by iVolve |
| Barbican | TBD by iVolve | TBD by iVolve |
| Ceilometer | TBD by iVolve | TBD by iVolve |
| Placement | TBD by iVolve | TBD by iVolve |
| Manila | TBD by iVolve | TBD by iVolve |

## 7.4. SSL Certificates

The OpenStack service hostnames in the previous table need to have SSL certificates and CA key generated by the customer and provided to Canonical to consume in OpenStack deployment.

## 7.5. OVN networking

Distributed Virtual Routing (DVR) is enabled in this cloud deployment.

Configuration uses the bridge-interface-mappings with the interface name and logical bridge name 'br-data'. This port will be used by ovn-chassis to provide connectivity to a physical network, therefore, it has no IP configured by MAAS. Two choices of configuration are possible:

1. A raw bond interface for ovn-chassis

i.e: bridge-interface-mappings: br-data:bond1

ovn-bridge-mappings: physnet1:br-data

flat-network-providers: #left intentionally blank

vlan-ranges: TBD by iVolve

* Note 1.1) No vlan range specified - only a physnet which means no vlan tenant networks and all vlan networks are provider networks created by admin user with a specific segmentation ID. Physnets correspond to fabrics in MAAS. Each fabric has its own set of independent VLANs from 1-4094.
* Note 1.2) Allows for dynamic addition of vlan provider networks
* Note 1.3) For Neutron-API vlan-ranges must match.

1. Flat provider networks with bonds for the data-port

i.e: bridge-interface-mappings: br-100:bond1.100 br-101:bond1.101

ovn-bridge-mappings: physnetvlan100:br-100 physnetvlan101:br-101

flat-network-providers: physnetvlan100 physnetvlan101

vlan-ranges: '' #Charm default cleared

* Note 2.1) A VLAN port can be used here (e.g. configured via MAAS) but note that in this case a provider network must be configured as 'flat' not as 'VLAN' as there will be two 802.1q headers appended - one by the OVS itself and one by the kernel 802.1q module which will result in the lack of connectivity.
* Note 2.2) This does not allow a dynamic addition of new provider networks with different VLANs.

The configuration assumes the first option with raw bond interface passed for ovn-chassis.

## 

## 7.6. Load Balancing as a Service (LBaaS)

Octavia is an open source, operator-scale load balancing solution designed to work with OpenStack. Octavia was born out of the Neutron LBaaS project. Octavia has become the reference implementation for Neutron LBaaS version 2. Octavia accomplishes its delivery of load balancing services depending on the driver type:

* By managing a fleet of virtual machines, containers, or bare metal servers collectively known as amphorae which it spins up on demand. This on-demand, horizontal scaling feature differentiates Octavia from other load balancing solutions, thereby making Octavia truly suited "for the cloud."
* By using native OVN load balancing capabilities. This limits the load balancing capabilities only to TCP and UDP. There is also no health monitoring available.

TLS certificates for traffic termination on the Load Balancer level are provided by Barbican. Barbican is a REST API designed for the secure storage, provisioning and management of secrets such as passwords and encryption keys. It is aimed at being useful for all environments, including large ephemeral Clouds.

### 7.6.1. Octavia use cases

The following list provides an overview of supported scenarios for setting up a load balancer.

**A. Deploy a basic HTTP load balancer with a health monitor**

<https://docs.openstack.org/octavia/queens/user/guides/basic-cookbook.html#deploy-a-basic-http-load-balancer-with-a-health-monitor>

This scenario does not depend on Distributed Virtual Routers. In fact, this scenario does not require any routers at all for load balancers' front-ends. Octavia's Amphorae front-end interfaces are connected directly to the provider network and backends are connected directly to the tenant network.

In this scenario, three IP addresses per load balancer in HA mode are consumed:

* two IP addresses for two Amphora instances,
* one IP address as VIP for the load balancer.

