# LAB – OpenStack Installation

Hands on experience with OpenStack installation using JuJu

References:

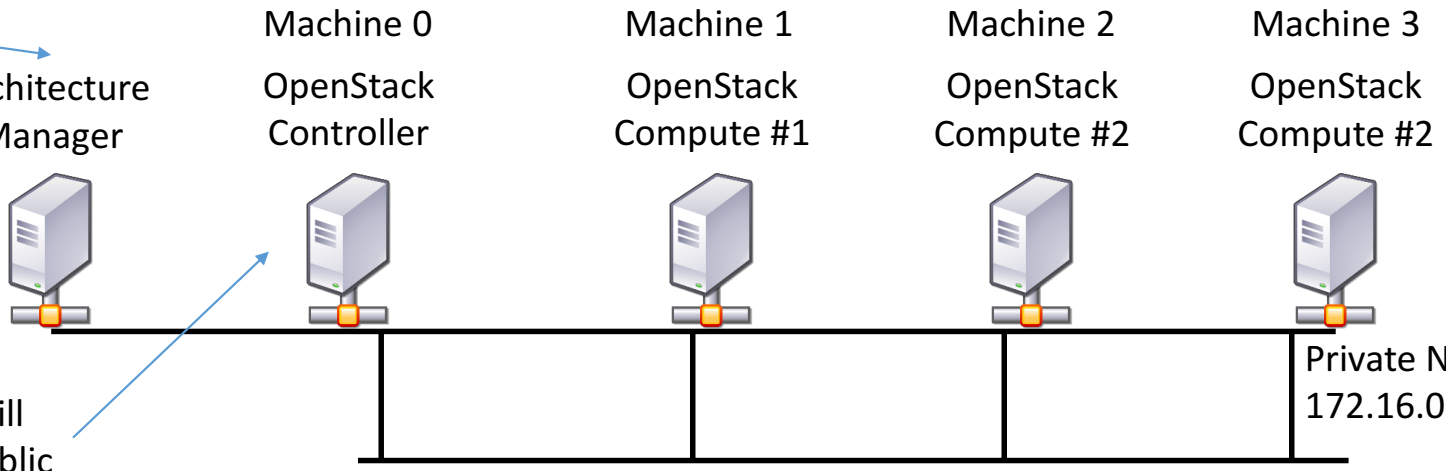
- JuJu openstack documentation https://docs.openstack.org/project-deploy-guide/charm-deployment-guide/latest/install-openstack.html

# VM Configuration

- Labs from now on need to be carried out in groups of 4 or 5
- Each VM will have a different role in the system resulting in the following overall architecture

The architecture manager Is a node that exposes to administrators a console to manage all the nodes of the platform

OpenStack controller (configured also as Network Node) and compute nodes will require with access to the public network (access network does not exist, we will emulate it)



# Network Configuration

- Each VM has only one physical network and it is connected to only one physical network
- This physical network will be the private network exploited by the machines to communicate to implement their functions
- The public physical machines will be emulated on top of the physical network using VLAN
- To this aim we need to modify the network configuration of each machine
- By default a VM has a pre-defined network configuration that is injected by the cloud platform into the VM
- We need to modify the network configuration so each VM has a static IP address and create at bootstrap a virtual network on top of the physical one using VLAN

# Netplan

Netplan is the tool adopted into Ubuntu 18.04 to manage the network configuration

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
        addresses:
        - 172.16.1.59/16
        gateway4: 172.16.0.1
        nameservers:
          addresses:
          - 8.8.8.8
```

```
vlans:
  vlan2:
    id: 2
    link: eth0
    dhcp4: no
    dhcp6: no
```

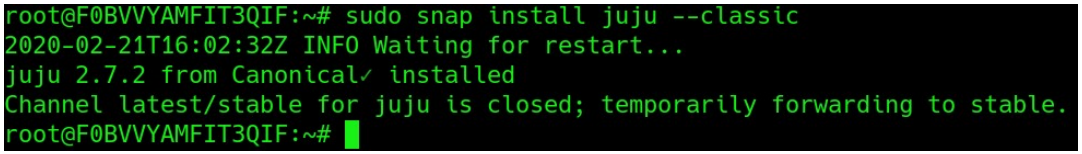
Vlan2 is a virtual network (manually created) to emulate the public network on top of the physical one

Apply the configuration to /etc/netplan/01-netcfg.yaml and run **`netplan apply`**

# JuJu – Installation

- Juju is an open source application tool developed by Canonical to facilitate quick deployment and configuration of public and private cloud services along bare metal servers or VMs
- Connect to the manager and install juju

