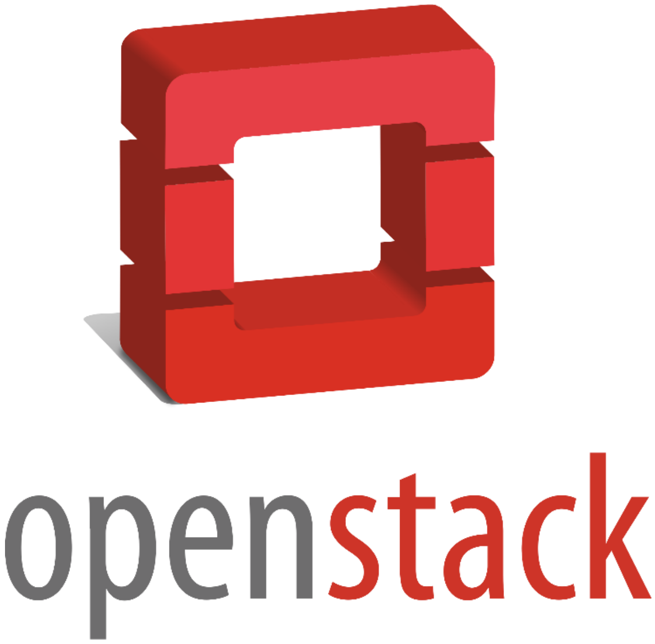
ROYAUME DU MAROC
UNIVERSITE ABDELMALEK ESSAÂDI
FACULTE DES SCIENCES ET TECHNIQUES
DEPARTEMENT GENIE INFORMATIQUE

IaaS & SaaS implementation with OpenStack



- **Realized by:** AMAYOU ANAS
- **Supervised by:** Chaker El amrani

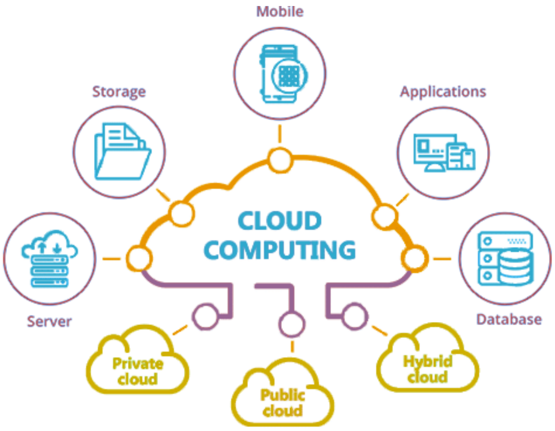
Summary :

1. General introduction : cloud computing & openstack .

➢ What is cloud computing ?

cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping you lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change.

➢ **Benefits of** cloud computing :



- One can access applications as utilities, over the Internet.
- Manipulate and configure the application online at any time.
- It does not require to install a specific piece of software to access or manipulate cloud application.
- Cloud Computing offers on-demand self-service. The resources can be used without interaction with cloud service provider.
- Cloud Computing offers load balancing that makes it more reliable.

➢ **Risks of Cloud Computing :**

Although Cloud Computing is a great innovation in the world of computing, there also exist downsides of cloud computing. Some of them are discussed below:

- **SECURITY & PRIVACY :**

It is the biggest concern about cloud computing. Since data management and infrastructure management in cloud is provided by third-party, it is always a risk to handover the sensitive information to such providers.

- **LOCK-IN :**

It is very difficult for the customers to switch from one Cloud Service Provider (CSP) to another. It results in dependency on a particular CSP for service.

- **ISOLATION FAILURE :**

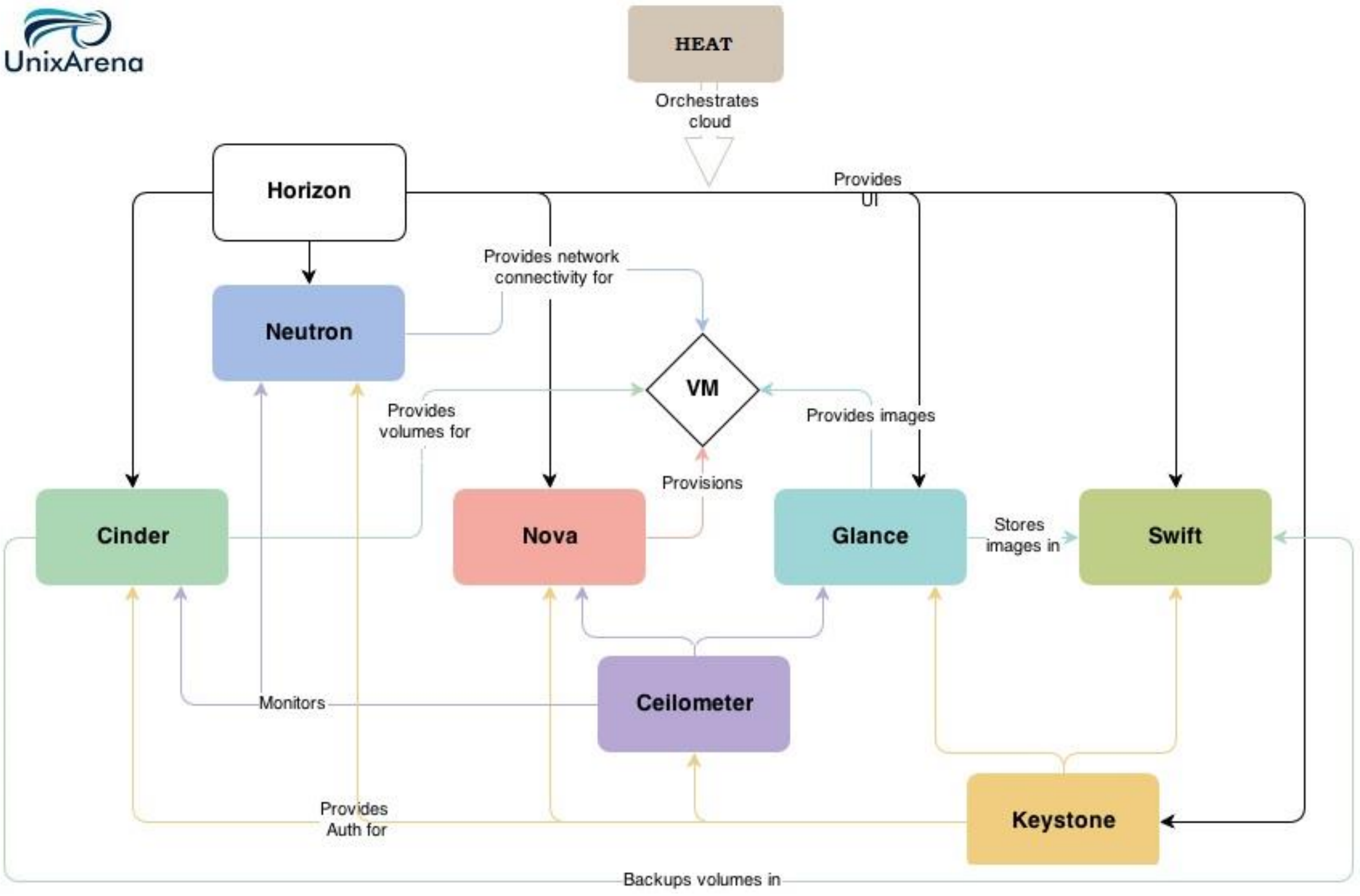
This risk involves the failure of isolation mechanism that separates storage, memory, routing between the different tenants.

✚ What is **OPENSTACK?**

- OpenStack is an open cloud computing software platform that allows the users to create Infrastructure as a Service (IaaS) cloud environments suited for all types of deployments and environments (prod, pre-prod, test, dev, etc.).
- It can provide the baseline for future integration of different modules, e.g., between OpenStack and OpenDaylight (Open Source Software Defined Networking Platform), where OpenDaylight is used to create networking services, together with the default module provided by OpenStack (called Neutron).
- OpenStack is an open source software giving open access to the source code, to make any changes or modifications needed, and freely sharing these changes back out with the community at large .

OPENSTACK Architecture :

The above architecture presents the OpenStack Services and how they communicate. Further, the paper will speak shortly about each presented service. OpenStack embraces a modular architecture to provide a set of core services that facilitate scalability and elasticity as core design tenets.



2. the reasons for choosing the OpenStack implementation method: Devstack.

To implement this project , I use the devstack method that helps me to install openstack on ubuntu distribution . DevStack is a series of extensible scripts used to quickly bring up a complete OpenStack environment based on the latest versions of everything from git master.

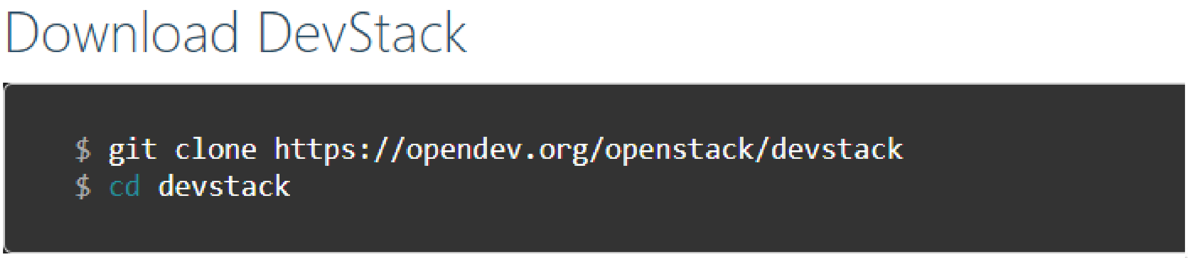
- It is used interactively as a development environment and as the basis for much of the OpenStack project’s functional testing.

- It can also be used to demonstrate starting/running OpenStack services and provide examples of using them from a command line.

- **click here for official devstack documentation :**

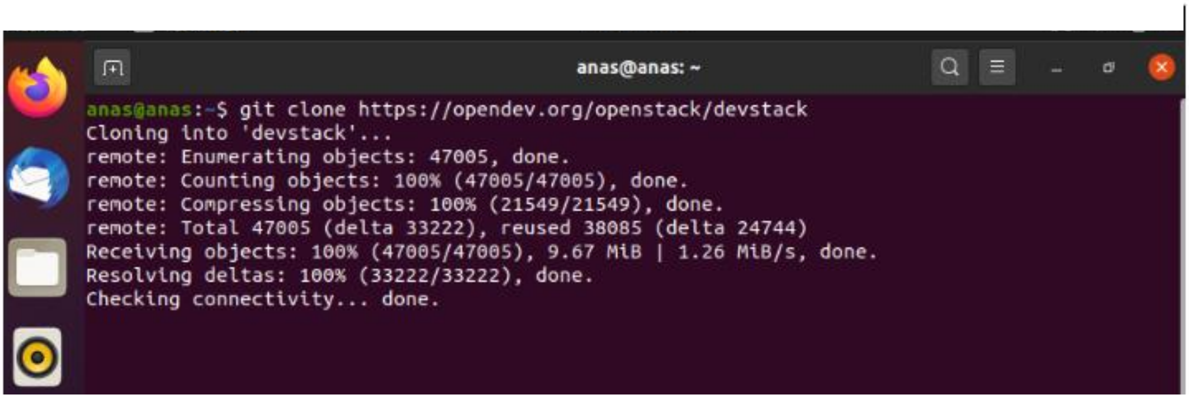
3. The installation steps.

To install opensatck , I used ubunut (20.04) installed in virtualBox .