**B. Deploy a basic HTTP load balancer using a Floating IP**

<https://docs.openstack.org/octavia/queens/user/guides/basic-cookbook.html#deploy-a-basic-http-load-balancer-using-a-floating-ip>

This is a working scenario on the condition that High Availability (HA) is not configured for the load balancers. With HA enabled, at the time of writing (April 2020), you would face a bug, which is being worked on in the upstream:

* <https://bugs.launchpad.net/neutron/+bug/1774459>
* <https://review.opendev.org/#/c/669938/>
* <https://review.opendev.org/#/c/685779/>

**C. OVN driven load balancing**

<https://docs.openstack.org/networking-ovn/latest/admin/loadbalancer.html>

This scenario uses OVN networking for load balancing. OVN Provider driver has few advantages when used a provider driver for Octavia over Amphora, like:

* OVN can be deployed without VMs. So there is no additional overhead of VMs as is required currently in Octavia when using the default Amphora driver.
* OVN Load Balancers can be deployed faster than default Load Balancers in Octavia (which use Amphora currently) because of no additional deployment requirement.

The limitations of this approach include:

* OVN currently supports TCP and UDP. So Layer-7 based Load Balancing is not possible with OVN. However, once Layer-7 support is integrated in OVN, this issue can be resolved.
* There are no Health Monitors currently in OVN’s Driver for Load Balancer. Therefore Health Checking is not possible with OVN.
* Currently, OVN driver supports a 1:1 protocol mapping between Listeners and associated Pools i.e. Listener which can handle TCP protocol can only be associated with pools associated to TCP protocol. Pools handling UDP protocol cannot be linked with TCP based Listeners. This limitation will be handled in the upcoming core OVN release.

### 7.6.2. Certificates for communication with Amphorae

Octavia uses client certificates for authentication and security of communication between Amphorae (load balancers) and the Octavia control plane. For the initial version of the Octavia charm, these must be generated by the operator and provided to the Octavia charm as configuration.

The certificates are generated as follows:

PASSWORD=$(cat ~/deploy/secrets/octavia-password.txt)

SUBJECT="/C=TODO/ST=TODO/O=TODO/CN=TODO"

DAYS=TBD by iVolve

mkdir -p demoCA/newcerts

touch demoCA/index.txt

touch demoCA/index.txt.attr

openssl genrsa -passout pass:${PASSWORD} -des3 -out issuing\_ca\_key.pem 2048

openssl req -x509 -passin pass:${PASSWORD} -new -nodes -key issuing\_ca\_key.pem \

-config /etc/ssl/openssl.cnf \

-subj ${SUBJECT} \

-days ${DAYS} \

-out issuing\_ca.pem

openssl genrsa -passout pass:${PASSWORD} -des3 -out controller\_ca\_key.pem 2048

openssl req -x509 -passin pass:${PASSWORD} -new -nodes \

-key controller\_ca\_key.pem \

-config /etc/ssl/openssl.cnf \

-subj ${SUBJECT} \

-days ${DAYS} \

-out controller\_ca.pem

openssl req \

-newkey rsa:2048 -nodes -keyout controller\_key.pem \

-subj ${SUBJECT} \

-out controller.csr

openssl ca -passin pass:${PASSWORD} -config /etc/ssl/openssl.cnf \

-cert controller\_ca.pem -keyfile controller\_ca\_key.pem \

-create\_serial -batch \

-in controller.csr -days ${DAYS} -out controller\_cert.pem

cat controller\_cert.pem controller\_key.pem > controller\_cert\_bundle.pem

Octavia charm is provisioned with the generated certificates and keys during the deployment.

For more details regarding Octavia charm deployment, see upstream documentation at <https://docs.openstack.org/project-deploy-guide/charm-deployment-guide/latest/app-octavia.html>.

### 7.6.3. Octavia Amphora image

Octavia uses Amphorae (cloud instances running HAProxy) to provide LBaaS services. Appropriate image must be uploaded to Glance and tagged with 'octavia-amphora'.

This deployment includes octavia-diskimage-retrofit charm that is used for transforming a stock Ubuntu cloud image into a Octavia HAProxy Amphora image.

octavia-diskimage-retrofit charm is used in concert with glance-simplestreams-sync charm, which is an Ubuntu Cloud images synchronization service. By default, glance-simplestreams-sync pulls official Ubuntu images every day. The service can be paused or a different schedule can be configured if required.