```
sudo snap install juju --classic
```



# JuJu – Configuration

- Add new machines, since your VMs are existing machines that are like bare metal machines we perform manual installation, which is used every time machines are not managed by any cloud platform https://jaas.ai/docs/manual-cloud
- The steps to be performed are the following:
  - Add cloud
  - Bootstrap
  - Create a controller
  - Check status
  - Add machines

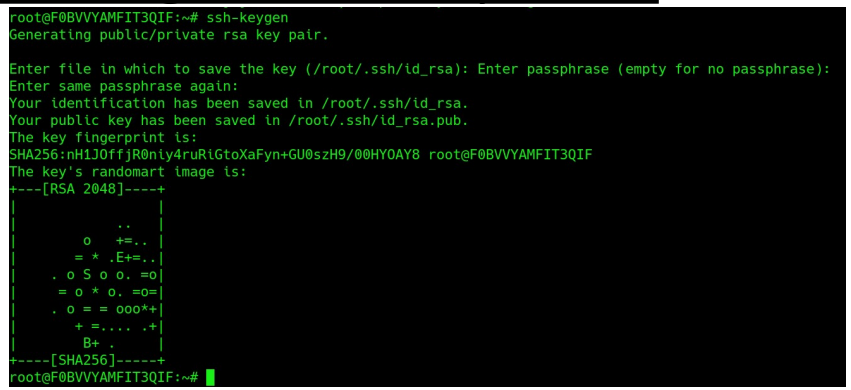
# Handle SSH keys

- Manual cloud installation require that the manager can connect to all the machines managed by the manager via ssh
- To this aim, ssh connection must be performed without password
- Create a set of ssh rsa key on the manager

```
ssh-keygen
```

- Add the key to all the machines (including the manager the host you are connected to!)

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@IP
```



# Add cloud

- Create a new cloud of manual type on the manager

  **`juju add-cloud`**

- **As IP of the controller select the Architecture manager** (the local machine you are connected to not the controller)

```
root@SNHOYM5GWPGME2L:~# juju add-cloud
Cloud Types
  lxd
  maas
  manual
  openstack
  vsphere

Select cloud type: manual

Enter a name for your manual cloud: mycloud

Enter the ssh connection string for controller, username@<hostname or IP> or <hostname or IP>: root@
172.16.0.128

This operation can be applied to both a copy on this client and to the one on a controller.
No current controller was detected and there are no registered controllers on this client: either bo
otstrap one or register one.
Do you ONLY want to add cloud "mycloud" to this client? (Y/n): Y

Cloud "mycloud" successfully added to your local client.
root@SNHOYM5GWPGME2L:~#
```

# Bootstrap

- Bootstrap the controller

  **`juju bootstrap mycloud manual-controller`**

It can take a while!

```
root@SNHOYM5GWPGME2L:~# juju bootstrap mycloud manual-controller
Creating Juju controller "manual-controller" on mycloud/default
Looking for packaged Juju agent version 2.7.0 for amd64
Installing Juju agent on bootstrap instance
Fetching Juju GUI 2.15.0
Running machine configuration script...

Bootstrap agent now started
Contacting Juju controller at 172.16.0.128 to verify accessibility...

Bootstrap complete, controller "manual-controller" now is available
Controller machines are in the "controller" model
Initial model "default" added
```

  **`juju status`**

```
root@SNHOYM5GWPGME2L:~# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.0    unsupported  08:29:29Z

Model "admin/default" is empty.
root@SNHOYM5GWPGME2L:~#
```

# Add machine

- Add all the remaining  machines (controller, and compute nodes)

  **`juju add-machine ssh:root@IP`**

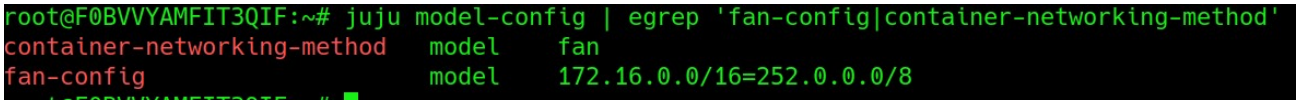
It can take a while!

```
root@SNHOYM5GWPGME2L:~# juju add-machine ssh:root@172.16.0.110
ubuntu:x:1001:
created machine 1
root@SNHOYM5GWPGME2L:~# juju add-machine ssh:root@172.16.0.111
ubuntu:x:1001:
created machine 2
root@SNHOYM5GWPGME2L:~# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.0    unsupported  08:36:28Z

Machine  State    DNS           Inst id              Series  AZ  Message
0        started  172.16.0.109  manual:172.16.0.109  bionic      Manually provisioned machine
1        started  172.16.0.110  manual:172.16.0.110  bionic      Manually provisioned machine
2        started  172.16.0.111  manual:172.16.0.111  bionic      Manually provisioned machine
```