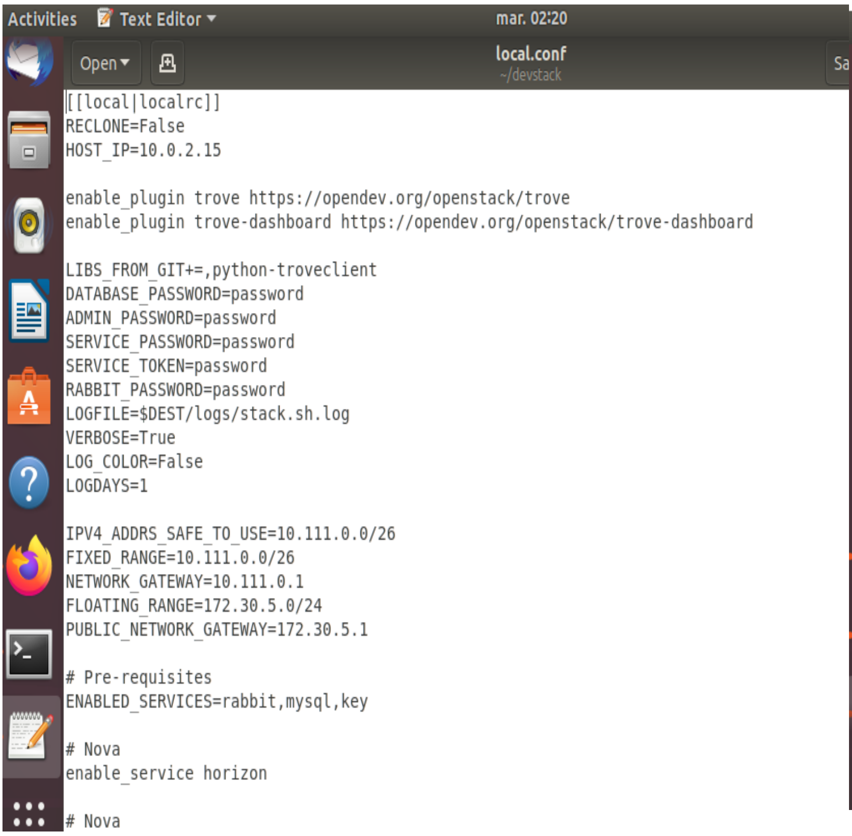
Download DevStack

```
$ git clone https://opendev.org/openstack/devstack
$ cd devstack
```

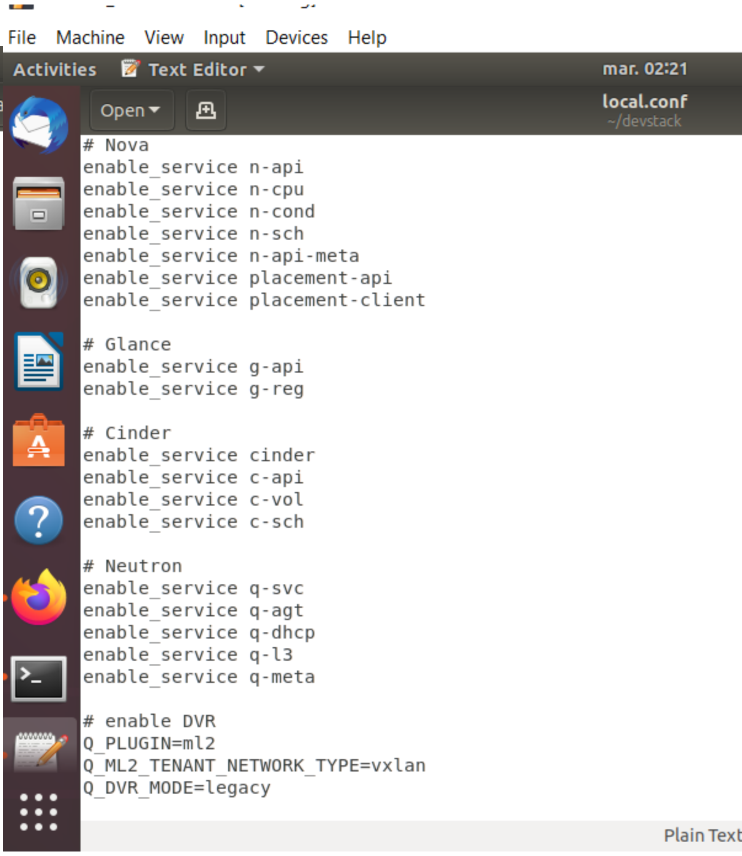


```
anas@anas:~$ git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 47005, done.
remote: Counting objects: 100% (47005/47005), done.
remote: Compressing objects: 100% (21549/21549), done.
remote: Total 47005 (delta 33222), reused 38085 (delta 24744)
Receiving objects: 100% (47005/47005), 9.67 MiB | 1.26 MiB/s, done.
Resolving deltas: 100% (33222/33222), done.
Checking connectivity... done.
```

- **Create local.conf (in devstack folder) :**



```
[[local|localrc]]
RECLONE=False
HOST_IP=10.0.2.15

enable_plugin trove https://opendev.org/openstack/trove
enable_plugin trove-dashboard https://opendev.org/openstack/trove-dashboard

LIBS_FROM_GIT+=,python-troveclient
DATABASE_PASSWORD=password
ADMIN_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
RABBIT_PASSWORD=password
LOGFILE=$DEST/logs/stack.sh.log
VERBOSE=True
LOG_COLOR=False
LOGDAYS=1

IPV4_ADDRS_SAFE_TO_USE=10.111.0.0/26
FIXED_RANGE=10.111.0.0/26
NETWORK_GATEWAY=10.111.0.1
FLOATING_RANGE=172.30.5.0/24
PUBLIC_NETWORK_GATEWAY=172.30.5.1

# Pre-requisites
ENABLED_SERVICES=rabbit,mysql,key

# Nova
enable_service horizon

# Nova
```



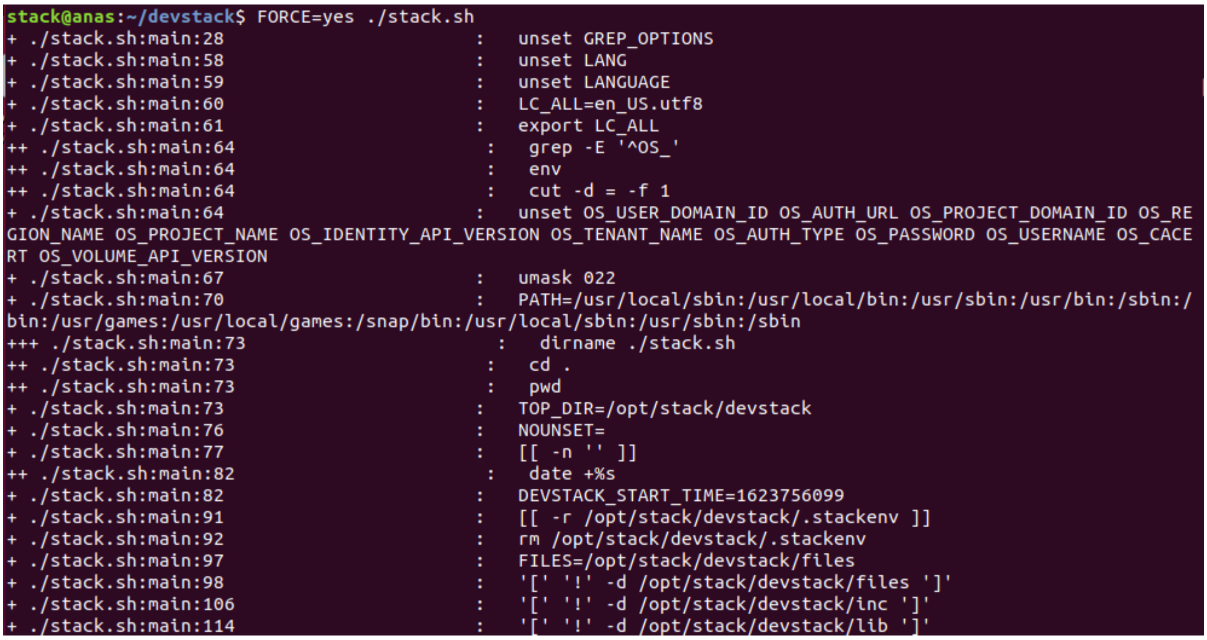
```
# Nova
enable_service n-api
enable_service n-cpu
enable_service n-cond
enable_service n-sch
enable_service n-api-meta
enable_service placement-api
enable_service placement-client

# Glance
enable_service g-api
enable_service g-reg

# Cinder
enable_service cinder
enable_service c-api
enable_service c-vol
enable_service c-sch

# Neutron
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta

# enable DVR
Q_PLUGIN=ml2
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_DVR_MODE=legacy
```

→**local.conf content example :**

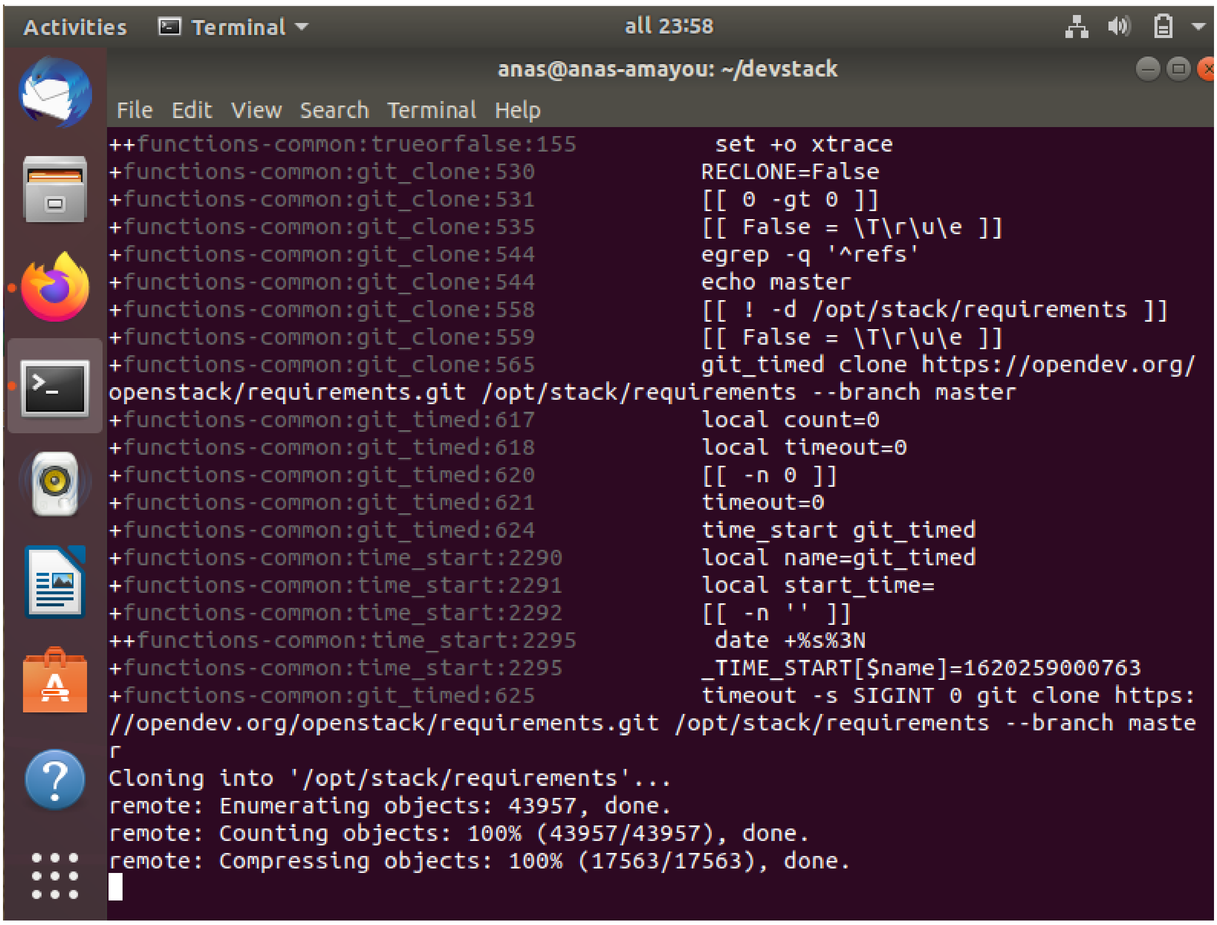
```
[[local|localrc]] # Password for KeyStone, Database, RabbitMQ and
Service ADMIN_PASSWORD=StrongAdminSecret
DATABASE_PASSWORD=$ADMIN_PASSWORD RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD # Host IP - get your Server/VM IP
address from ip addr command HOST_IP=10.208.0.10
```

➔ **After we lunch the installation commant : ./stack.sh (it take some time)**

```
./stack.sh
```