For details regarding how to rotate the Amphora images, see the upstream documentation at <https://docs.openstack.org/octavia/latest/admin/guides/operator-maintenance.html#rotating-the-amphora-images>

# 8. Storage

Storage services are provided by Ceph (block, object, and shared file system). Unlike the traditional approach where appliance solutions are used, the environment will use Ceph to provide all storage requirements.

There are four main storage functions:

* Object storage
* Permanent storage
* Ephemeral storage
* OpenStack images store
* Shared file system storage

Of these, Object Storage will be provided by Ceph Rados Gateway, with Permanent, Ephemeral and Image storage are provided by Ceph RBD. Shared file system storage is provided by CephFS.

Data in Ceph is stored in three copies and it’s configured to spread those copies over three availability zones. This ensures that each availability zone can be lost without short term impact to the service. In case of longer AZ downtime, Ceph will try to ensure replication count of 3 and therefore will use available space within existing availability zones.

## 8.1. Ceph

### 8.1.1. Capacity

Ceph’s capacity can be calculated with few parameters. Number of disks per machine, the size of the disk and the number of machines.

In the setup, across the 11 Ceph nodes will provide a maximum cluster size of 1776TB. This Ceph cluster will have two types of pools.

1. Hybrid - 5 nodes
   1. 20x HDDs with 2x NVMe SSD drives as a bcache
   2. Total raw capacity: 1200TB(240TB per machine)
2. All-flash - 6 nodes
   1. 24x NVMe SSD drives
   2. Total raw capacity: 576TB(96TB per machine)

Since all data is replicated in three copies, the total storage that can be used are these numbers divided in 3.

### 8.1.2. Crush rules and Services

Ceph’s capacity can be calculated with few parameters. Number of disks per machine, the size of the disk and the number of machines.

|  |  |  |
| --- | --- | --- |
| **Machines** | **Device class/Crush rule** | **Services** |
| Storage nodes - Hybrid | hybrid | Cinder  Rados Gateway  CephFS  Gnocchi |
| Storage nodes - All Flash | allflash | Cinder  Nova (Ephemeral storage) |

## 8.2. Disk layout

### 8.2.1. Infrastructure

Each infra node will have the same disk layout. Two 2TB HDD drives on each machine will be assembled in a RAID 1 and backed by an SSD bcache.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disk** | **Device** | **Partition** | **Size** | **Mount Point** | **Usage** |
| **RAID 1**  **(HDD)** | /dev/sda | /dev/sda1 | 512M | /boot/efi | EFI partition |
| /dev/sda2 | 1G | /boot | Boot partition |
| /dev/sda3 | 16G | swap | Swap |
| /dev/sda4 | 240G | / | Root partition |
|
| **RAID 1 (SSD)** | /dev/sdb |  | 240G | Bcache cache device | Bcache |
| **BCACHE** | /dev/bcache0 |  | 1.7T | /var/lib/virt | VMs partition |

### 8.2.2. Controller

Each controller node will have the same disk layout. Two 240GB SSD drives on each machine will be assembled in a RAID 1 for boot and OS purposes, and two 1.92TB SSD drives in RAID 1 for /var partition.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disk** | **Device** | **Partition** | **Size** | **Mount Point** | **Usage** |
| **RAID 1**  **(SSD)** | /dev/sda | /dev/sda1 | 512M | /boot/efi | EFI partition |
| /dev/sda2 | 1G | /boot | Boot partition |
| /dev/sda4 | 238G | / | Root partition |
|
| **RAID 1**  **(SSD)** | /dev/sdb | /dev/sdb1 | 1.92T | /var | Var partition |

### 8.2.3. Compute

Each infra node will have the same disk layout. Two 240GB SSD drives on each machine will be assembled in a RAID 1 for boot and OS purposes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disk** | **Device** | **Partition** | **Size** | **Mount Point** | **Usage** |
| **RAID 1**  **(SSD)** | /dev/sda | /dev/sda1 | 512M | /boot/efi | EFI partition |
| /dev/sda2 | 1G | /boot | Boot partition |
| /dev/sda3 | 238G | / | Root partition |
|