# Configure the network for containers

- Some components will be deployed as container
  - Some module is developed to run as the only software installed on a machine, those has to be installed on a single machine, a container in this case, otherwise they will conflict each other (e.g. openstack-dashboard and nova-controller cannot run on the same machine)
- In order to have containers running on different machine to communicate each other we have to change the juju network configuration for container in order to use FAN, which is a software component that creates an overlay network across different containers running on different machine

```
juju model-config fan-config=172.16.0.0/16=252.0.0.0/8
juju model-config container-networking-method=fan
juju model-config | egrep 'fan-config|container-networking-method'
```




252.0.0.0/8 is the overlay network address

# Use already provisioned machines

- If you want to reuse a machine that has been provisioned previously (using juju) you need to run the following steps:

```
sudo rm -rf /var/lib/juju
sudo rm -rf /lib/systemd/system/juju*
sudo rm -rf /run/systemd/units/invocation:juju*
sudo rm -rf /etc/systemd/system/juju*
```

**Execute this only in case of errors, and you need to perform again one of the previous operations!**

- If you need to reset the architecture manager run:

```
rm -rf /root/.local/share/juju
```

# Storage

- Each VM has two hard drives attached, one for the OS (/dev/sda) and another additional hard drive for storage (/dev/sdb)
- Controller node will use the additional hard drive to install the platform software, compute nodes will use it as storage for VMs
- Using the hard drive on the **controller** node requires some manual steps (disks on the compute nodes will be managed automatically by the platform):
  - Create a new EXT4 partition
    - **`fdisk /dev/sdb`** (option n to create a new partition and then option w to write changes)
  - Format the partition
    - **`mkfs.ext4 /dev/sdb1`**
  - Mount the partition on the folder where JuJu will install the software running on the controller
    - **`mkdir -p /var/lib/lxd/storage-pools/default/`**
    - **`mount /dev/sdb1 /var/lib/lxd/storage-pools/default/`**
    - **`rm -rf /var/lib/lxd/storage-pools/default/lost+found`**
  - Add the following line to the file /etc/fstab to make the change permanent
    - **`/dev/sdb1 /var/lib/lxd/storage-pools/default ext4 errors=remount-ro 0 1`**

# Storage

```
root@NACH6ARD4Y7ZDQI:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x3bde43a3.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-83886079, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-83886079, default 83886079):

Created a new partition 1 of type 'Linux' and of size 40 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@NACH6ARD4Y7ZDQI:~#
```

```
root@NACH6ARD4Y7ZDQI:~# mkfs.ext4 /dev/sdb1
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 10485504 4k blocks and 2621440 inodes
Filesystem UUID: 0918352d-0398-47a3-b505-5872b4731b24
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

root@NACH6ARD4Y7ZDQI:~# mount /dev/sdb1 /var/lib/lxd/storage-pools/default/containers
mount: /var/lib/lxd/storage-pools/default/containers: mount point does not exist.
root@NACH6ARD4Y7ZDQI:~# mkdir -p /var/lib/lxd/storage-pools/default/containers
root@NACH6ARD4Y7ZDQI:~# mount /dev/sdb1 /var/lib/lxd/storage-pools/default/containers
```

# Manual OpenStack Installation

- In the following steps we will install OpenStack manually component by component
- For each component juju require a configuration, the number of instances to deploy and the machines to which deploy the instances
- Configuration is specified though a configuration file yaml
- The syntax of the command in general is the following

```
juju deploy --to ID_HOST --config config.yaml NAME_COMPONENT
```

- To deploy an additional instance of the component

```
juju add-unit --to ID_HOST NAME_COMPONENT
```