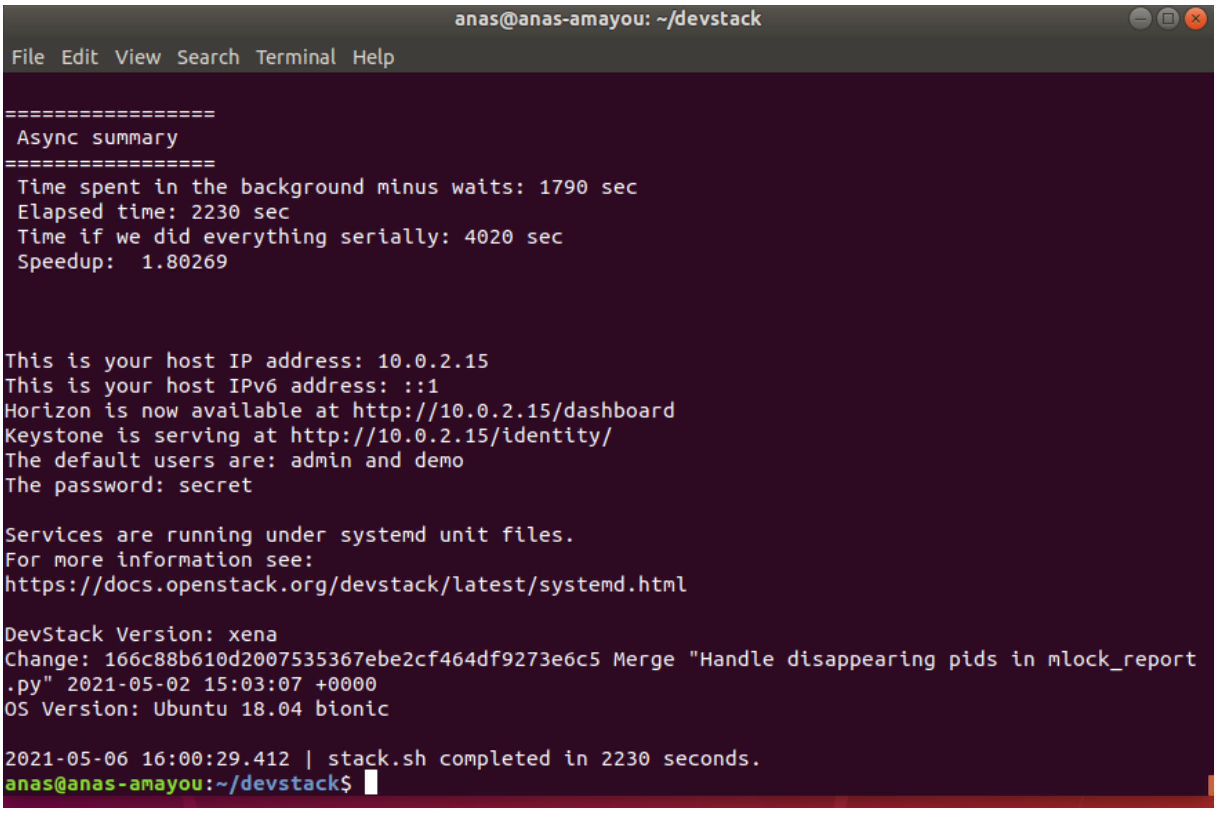


```
stack@anas:~/devstack$ FORCE=yes ./stack.sh
+ ./stack.sh:main:28                       :   unset GREP_OPTIONS
+ ./stack.sh:main:58                       :   unset LANG
+ ./stack.sh:main:59                       :   unset LANGUAGE
+ ./stack.sh:main:60                       :   LC_ALL=en_US.utf8
+ ./stack.sh:main:61                       :   export LC_ALL
++ ./stack.sh:main:64                       :   grep -E '^OS_'
++ ./stack.sh:main:64                       :   env
++ ./stack.sh:main:64                       :   cut -d = -f 1
+ ./stack.sh:main:64                       :   unset OS_USER_DOMAIN_ID OS_AUTH_URL OS_PROJECT_DOMAIN_ID OS_RE
GION_NAME OS_PROJECT_NAME OS_IDENTITY_API_VERSION OS_TENANT_NAME OS_AUTH_TYPE OS_PASSWORD OS_USERNAME OS_CACE
RT OS_VOLUME_API_VERSION
+ ./stack.sh:main:67                       :   umask 022
+ ./stack.sh:main:70                       :   PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/sbin:/usr/sbin:/sbin
+++ ./stack.sh:main:73                       :   dirname ./stack.sh
++ ./stack.sh:main:73                       :   cd .
++ ./stack.sh:main:73                       :   pwd
+ ./stack.sh:main:73                       :   TOP_DIR=/opt/stack/devstack
+ ./stack.sh:main:76                       :   NOUNSET=
+ ./stack.sh:main:77                       :   [[ -n '' ]]
++ ./stack.sh:main:82                       :   date +%s
+ ./stack.sh:main:82                       :   DEVSTACK_START_TIME=1623756099
+ ./stack.sh:main:91                       :   [[ -r /opt/stack/devstack/.stackenv ]]
+ ./stack.sh:main:92                       :   rm /opt/stack/devstack/.stackenv
+ ./stack.sh:main:97                       :   FILES=/opt/stack/devstack/files
+ ./stack.sh:main:98                       :   '[' '!' -d /opt/stack/devstack/files ']'
+ ./stack.sh:main:106                      :   '[' '!' -d /opt/stack/devstack/inc ']'
+ ./stack.sh:main:114                      :   '[' '!' -d /opt/stack/devstack/lib ']'
```


Activities
Terminal
all 23:58
anas@anas-amayou: ~/devstack
File Edit View Search Terminal Help

```
++functions-common:trueorfalse:155          set +o xtrace
+functions-common:git_clone:530            RECLONE=False
+functions-common:git_clone:531            [[ 0 -gt 0 ]]
+functions-common:git_clone:535            [[ False = \T\r\u\e ]]
+functions-common:git_clone:544            egrep -q '^refs'
+functions-common:git_clone:544            echo master
+functions-common:git_clone:558            [[ ! -d /opt/stack/requirements ]]
+functions-common:git_clone:559            [[ False = \T\r\u\e ]]
+functions-common:git_clone:565            git_timed clone https://opendev.org/
openstack/requirements.git /opt/stack/requirements --branch master
+functions-common:git_timed:617            local count=0
+functions-common:git_timed:618            local timeout=0
+functions-common:git_timed:620            [[ -n 0 ]]
+functions-common:git_timed:621            timeout=0
+functions-common:git_timed:624            time_start git_timed
+functions-common:time_start:2290          local name=git_timed
+functions-common:time_start:2291          local start_time=
+functions-common:time_start:2292          [[ -n '' ]]
++functions-common:time_start:2295          date +%s%3N
+functions-common:time_start:2295          _TIME_START[$name]=1620259000763
+functions-common:git_timed:625            timeout -s SIGINT 0 git clone https:
//opendev.org/openstack/requirements.git /opt/stack/requirements --branch maste
r
Cloning into '/opt/stack/requirements'...
remote: Enumerating objects: 43957, done.
remote: Counting objects: 100% (43957/43957), done.
remote: Compressing objects: 100% (17563/17563), done.
```

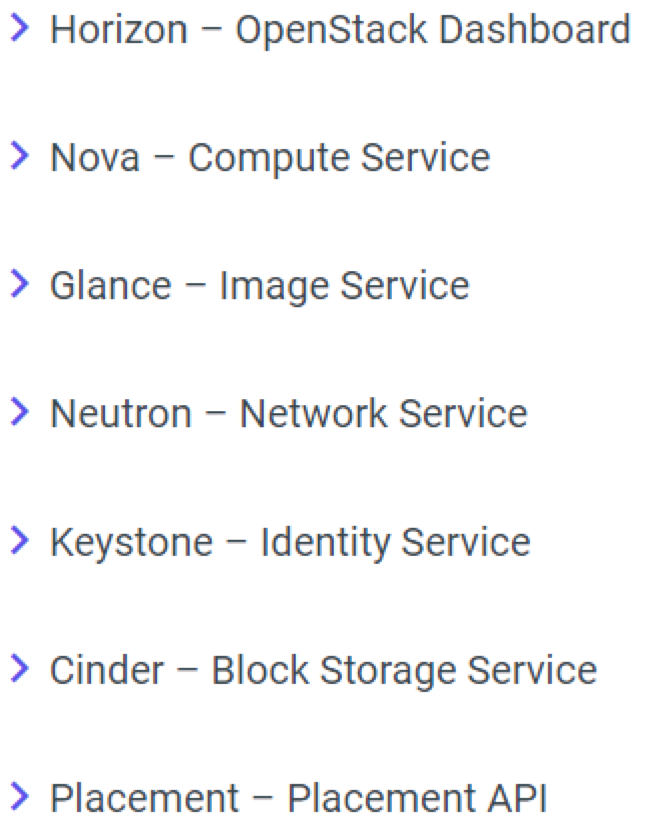
- ➢ **Finally , openstack installed , and there is my ip address :**


anas@anas-amayou: ~/devstack
File Edit View Search Terminal Help

```

=================
 Async summary
=================
 Time spent in the background minus waits: 1790 sec
 Elapsed time: 2230 sec
 Time if we did everything serially: 4020 sec
 Speedup:  1.80269



This is your host IP address: 10.0.2.15
This is your host IPv6 address: ::1
Horizon is now available at http://10.0.2.15/dashboard
Keystone is serving at http://10.0.2.15/identity/
The default users are: admin and demo
The password: secret

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: xena
Change: 166c88b610d2007535367ebe2cf464df9273e6c5 Merge "Handle disappearing pids in mlock_report
.py" 2021-05-02 15:03:07 +0000
OS Version: Ubuntu 18.04 bionic

2021-05-06 16:00:29.412 | stack.sh completed in 2230 seconds.
anas@anas-amayou:~/devstack$
```

4. The installed modules and their roles.

The following features are installed:



- Horizon – OpenStack Dashboard
- Nova – Compute Service
- Glance – Image Service
- Neutron – Network Service
- Keystone – Identity Service
- Cinder – Block Storage Service
- Placement – Placement API

- **Now we will explain the role of each one :**

1. Dashboard (Horizon)

Horizon is the authorized implementation of OpenStack’s Dashboard, which is the only graphical interface to automate cloud-based resources. To service providers and other commercial vendors, it supports with third party services such as monitoring, billing, and other management tools. Developers can automate tools to manage OpenStack resources using EC2 compatibility API or the native OpenStack API.

2. Compute (Nova)

OpenStack Compute is a cloud computing fabric controller, which manages pools of computer resources and work with virtualization technologies, bare metals, and high-performance computing configurations. Nova's architecture provides flexibility to design the cloud with no proprietary software or hardware requirements and also delivers the ability to integrate the legacy systems and third-party products.

3. Image Service (Glance)

OpenStack image service offers discovering, registering, and restoring virtual machine images. Glance has client-server architecture and delivers a user REST API, which allows querying of virtual machine image metadata and also retrieval of the actual image. While deploying new virtual machine instances, Glance uses the stored images as templates.

4. Identity Service (Keystone)

Keystone provides a central list of users, mapped against all the OpenStack services, which they can access. It integrates with existing backend services such as LDAP while acting as a common authentication system across the cloud computing system.

5. Block Storage (Cinder)

OpenStack Cinder delivers determined block-level storage devices for application with OpenStack compute instances. A cloud user can manage their storage needs by integrating block storage volumes with Dashboard and Nova.