### 8.2.4. Storage (Hybrid)

Each storage node will have the same disk layout. Two 240GB SSD drives on each machine will be assembled in a RAID 1 for boot and OS purposes, and 20x 12TB HDD SAS drives will be left for Ceph OSD. The 2x 3.84TB NVMe drives are used to form 20 bcache devices; one for each Ceph OSD device.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disk** | **Device** | **Partition** | **Size** | **Mount Point** | **Usage** |
| **RAID 1 (SSD)** | /dev/sda | /dev/sda1 | 512M | /boot/efi | EFI partition |
| /dev/sda2 | 1G | /boot | Boot partition |
| /dev/sda3 | 238G | / | Root partition |
| **4x NVMe** | /dev/nvme0n1 | - | 3.84T | Bcache cache device | Bcache for 10 HDD Ceph OSDs |
| /dev/nvme1n1 | - | 3.84T | Bcache cache device | Bcache for 10 HDD Ceph OSDs |
| **20x HDD** | /dev/sd[b..u] | /dev/disk/by-dname/osddisk[1..20] | 12T | Bcache backing device | Ceph OSD |

### 8.2.5. Storage (All Flash)

Each storage node will have the same disk layout. Two 240GB SSD drives on each machine will be assembled in a RAID 1 for boot and OS purposes, and 24x 4TB SSD drives will be left for Ceph OSD.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disk** | **Device** | **Partition** | **Size** | **Mount Point** | **Usage** |
| **RAID 1 (SSD)** | /dev/sda | /dev/sda1 | 512M | /boot/efi | EFI partition |
| /dev/sda2 | 1G | /boot | Boot partition |
| /dev/sda3 | 238G | / | Root partition |
| **24 x SSD**  **(JBOD)** | /dev/sd[b..y] | /dev/disk/by-dname/osddisk[1..24] | 4T | - | Ceph OSD |

## 8.3. Cinder

### 8.3.1. Volume type

Cinder will be configured with the following volume types.

|  |  |
| --- | --- |
| **Volume type name** | **Pool name** |
| premium-volume | cinder-ceph-allflash |
| standard-volume | cinder-ceph-standard |

### 8.3.2. Default Volume type

The following volume type will be configured as the default volume type, which will be used when no volume type is specified.

* standard-volume

## 8.4. Encryption at Rest

As of the 18.05 release, the OpenStack charms support encryption of data in two key areas - local ephemeral instance storage for Nova instances, Ceph OSD block devices.

The objective of this feature is to mitigate the risk of data compromise in the event that disks or full servers are removed from data center deployments.

Encryption of underlying block devices is performed using dm-crypt with LUKS; key management is provided by Vault, which provides secure encrypted storage of the keys used for each block device with automatic sealing of secrets in the event of reboot/restart of services.

## 8.5. Security Design

Consuming application units access Vault using a Vault AppRole and associated policy which is specific to each machine in the deployment; The AppRole enforces uses of a secret id and access is only permitted from the configured network address of the consuming unit. The associated policy only allows the consuming unit to store and retrieve secrets from a specific secrets back-end under a specific sub path (in this case the hostname of the unit).

The secret id for the AppRole is retrieved out-of-band from Juju by the consuming charm; a one-shot retrieval token is provided over the relation from vault to each consuming application which is specific to each unit which can be used to retrieve the actual secret id; the token also has a limited TTL (2 hours) and the call must originate from the configured network address of the consuming unit. The secret id is only ever visible to the consuming unit and vault itself, providing an additional layer of protection for deployments.

LUKS encryption keys are never stored on local disk; vaultlocker is used to encrypt and store the key in vault, and to retrieve the key and open encrypted block devices during boot. Keys are only ever held in memory.

1. https://github.com/CanonicalLtd/prometheus-openstack-exporter [↑](#footnote-ref-0)
2. Image synchronization from image archite to Glance is implemented in glance-simplestreams-sync charm. Additionally charm octavia-diskimage-retrofit converts the regular Ubuntu images into Octavia-compatible images [↑](#footnote-ref-1)