# Relations

- For each deployment juju allows to define relations
- A relation is a dependency between two components:
  - A configuration dependency, i.e. a parameter of a module is a configuration parameter of the other (e.g. the IP address of RabbitMQ is required by other modules like Nova)
  - An installation dependency, i.e. the installation of a module triggers automatically the installation of other modules
- The type of relation is specified by the module programmer
- A relation can be declared through this command:

```
juju add-relation COMPONENT1 COMPONENT2
```

# Nova Compute

- Deploy nova compute on the compute nodes {1,2,3}
- compute.yaml

```
nova-compute:
  enable-live-migration: True
  enable-resize: True
  migration-auth-type: ssh
  virt-type: qemu
  openstack-origin: cloud:bionic-train
```

Qemu (libvirt) will be the hypervisor

Version of the component

```
juju deploy --to 1 --config compute.yaml nova-compute
juju add-unit --to 2 nova-compute
```

# Nova Compute

```
root@F0BVVYAMFIT3QIF:~/config# juju deploy --to 1 --config compute.yaml nova-compute
Located charm "cs:nova-compute-312".
Deploying charm "cs:nova-compute-312".
root@F0BVVYAMFIT3QIF:~/config# juju add-unit --to 2 nova-compute

root@F0BVVYAMFIT3QIF:~/config#
root@F0BVVYAMFIT3QIF:~/config# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.2    unsupported  17:01:56Z

App           Version  Status   Scale  Charm         Store       Rev  OS      Notes
nova-compute           waiting    1/2  nova-compute  jujucharms  312  ubuntu

Unit             Workload     Agent       Machine  Public address  Ports  Message
nova-compute/0*  maintenance  executing   1        172.16.0.246           (install) installing charm software
nova-compute/1   waiting      allocating  2        172.16.0.248           agent initializing

Machine  State    DNS           Inst id              Series  AZ  Message
0        started  172.16.0.242  manual:172.16.0.242  bionic      Manually provisioned machine
1        started  172.16.0.246  manual:172.16.0.246  bionic      Manually provisioned machine
2        started  172.16.0.248  manual:172.16.0.248  bionic      Manually provisioned machine
```

# Neutron - Controller

- Neutron is composed of different components, running both on the controller and on the compute nodes
- Let's start with the components running in the controller
- neutron.yaml

```
neutron-gateway:
    bridge-mappings:          physnet1:br-ex
    data-port:                br-ex:vlan2
    openstack-origin: cloud:bionic-train

neutron-api:
    flat-network-providers:  physnet1
    overlay-network-type: gre
    neutron-security-groups: True
    openstack-origin: cloud:bionic-train
```

The port to the external network

Physnet1 won't adopt any virtualization mechanism (flat network)

Type of mechanism to create virtual networks over the same physical network (eth0 in this case)

# Neutron - Controller

- Deploy the components

```
juju deploy --to 0 --config neutron.yaml neutron-gateway
juju deploy --to lxd:0 --config neutron.yaml neutron-api
```

Neutron-api is deployed inside a container. Again, this component as others are developed starting from the assumption that they are installed on an entire system available only for them, if multiple components are installed on the same machine conflicts might arise

Neutron-gateway manages the connection with external networks. It must have direct access to the external interface, so it cannot be deployed in a container

# Neutron - Controller

```
root@E8C4KAONA4GYMWT:~/config# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.3    unsupported  17:53:42Z

App             Version  Status       Scale  Charm           Store       Rev  OS      Notes
neutron-api              waiting        0/1  neutron-api     jujucharms  283  ubuntu
neutron-gateway          maintenance      1  neutron-gateway jujucharms  279  ubuntu
nova-compute             maintenance      2  nova-compute    jujucharms  312  ubuntu

Unit               Workload     Agent       Machine  Public address  Ports  Message
neutron-api/0      waiting      allocating  0/lxd/0                         waiting for machine
neutron-gateway/0* maintenance  executing   0        172.16.1.114           (install) Installing apt packages
nova-compute/0*    maintenance  executing   1        172.16.1.115           (install) Installing apt packages
nova-compute/1     maintenance  executing   2        172.16.1.117           (install) Installing apt packages

Machine  State    DNS           Inst id              Series  AZ  Message
0        started  172.16.1.114  manual:172.16.1.114  bionic      Manually provisioned machine
0/lxd/0  pending                pending              bionic
1        started  172.16.1.115  manual:172.16.1.115  bionic      Manually provisioned machine
2        started  172.16.1.117  manual:172.16.1.117  bionic      Manually provisioned machine
3        started  172.16.1.118  manual:172.16.1.118  bionic      Manually provisioned machine
```