5. Details and tests on the IaaS created.

The implemented architecture shows the OpenStack services, as components used to implement the OpenStack cloud IaaS Environment and its capabilities. The capabilities are presented in Figure 3, with dotted-lines emphasizing future integration and work that can be done based on the created environment:

➢ Iaas Implementation steps :

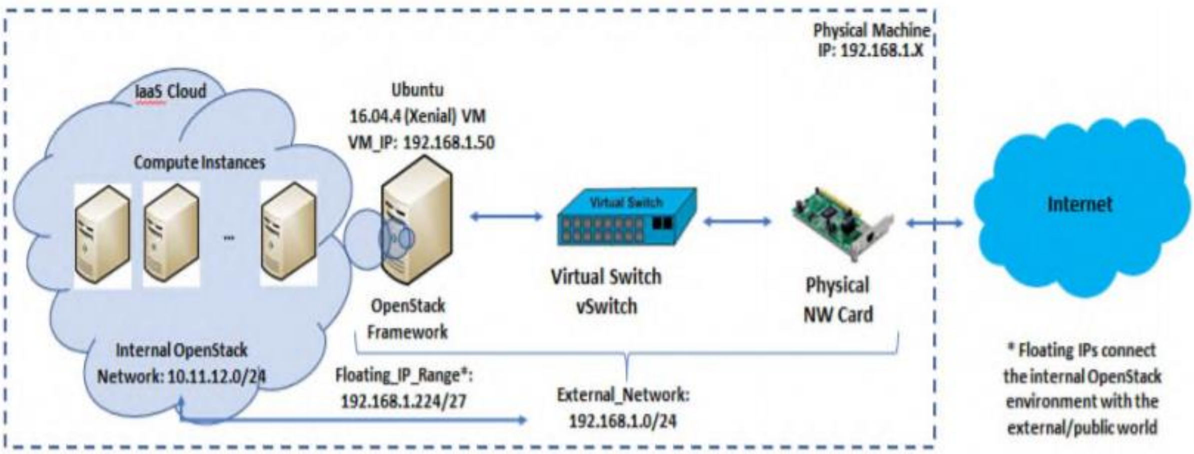
After the creation of the IaaS Environment, the following tests can be performed:

- Ping the Compute instances from the external network (Internet Service Provider network in this case). This will be done from the Ubuntu machine;

- Connect to the Compute instances, using SSH, from the external network. This will be done from the Ubuntu machine;

- Test the connection between the Compute instances;

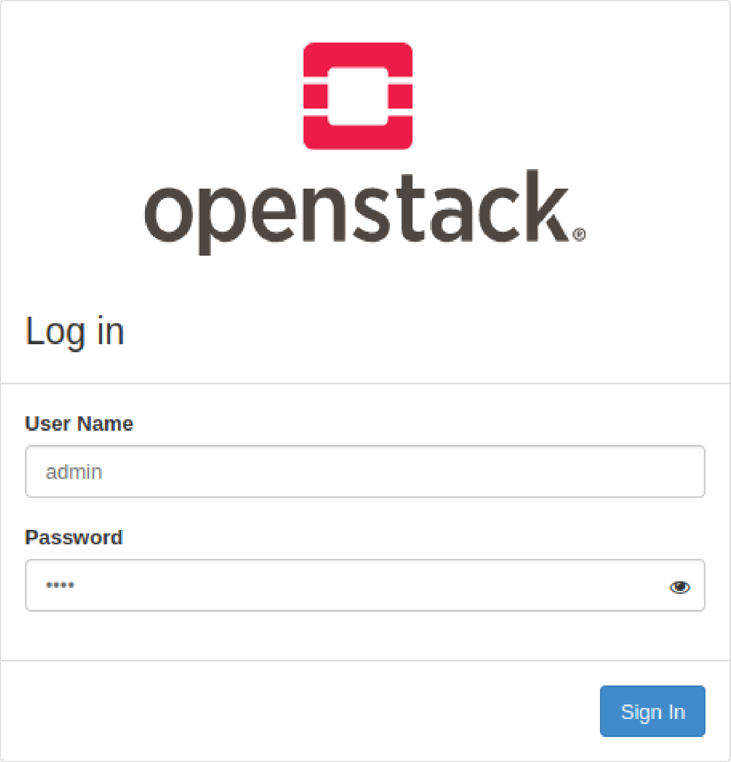
- Test if the Compute instances have access to the Internet. This will be tested by issuing a ping command to "google.com", from the Compute instances.



Iaas deployment components

➢ Iaas deployment steps (detailed) :

a) Login to the OpenStack Dashboard



Log in

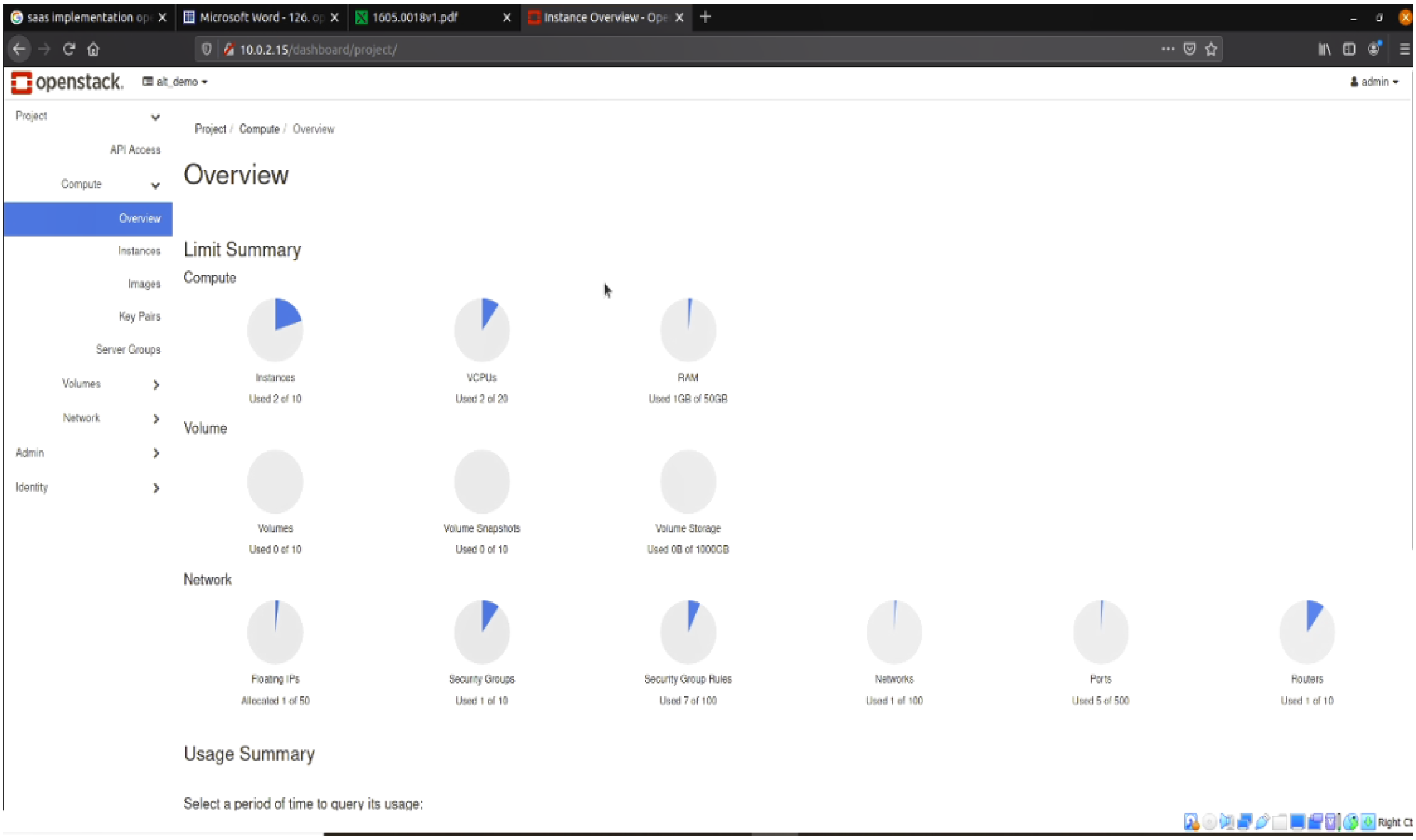
User Name

admin

Password

••••

→ After login , this is my dashboard : Horizon



10.0.2.15/dashboard/project/

Project / Compute / Overview

Overview

Limit Summary

Compute

Instances
Used 2 of 10

VCPUs
Used 2 of 20

RAM
Used 1GB of 50GB

Volume

Volumes
Used 0 of 10

Volume Snapshots
Used 0 of 10

Volume Storage
Used 0B of 1000GB

Network

Floating IPs
Allocated 1 of 50

Security Groups
Used 1 of 10

Security Group Rules
Used 7 of 100

Networks
Used 1 of 100

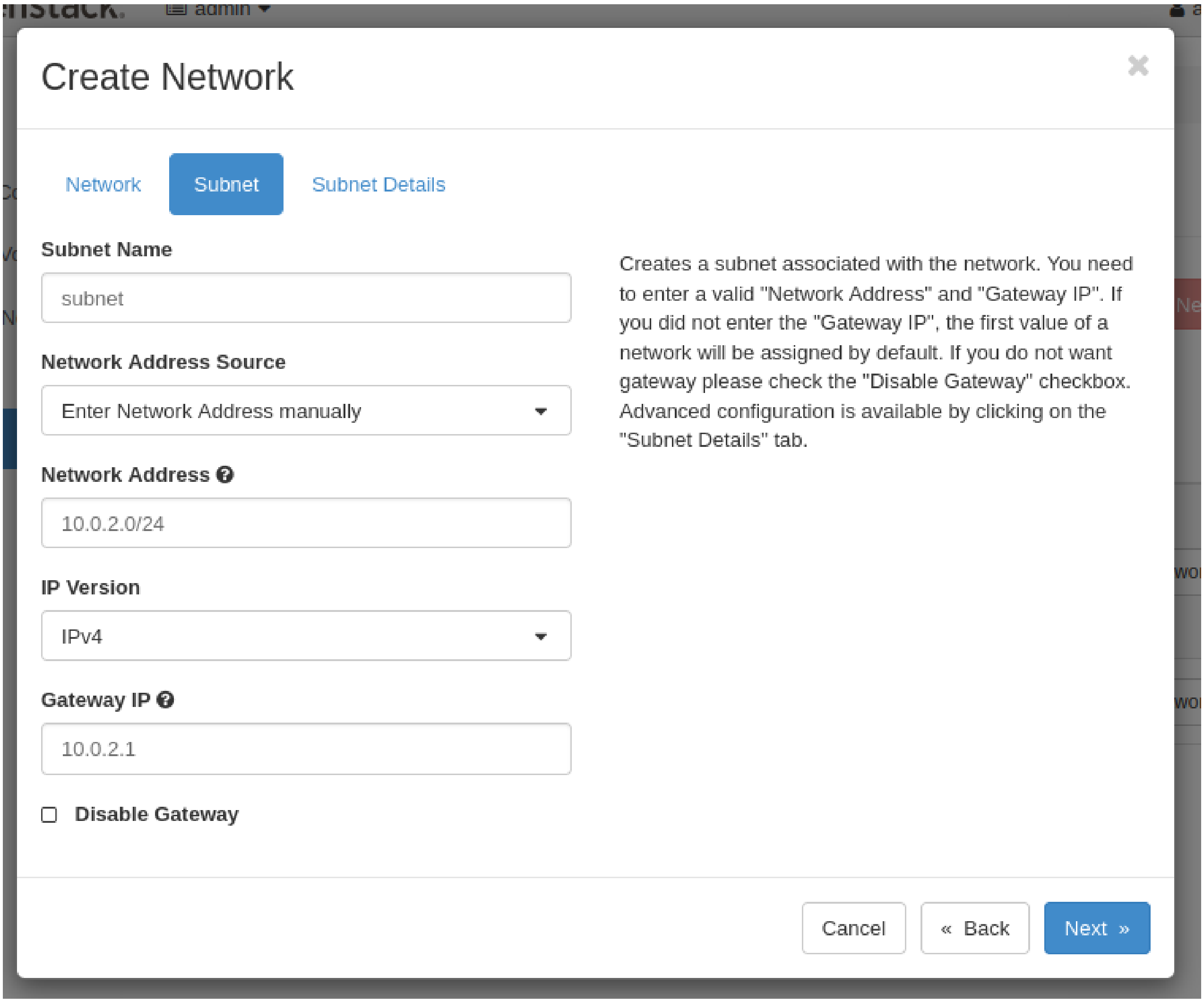
Ports
Used 5 of 500

Routers
Used 1 of 10

Usage Summary

Select a period of time to query its usage:

b) First, create an internal network using the subnet specified in the local.conf file



Subnet Name: subnet

Network Address Source: Enter Network Address manually

Network Address: 10.0.2.0/24

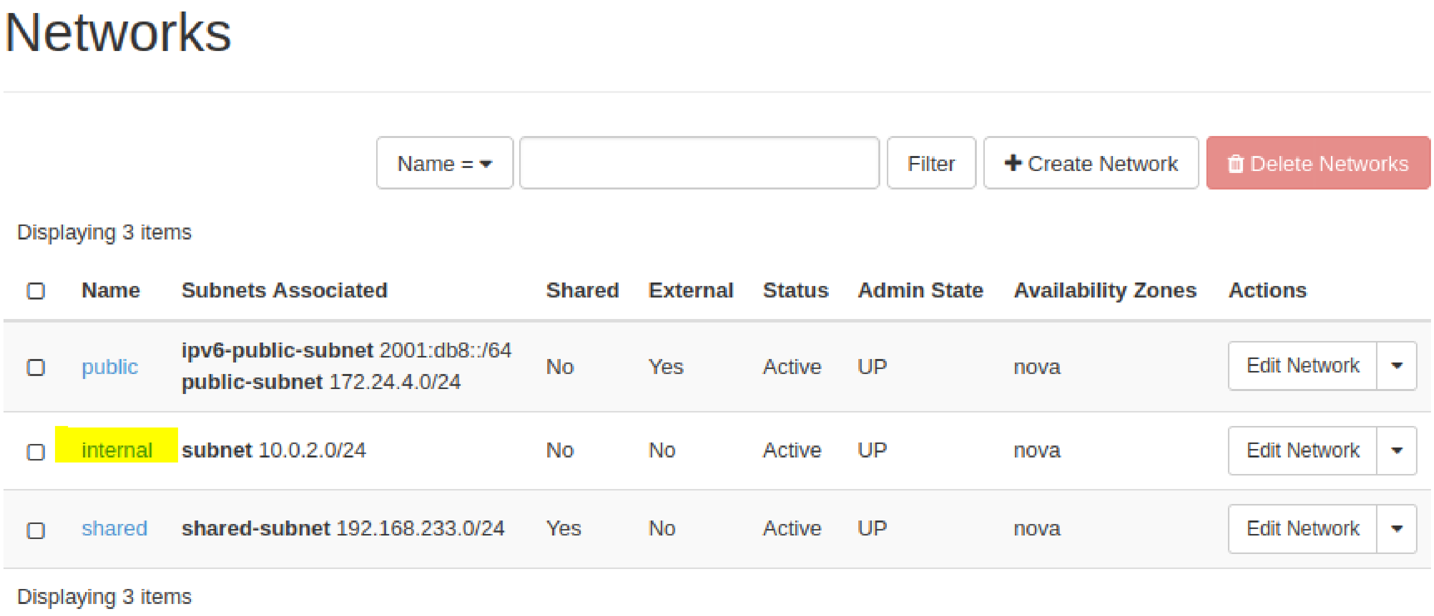
IP Version: IPv4

Gateway IP: 10.0.2.1

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

- ➢ **Network created :**

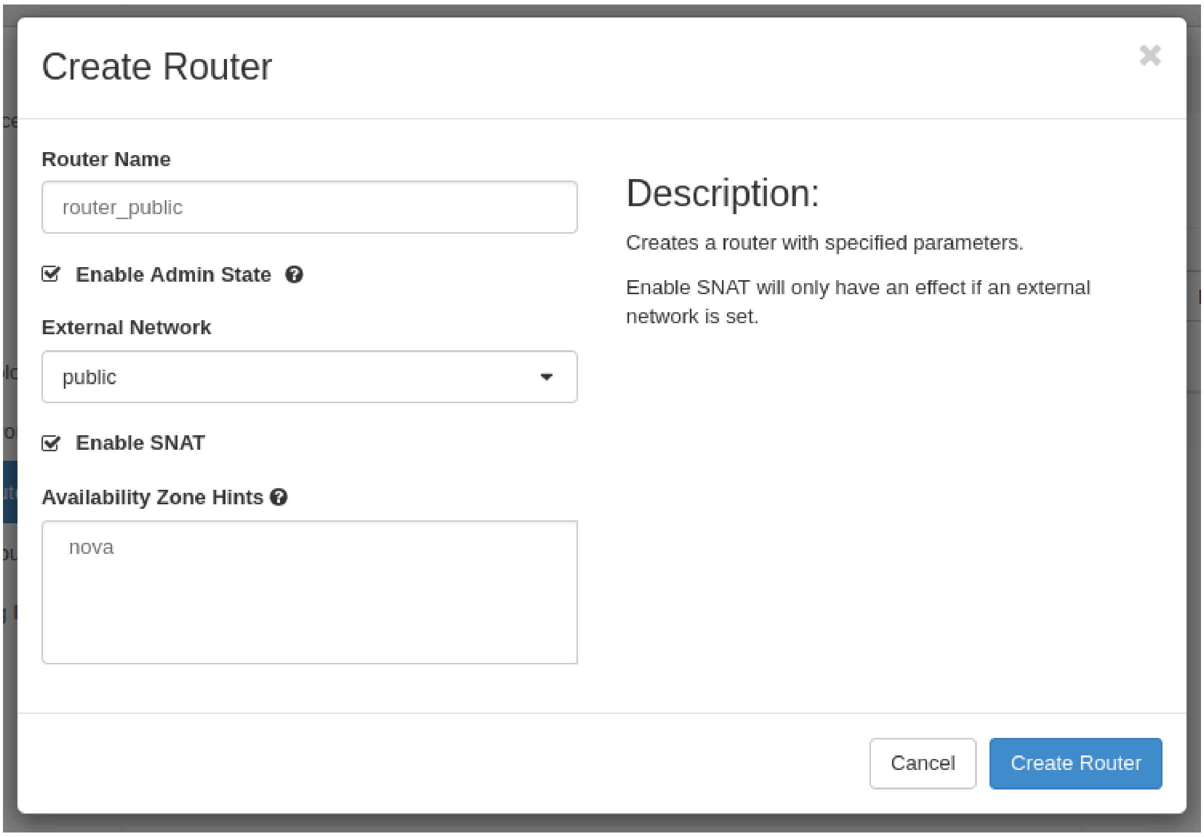


Displaying 3 items

□	Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
□	public	**ipv6-public-subnet** 2001:db8::/64 **public-subnet** 172.24.4.0/24	No	Yes	Active	UP	nova	Edit Network ▾
□	internal	**subnet** 10.0.2.0/24	No	No	Active	UP	nova	Edit Network ▾
□	shared	**shared-subnet** 192.168.233.0/24	Yes	No	Active	UP	nova	Edit Network ▾

Displaying 3 items

c) Create a Router that will route the traffic in the internal network :



Create Router

Router Name

router_public

☑ **Enable Admin State**

External Network

public

☑ **Enable SNAT**

Availability Zone Hints

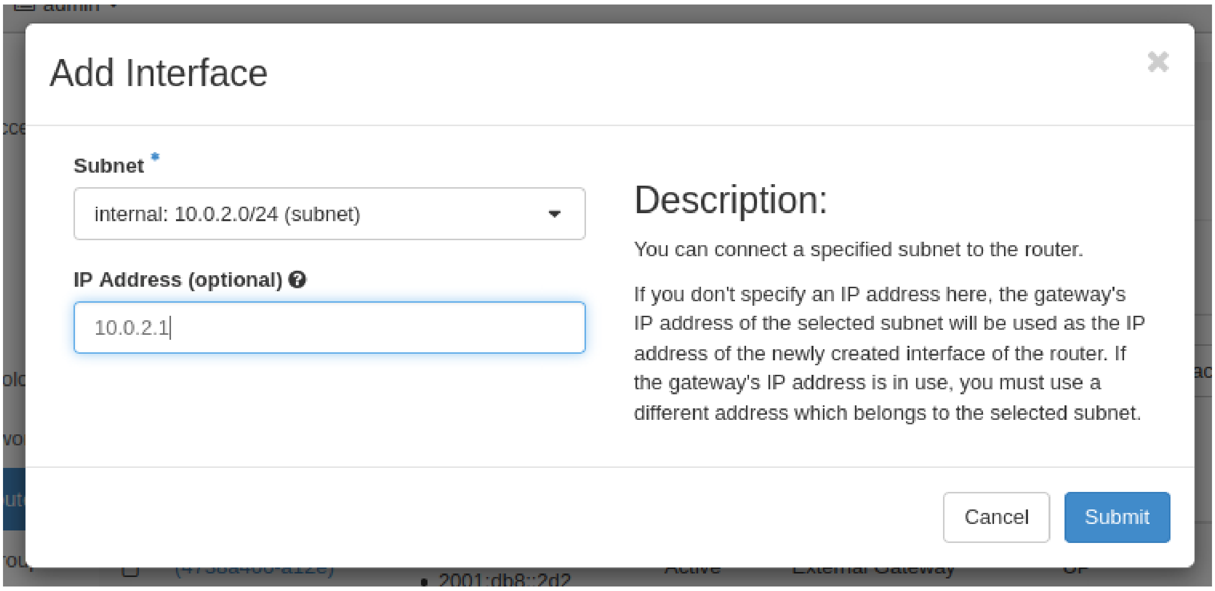
nova

Description:

Creates a router with specified parameters.

Enable SNAT will only have an effect if an external network is set.

➢ Add interface to the router :



Add Interface

Subnet *

internal: 10.0.2.0/24 (subnet)

IP Address (optional)

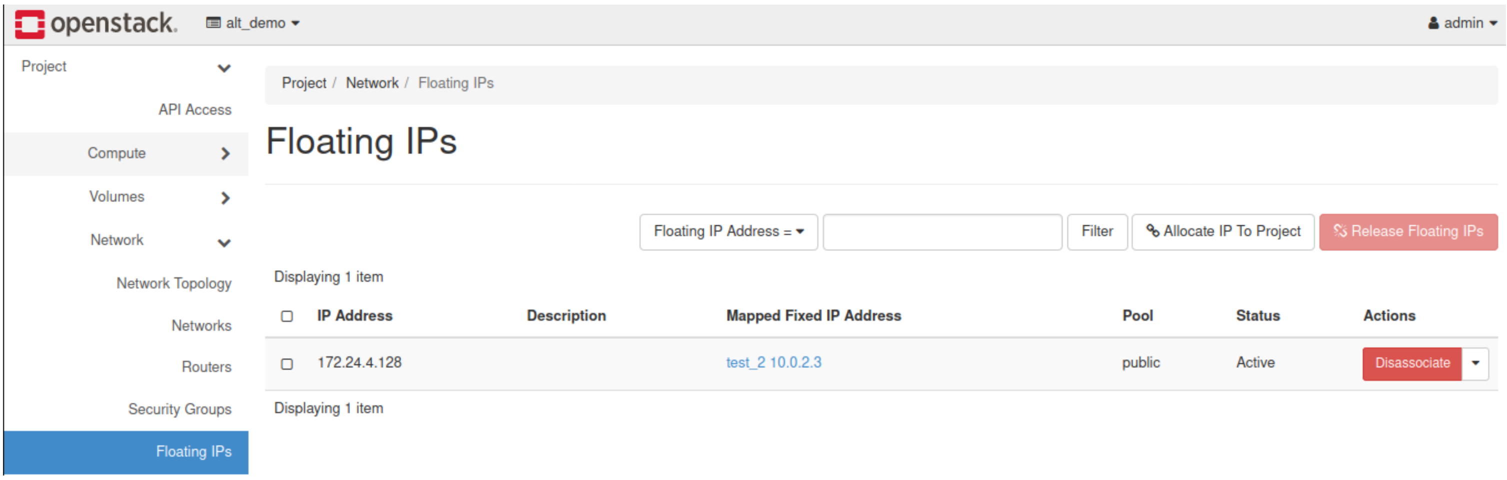
10.0.2.1

Description:

You can connect a specified subnet to the router.