# Neutron - Compute

- Each compute node has a neutron instance that control openvswitch
- neutron.yaml

```
neutron-openvswitch:
    bridge-mappings:          physnet1:br-ex
    data-port:                br-ex:vlan2
    firewall-driver:          openvswitch
```

```
juju deploy --config neutron.yaml neutron-openvswitch
```

Deploy openvswitch and let juju select the nodes based on dependencies (it will be deployed on compute nodes)

# Neutron - Relations

```
juju add-relation neutron-api:neutron-plugin-api
neutron-gateway:neutron-plugin-api
juju add-relation neutron-api:neutron-plugin-api
neutron-openvswitch:neutron-plugin-api
juju add-relation neutron-openvswitch:neutron-
plugin nova-compute:neutron-plugin
```

# MySQL

- OpenStack requires a shared database, so we have to deploy a MySQL on the controller node {0} inside a container
- mysql.yaml

```
mysql:
  max-connections: 20000
  source: cloud:bionic-train
```

```
juju deploy --to lxd:0 --config mysql.yaml percona-cluster mysql
juju add-relation neutron-api:shared-db mysql:shared-db
```

# MySQL

```
root@E8C4KAONA4GYMWT:~/config# juju status
Model    Controller          Cloud/Region     Version  SLA          Timestamp
default  manual-controller   mycloud/default  2.7.3    unsupported  18:09:51Z

App                   Version  Status       Scale  Charm               Store       Rev  OS      Notes
mysql                          maintenance      1  percona-cluster     jujucharms  284  ubuntu
neutron-api                    maintenance      1  neutron-api         jujucharms  283  ubuntu
neutron-gateway       15.0.1   blocked          1  neutron-gateway     jujucharms  279  ubuntu
neutron-openvswitch   15.0.1   blocked          2  neutron-openvswitch jujucharms  273  ubuntu
nova-compute          20.0.1   blocked          2  nova-compute        jujucharms  312  ubuntu

Unit                      Workload     Agent      Machine  Public address  Ports  Message
mysql/0*                  maintenance  executing  0/lxd/1  252.1.114.82           (install) installing charm software
neutron-api/0*            maintenance  executing  0/lxd/0  252.1.114.25           (install) Installing apt packages
neutron-gateway/0*        blocked      idle       0        172.16.1.114           Missing relations: messaging
nova-compute/0*           blocked      idle       1        172.16.1.115           Missing relations: image, messaging
  neutron-openvswitch/0*  blocked      idle                172.16.1.115           Missing relations: messaging
nova-compute/1            blocked      idle       2        172.16.1.117           Missing relations: image, messaging
  neutron-openvswitch/1   blocked      idle                172.16.1.117           Missing relations: messaging

Machine  State    DNS           Inst id              Series  AZ  Message
0        started  172.16.1.114  manual:172.16.1.114  bionic      Manually provisioned machine
0/lxd/0  started  252.1.114.25  juju-6dd12f-0-lxd-0  bionic      Container started
0/lxd/1  started  252.1.114.82  juju-6dd12f-0-lxd-1  bionic      Container started
1        started  172.16.1.115  manual:172.16.1.115  bionic      Manually provisioned machine
2        started  172.16.1.117  manual:172.16.1.117  bionic      Manually provisioned machine
3        started  172.16.1.118  manual:172.16.1.118  bionic      Manually provisioned machine
```

# Keystone

- Deploy keystone on the controller node {0} in a container
- keystone.yaml

```
keystone:
  admin-password: openstack
  openstack-origin: cloud:bionic-train
```

```
juju deploy --to lxd:0 --config keystone.yaml keystone
juju add-relation keystone:shared-db mysql:shared-db
juju add-relation keystone:identity-service neutron-api:identity-service
```

# RabbitMQ

- Deploy RabbitMQ on the controller node {0} in a container

```
juju deploy --to lxd:0 rabbitmq-server
```

```
juju add-relation rabbitmq-server:amqp neutron-api:amqp
juju add-relation rabbitmq-server:amqp neutron-openvswitch:amqp
juju add-relation rabbitmq-server:amqp nova-compute:amqp
juju add-relation rabbitmq-server:amqp neutron-gateway:amqp
```

# Nova Controller

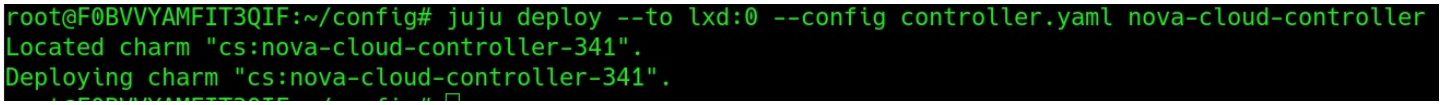
- Deploy nova controller on the controller node {0} in a container
- controller.yaml

```
nova-cloud-controller:
   network-manager: Neutron
   console-access-protocol: novnc
   console-proxy-ip: 172.16.3.26
   openstack-origin: cloud:bionic-train
```

IP of the controller that will exposes the dashboard

Protocol for the console connection

```
juju deploy --to lxd:0 --config controller.yaml
nova-cloud-controller
```



# Nova Controller

- Set the following dependencies

```
juju add-relation nova-cloud-controller:shared-db mysql:shared-db
juju add-relation nova-cloud-controller:identity-service keystone:identity-service
juju add-relation nova-cloud-controller:amqp rabbitmq-server:amqp
juju add-relation nova-cloud-controller:quantum-network-service neutron-gateway:quantum-network-service
juju add-relation nova-cloud-controller:neutron-api neutron-api:neutron-api
juju add-relation nova-cloud-controller:cloud-compute nova-compute:cloud-compute
```

# Placement

- Placement is a recent component that implement the nova scheduling functionalities
- Deploy a placement instance on the controller node {0} in a container

**placement.yaml**

```
placement:
  openstack-origin: cloud:bionic-train
```

```
juju deploy --to lxd:0 --config placement.yaml placement
```

```
juju add-relation placement:shared-db mysql:shared-db
juju add-relation placement:identity-service keystone:identity-service
juju add-relation placement:placement nova-cloud-controller:placement
```

# Horizon - Dashboard

- Deploy the dashboard on the controller node {0}. Deploy on the physical node in order to ensure that the dashboard is reachable from the network through the IP address on the controller node

IP of the controller that will exposes the dashboard

```
dashboard.yaml
openstack-dashboard:
  openstack-origin: cloud:bionic-train
  os-public-hostname: 172.16.3.26
```

```
juju deploy --to 0 --config dashboard.yaml openstack-dashboard
```

```
juju add-relation openstack-dashboard:identity-service
keystone:identity-service
```

# Glance

- Deploy glance on the controller node {0} in a container

**glance.yaml**

```
glance:
  openstack-origin: cloud:bionic-train
```

```
juju deploy --to lxd:0 --config glance.yaml glance
```

```
juju add-relation glance:image-service nova-cloud-controller:image-service
juju add-relation glance:image-service nova-compute:image-service
juju add-relation glance:shared-db mysql:shared-db
juju add-relation glance:identity-service keystone:identity-service
juju add-relation glance:amqp rabbitmq-server:amqp
```

# Cinder

- Deploy cinder on the controller node {0} in a container
- cinder.yaml

```
cinder:
  glance-api-version: 2
  block-device: None
  openstack-origin: cloud:bionic-train
```

```
juju deploy --to lxd:0 --config cinder.yaml cinder
```

```
juju add-relation cinder:cinder-volume-service nova-cloud-controller:cinder-volume-service
juju add-relation cinder:shared-db mysql:shared-db
juju add-relation cinder:identity-service keystone:identity-service
juju add-relation cinder:amqp rabbitmq-server:amqp
juju add-relation cinder:image-service glance:image-service
```

# Ceph

- Ceph is a specific module to create a distributed file system across different machines, to create a cloud file system
- Since OpenStack requires a file system to work (to store the images of the OS and the virtual disks of the VMs) we also deploy ceph
- We will cover ceph specifically later on, so in this lab you have to take it as a black box
- Ceph is composed of two components (OSD and MON) that are installed on each compute node
- Ceph will use the secondary hard drive of each compute node to create a distributed file system

# CephOSD

- Deploy the component CephOSD on each compute node {1,2,3}, the component has to be deployed on the bare metal as it needs access to the secondary drive

```
ceph-osd:
  osd-devices: /dev/sdb
  source: cloud:bionic-train

juju deploy --to 1 --config ceph-osd.yaml ceph-osd
juju add-unit --to 2 ceph-osd
juju add-unit --to 3 ceph-osd
```