If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

d) Create the Floating IPs that will be assigned to the Compute instances :



Project / Network / Floating IPs

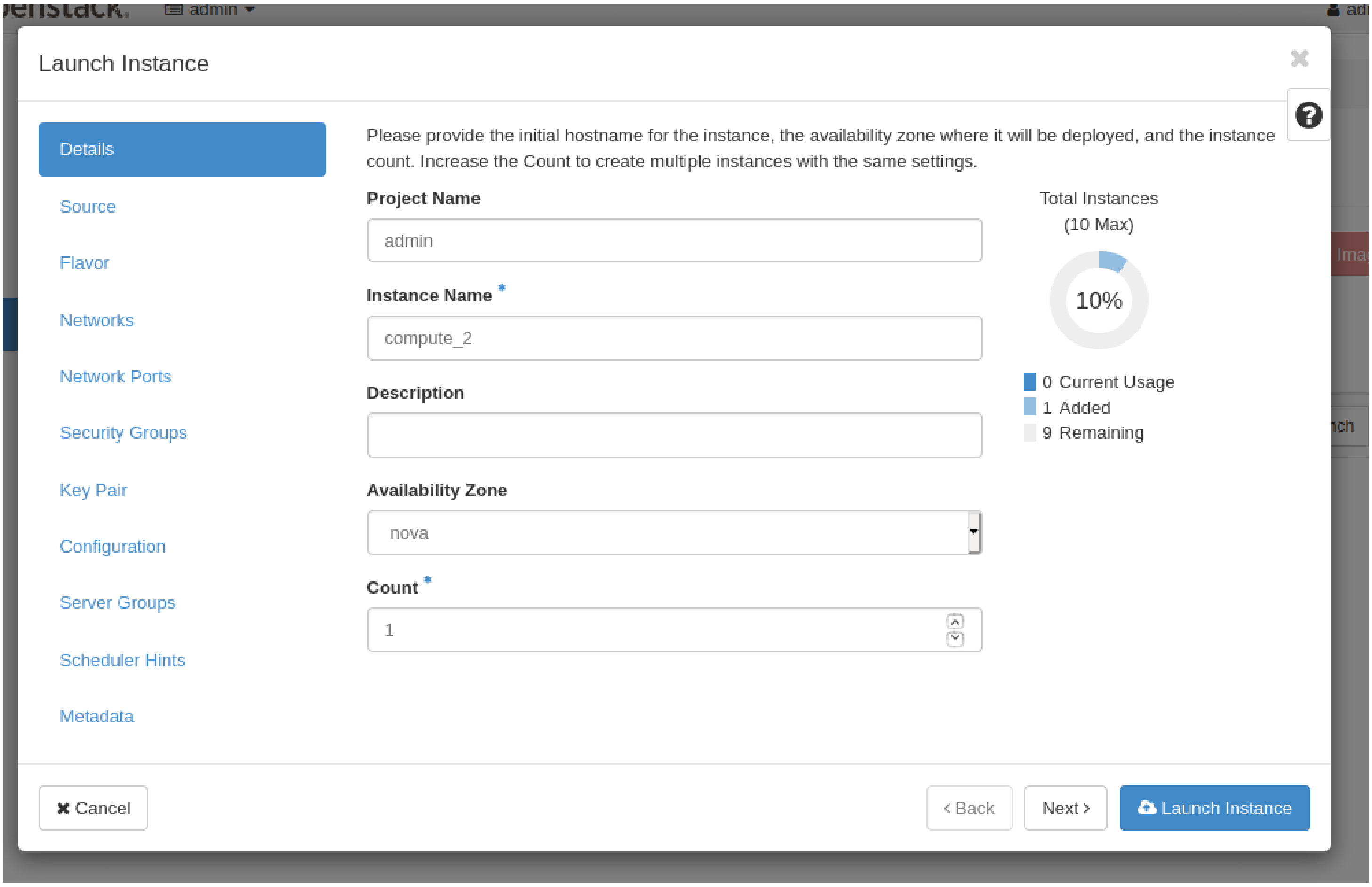
Floating IPs

Displaying 1 item

	IP Address	Description	Mapped Fixed IP Address	Pool	Status	Actions
□	172.24.4.128		test_2 10.0.2.3	public	Active	Disassociate

Displaying 1 item

e) Create the Compute instance



Launch Instance

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Project Name: admin

Instance Name *: compute_2

Description:

Availability Zone: nova

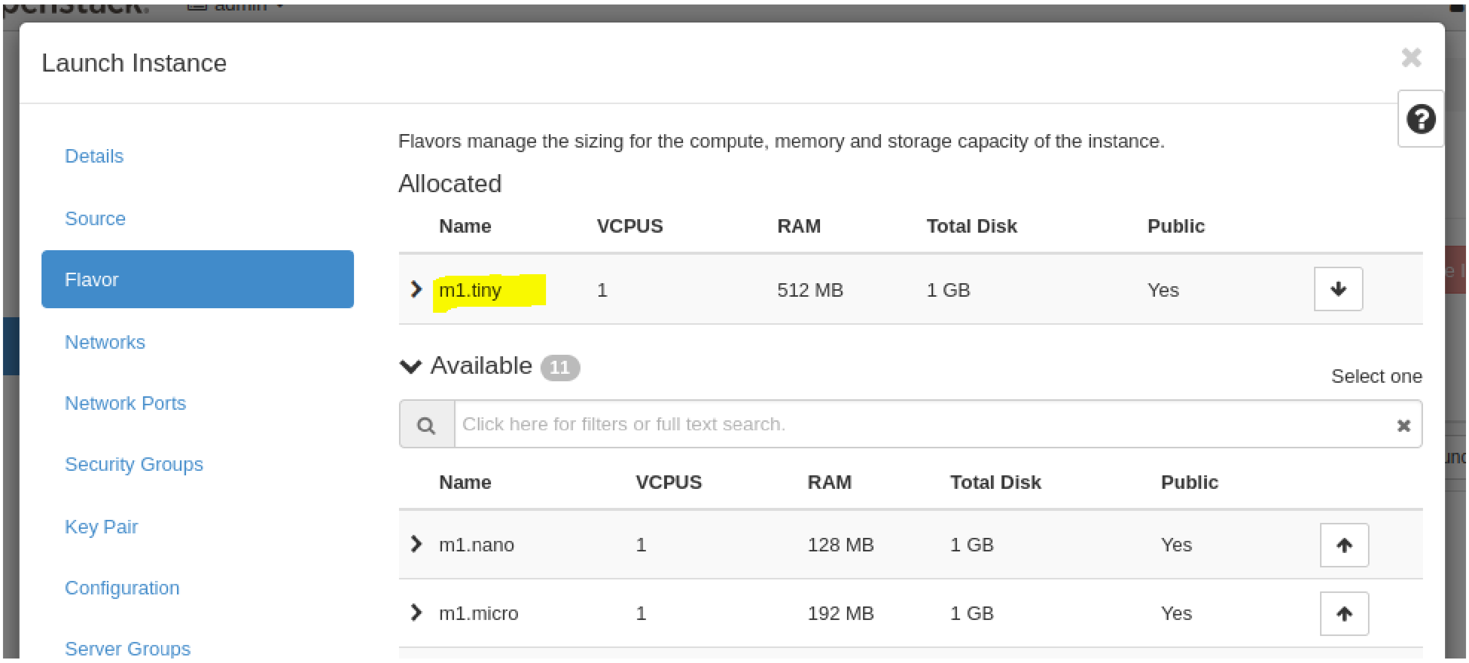
Count *: 1

Total Instances (10 Max)

10%

- 0 Current Usage
- 1 Added
- 9 Remaining

→ In the Flavor Section, select m1.tiny (any type of image can be selected based on the power provided by the host machine) :



Launch Instance

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Public
m1.tiny	1	512 MB	1 GB	Yes

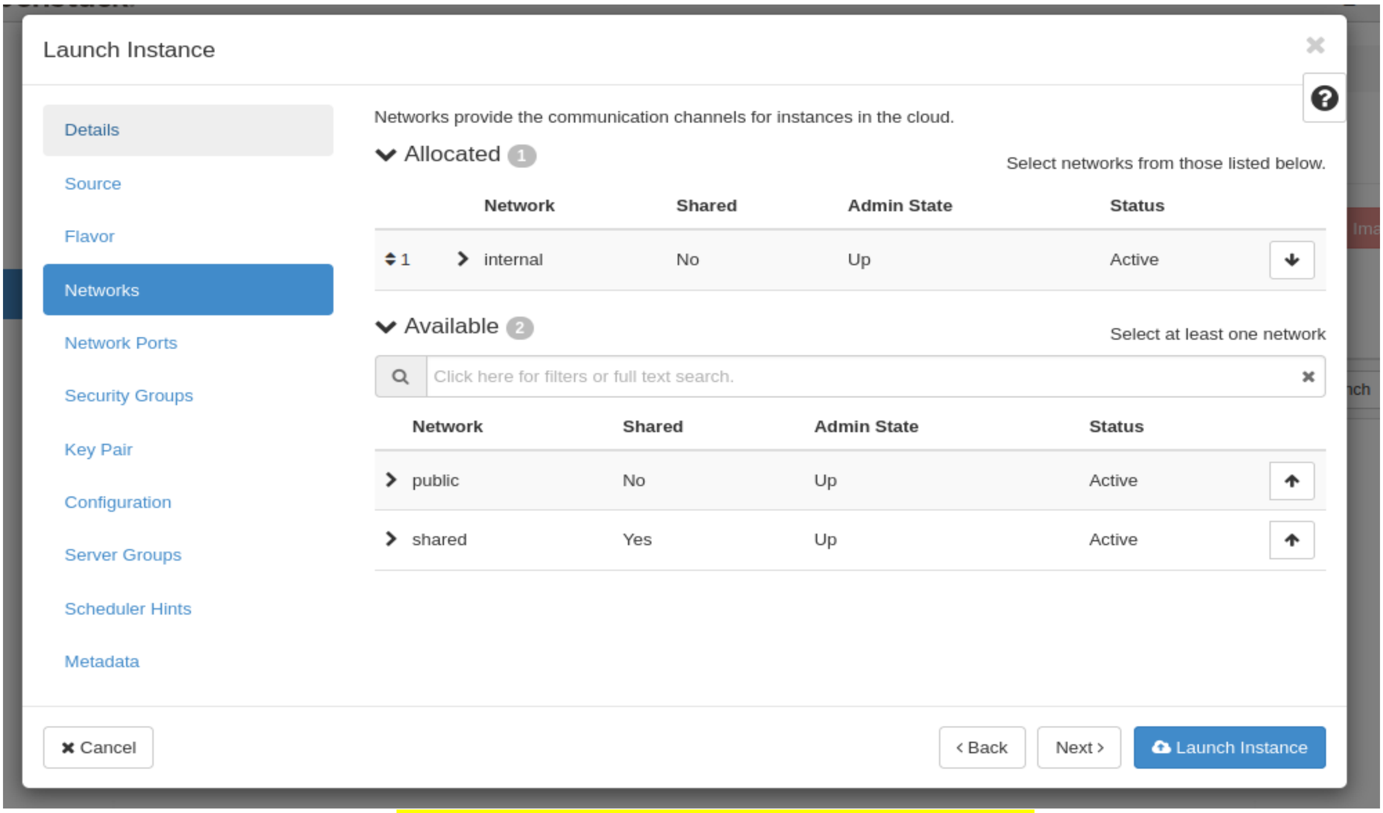
Available 11

Select one

Click here for filters or full text search.

Name	VCPUS	RAM	Total Disk	Public
m1.nano	1	128 MB	1 GB	Yes
m1.micro	1	192 MB	1 GB	Yes

In the Network Section, we select the internal network :



Launch Instance

Networks provide the communication channels for instances in the cloud.

Allocated 1

Select networks from those listed below.

	Network	Shared	Admin State	Status
1	internal	No	Up	Active

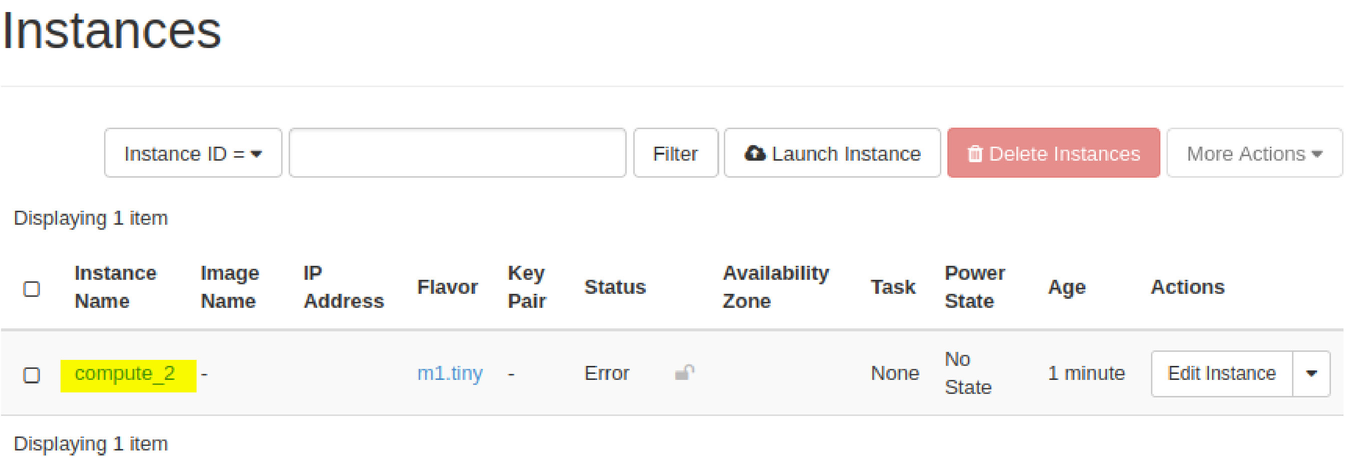
Available 2

Select at least one network

Click here for filters or full text search.