# CephMON

- Deploy the component CephMON on each compute node {1,2,3}

```
ceph-mon:
source: cloud:bionic-train

juju deploy --to lxd:1 --config ceph-mon.yaml ceph-mon
juju add-unit --to lxd:2 ceph-mon
juju add-unit --to lxd:3 ceph-mon

juju add-relation ceph-mon:osd ceph-osd:mon
juju add-relation ceph-mon:client nova-compute:ceph
juju add-relation ceph-mon:client glance:ceph
juju add-relation cinder ceph-mon
```

# NTP

- Ceph requires NTP to synchronize the time across different nodes of the distributed file system

```
juju deploy ntp
```

```
juju add-relation ceph-osd:juju-info ntp:juju-info
```

# Final configuration

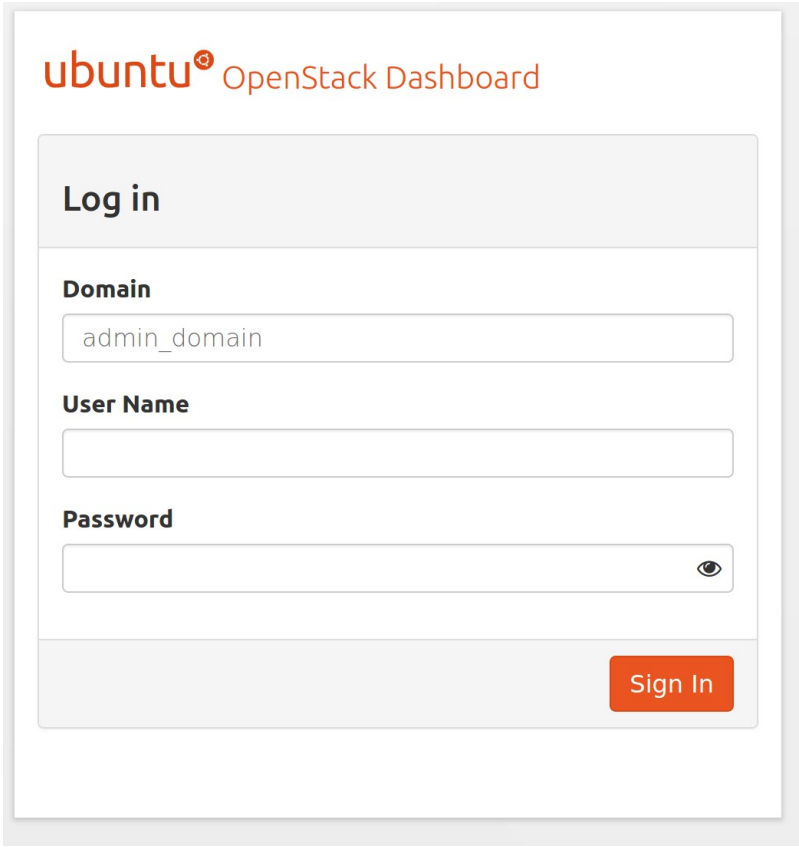
Retrieve the IP address of the dashboard via

```
juju status --format=yaml
openstack-dashboard | grep public-
address | awk '{print $2}'
```

```
root@F0BVVYAMFIT3QIF:~/config# juju status
Model    Controller         Cloud/Region     Version  SLA          Timestamp
default  manual-controller  mycloud/default  2.7.2    unsupported  13:57:51Z

App                    Version  Status  Scale  Charm                  Store       Rev  OS      Notes
ceph-mon               14.2.4   active      3  ceph-mon               jujucharms   45  ubuntu
ceph-osd               14.2.4   active      3  ceph-osd               jujucharms  298  ubuntu
cinder                 15.0.1   active      1  cinder                 jujucharms  300  ubuntu
glance                 19.0.0   active      1  glance                 jujucharms  294  ubuntu
keystone               16.0.0   active      1  keystone               jujucharms  310  ubuntu
mysql                  5.7.20   active      1  percona-cluster        jujucharms  284  ubuntu
neutron-api            15.0.1   active      1  neutron-api            jujucharms  283  ubuntu
neutron-gateway        15.0.1   active      1  neutron-gateway        jujucharms  279  ubuntu
neutron-openvswitch    15.0.1   active      3  neutron-openvswitch    jujucharms  273  ubuntu
nova-cloud-controller  20.0.1   active      1  nova-cloud-controller  jujucharms  341  ubuntu
nova-compute           20.0.1   active      3  nova-compute           jujucharms  312  ubuntu
ntp                    3.2      active      3  ntp                    jujucharms   38  ubuntu
openstack-dashboard    16.0.0   active      1  openstack-dashboard    jujucharms  299  ubuntu
placement              2.0.0    active      1  placement              jujucharms    4  ubuntu
rabbitmq-server        3.6.10   active      1  rabbitmq-server        jujucharms   99  ubuntu

Unit                      Workload  Agent  Machine    Public address  Ports              Message
ceph-mon/3*               active    idle   31         172.16.1.88                        Unit is ready and clustered
ceph-mon/4                active    idle   33         172.16.1.99                        Unit is ready and clustered
ceph-mon/5                active    idle   34         172.16.1.97                        Unit is ready and clustered
ceph-osd/0*               active    idle   31         172.16.1.88                        Unit is ready (1 OSD)
  ntp/0*                  active    idle              172.16.1.88     123/udp            chrony: Ready
ceph-osd/1                active    idle   33         172.16.1.99                        Unit is ready (1 OSD)
  ntp/1                   active    idle              172.16.1.99     123/udp            chrony: Ready
ceph-osd/2                active    idle   34         172.16.1.97                        Unit is ready (1 OSD)
  ntp/2                   active    idle              172.16.1.97     123/udp            chrony: Ready
cinder/9*                 active    idle   30/lxd/16  10.205.237.61   8776/tcp           Unit is ready
glance/12*                active    idle   30/lxd/15  10.205.237.213  9292/tcp           Unit is ready
keystone/12*              active    idle   30/lxd/10  10.205.237.26   5000/tcp           Unit is ready
mysql/11*                 active    idle   30/lxd/9   10.205.237.240  3306/tcp           Unit is ready
neutron-api/10*           active    idle   30/lxd/8   10.205.237.86   9696/tcp           Unit is ready
neutron-gateway/11*       active    idle   30         172.16.1.93                        Unit is ready
nova-cloud-controller/12* active    idle   30/lxd/12  10.205.237.206  8774/tcp,8775/tcp  Unit is ready
nova-compute/14*          active    idle   31         172.16.1.88                        Unit is ready
  neutron-openvswitch/8*  active    idle              172.16.1.88                        Unit is ready
nova-compute/15           active    idle   33         172.16.1.99                        Unit is ready
  neutron-openvswitch/9   active    idle              172.16.1.99                        Unit is ready
nova-compute/16           active    idle   34         172.16.1.97                        Unit is ready
  neutron-openvswitch/10  active    idle              172.16.1.97                        Unit is ready
openstack-dashboard/12*   active    idle   30         172.16.1.93     80/tcp,443/tcp     Unit is ready
placement/2*              active    idle   30/lxd/13  10.205.237.105  8778/tcp           Unit is ready
rabbitmq-server/9*        active    idle   30/lxd/11  10.205.237.95   5672/tcp           Unit is ready

Machine    State    DNS            Inst id               Series  AZ  Message
30         started  172.16.1.93    manual:172.16.1.93    bionic      Manually provisioned machine
30/lxd/8   started  10.205.237.86  juju-f92316-30-lxd-8  bionic      Container started
30/lxd/9   started  10.205.237.240 juju-f92316-30-lxd-9  bionic      Container started
30/lxd/10  started  10.205.237.26  juju-f92316-30-lxd-10 bionic      Container started
30/lxd/11  started  10.205.237.95  juju-f92316-30-lxd-11 bionic      Container started
30/lxd/12  started  10.205.237.206 juju-f92316-30-lxd-12 bionic      Container started
30/lxd/13  started  10.205.237.105 juju-f92316-30-lxd-13 bionic      Container started
30/lxd/15  started  10.205.237.213 juju-f92316-30-lxd-15 bionic      Container started
30/lxd/16  started  10.205.237.61  juju-f92316-30-lxd-16 bionic      Container started
31         started  172.16.1.88    manual:172.16.1.88    bionic      Manually provisioned machine
33         started  172.16.1.99    manual:172.16.1.99    bionic      Manually provisioned machine
34         started  172.16.1.97    manual:172.16.1.97    bionic      Manually provisioned machine
```

# Connect to the dashboard

- Open the browser to the URL:
  - http://CONTROLLER_IP/horizon/auth/login/




- Domain: admin_domain
- User Name: admin
- Password: openstack

# Check that compute nodes are working

- Go to
  - Admin-> System -> System Information -> Compute Service