Network	Shared	Admin State	Status
public	No	Up	Active
shared	Yes	Up	Active

➢ Instance created successfully



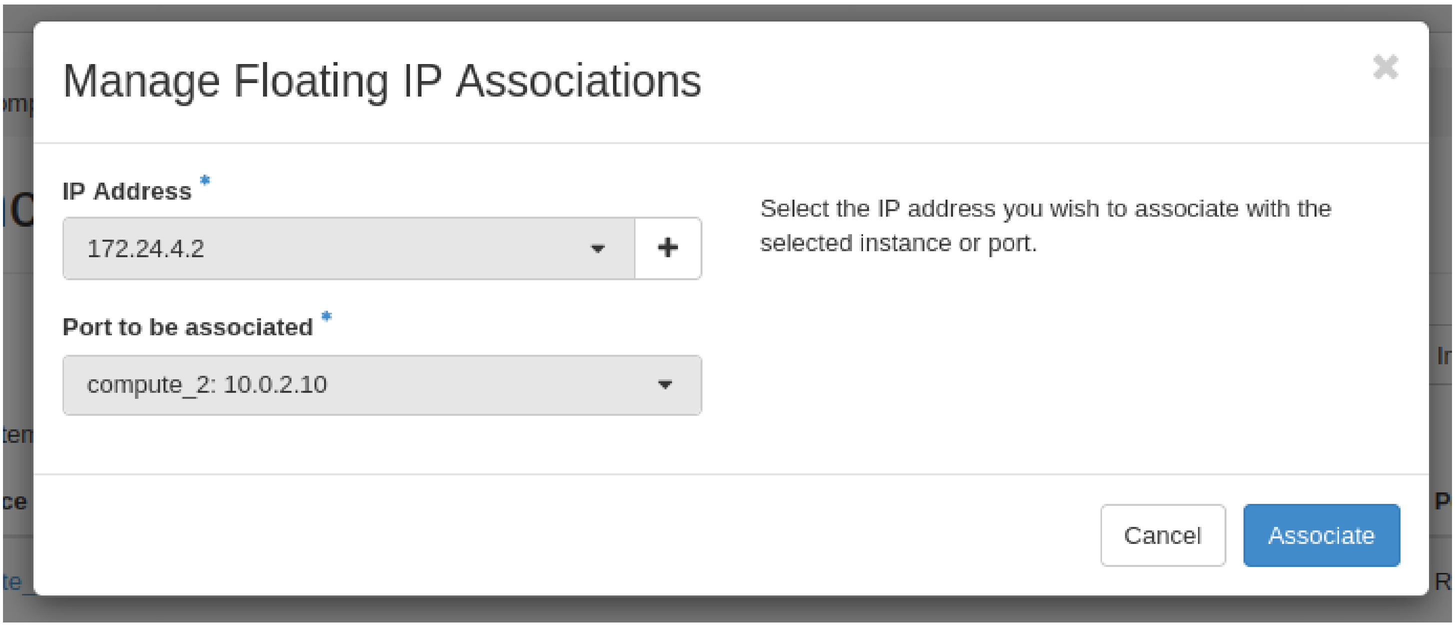
Instances

Displaying 1 item

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
□	compute_2	-		m1.tiny	-	Error		None	No State	1 minute	Edit Instance ▾

Displaying 1 item

f) Assign Floating IP to be able to connect the created instance :



Manage Floating IP Associations

IP Address *

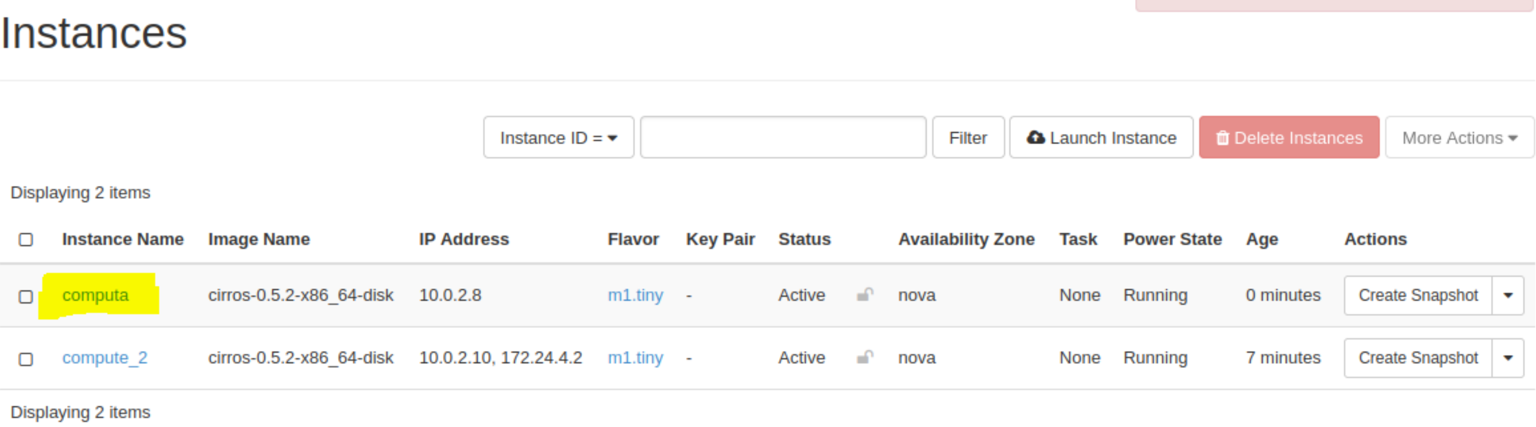
172.24.4.2

Port to be associated *

compute_2: 10.0.2.10

Select the IP address you wish to associate with the selected instance or port.

g) create another Compute Instance with the same steps:



Instances

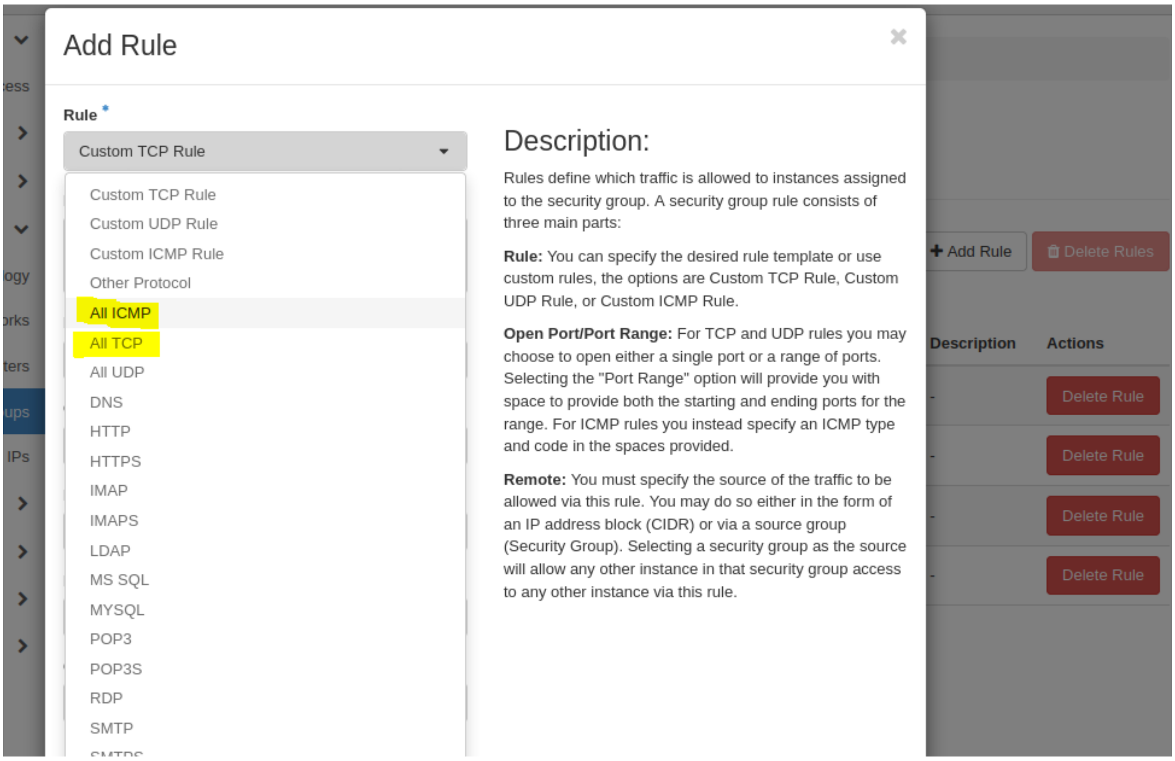
Displaying 2 items

☐	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Age	Actions
☐	computa	cirros-0.5.2-x86_64-disk	10.0.2.8	m1.tiny	-	Active	🔓	nova	None	Running	0 minutes	Create Snapshot
☐	compute_2	cirros-0.5.2-x86_64-disk	10.0.2.10, 172.24.4.2	m1.tiny	-	Active	🔓	nova	None	Running	7 minutes	Create Snapshot

Displaying 2 items

h) Allow Internet Control Message Protocol (ICMP) TCP & SSH .

all icmp amf all Tcp :



Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

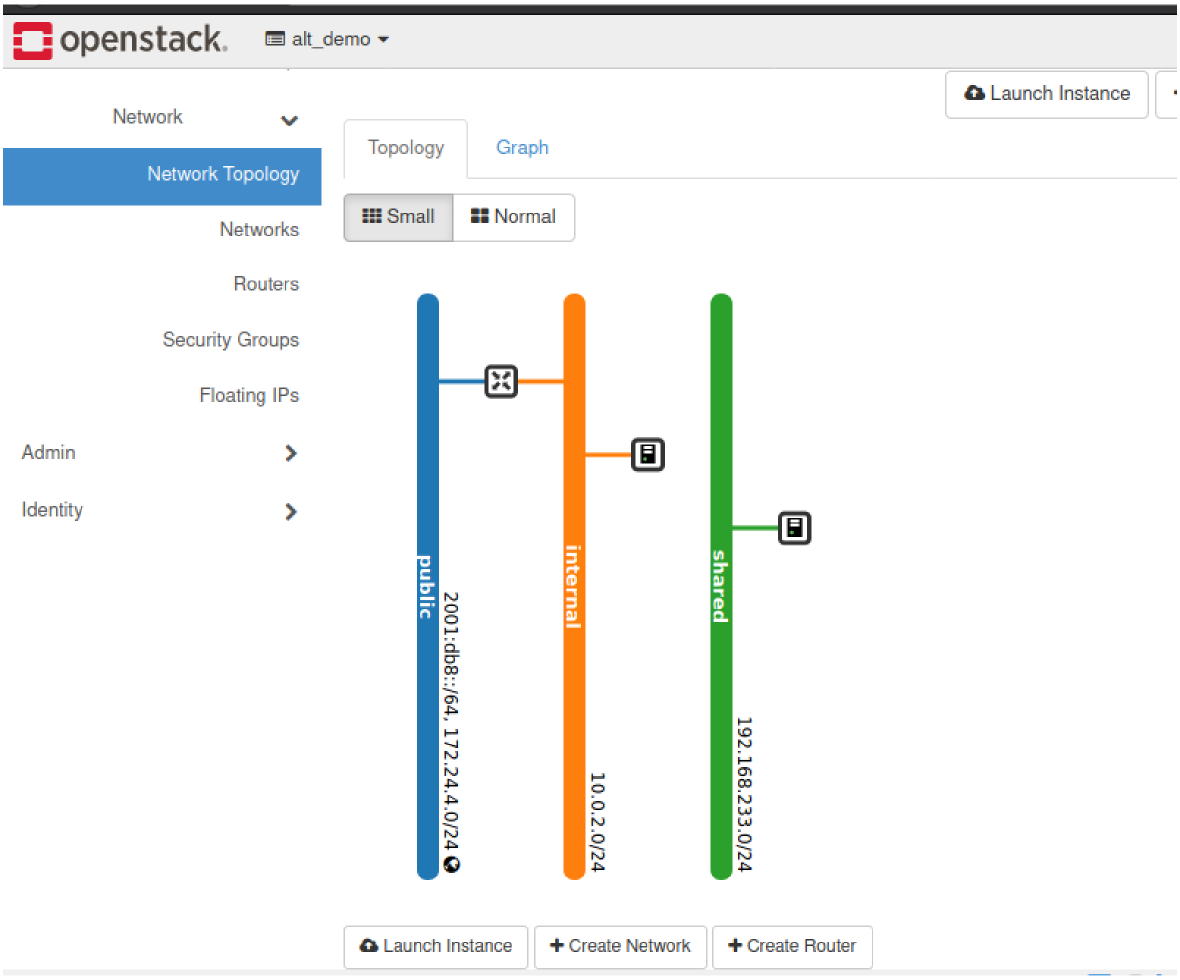
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

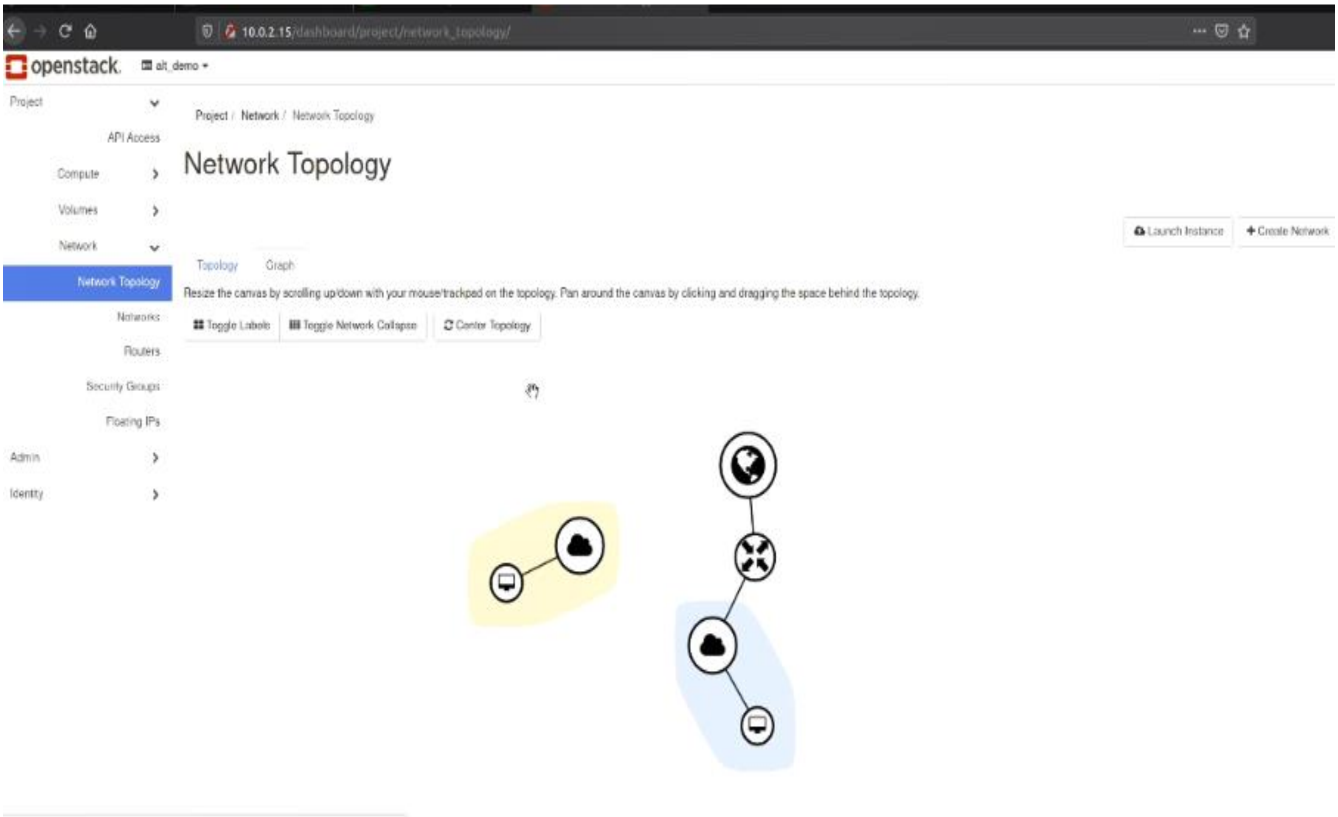
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

i) Iaas network topologie :

→This is the network topology of my iaas project :



→Graph topology :

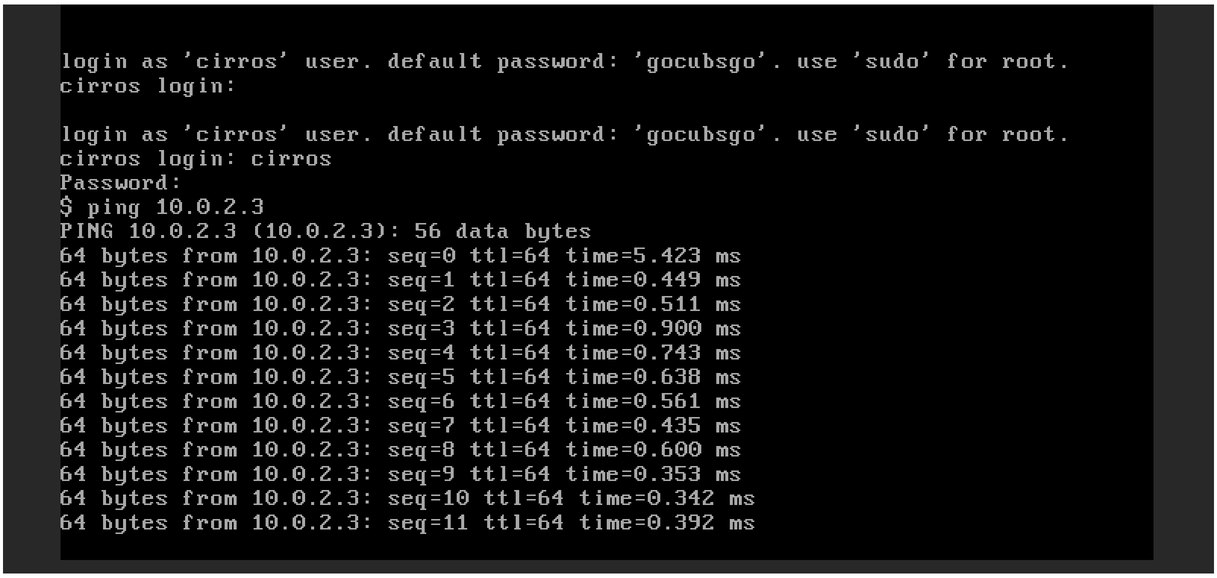


Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

j) Testing the IaaS environment

1.) Ping the Compute instances from the internal network :

Console ping (cirros default linux of openstack)



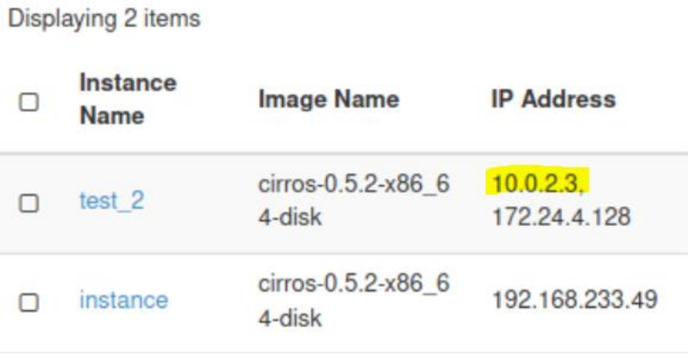
```
login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root.
cirros login:

login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root.
cirros login: cirros
Password:
$ ping 10.0.2.3
PING 10.0.2.3 (10.0.2.3): 56 data bytes
64 bytes from 10.0.2.3: seq=0 ttl=64 time=5.423 ms
64 bytes from 10.0.2.3: seq=1 ttl=64 time=0.449 ms
64 bytes from 10.0.2.3: seq=2 ttl=64 time=0.511 ms
64 bytes from 10.0.2.3: seq=3 ttl=64 time=0.900 ms
64 bytes from 10.0.2.3: seq=4 ttl=64 time=0.743 ms
64 bytes from 10.0.2.3: seq=5 ttl=64 time=0.638 ms
64 bytes from 10.0.2.3: seq=6 ttl=64 time=0.561 ms
64 bytes from 10.0.2.3: seq=7 ttl=64 time=0.435 ms
64 bytes from 10.0.2.3: seq=8 ttl=64 time=0.600 ms
64 bytes from 10.0.2.3: seq=9 ttl=64 time=0.353 ms
64 bytes from 10.0.2.3: seq=10 ttl=64 time=0.342 ms
64 bytes from 10.0.2.3: seq=11 ttl=64 time=0.392 ms
```

2.) **Ping the Compute instances from the external network :**

From ISP network in this case : This will be done from the Ubuntu Machine.

- **First ping the ip add of instance :**

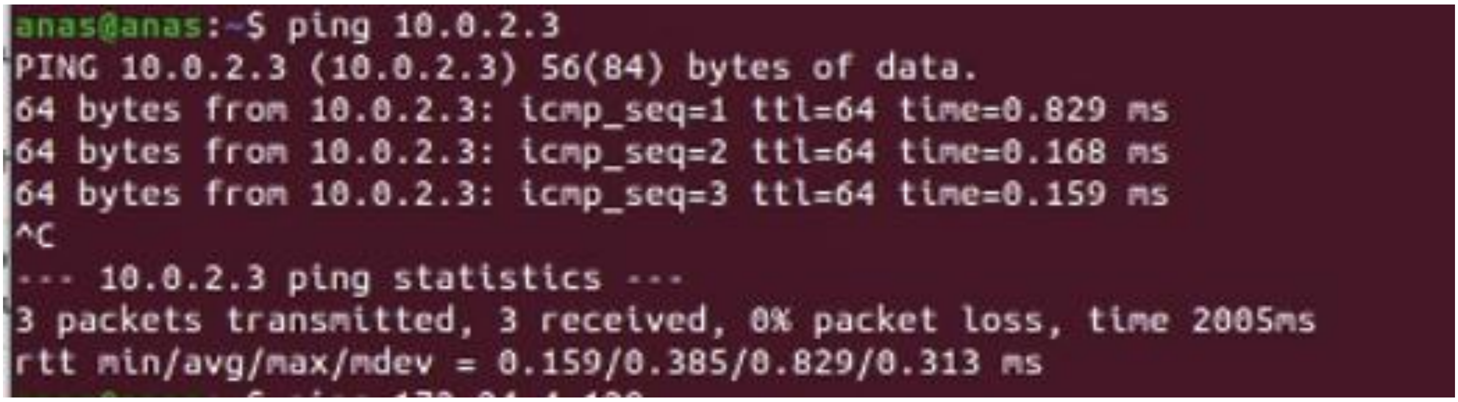
ip add = 10.0.2.3



Displaying 2 items

□	Instance Name	Image Name	IP Address
□	test_2	cirros-0.5.2-x86_64-disk	10.0.2.3, 172.24.4.128
□	instance	cirros-0.5.2-x86_64-disk	192.168.233.49

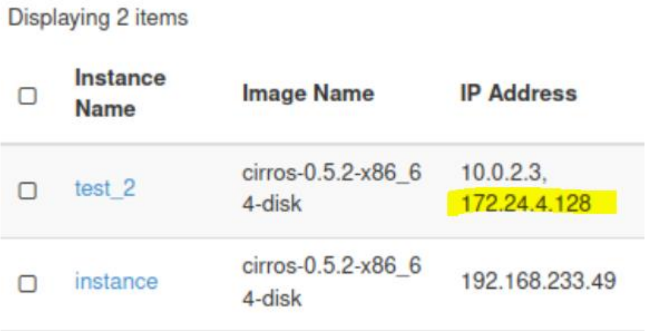
result : ping 10.0.2.3 :



```
anas@anas:~$ ping 10.0.2.3
PING 10.0.2.3 (10.0.2.3) 56(84) bytes of data.
64 bytes from 10.0.2.3: icmp_seq=1 ttl=64 time=0.829 ms
64 bytes from 10.0.2.3: icmp_seq=2 ttl=64 time=0.168 ms
64 bytes from 10.0.2.3: icmp_seq=3 ttl=64 time=0.159 ms
^C
--- 10.0.2.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.159/0.385/0.829/0.313 ms
```

- **Second, ping the Floating ip add of instance :**

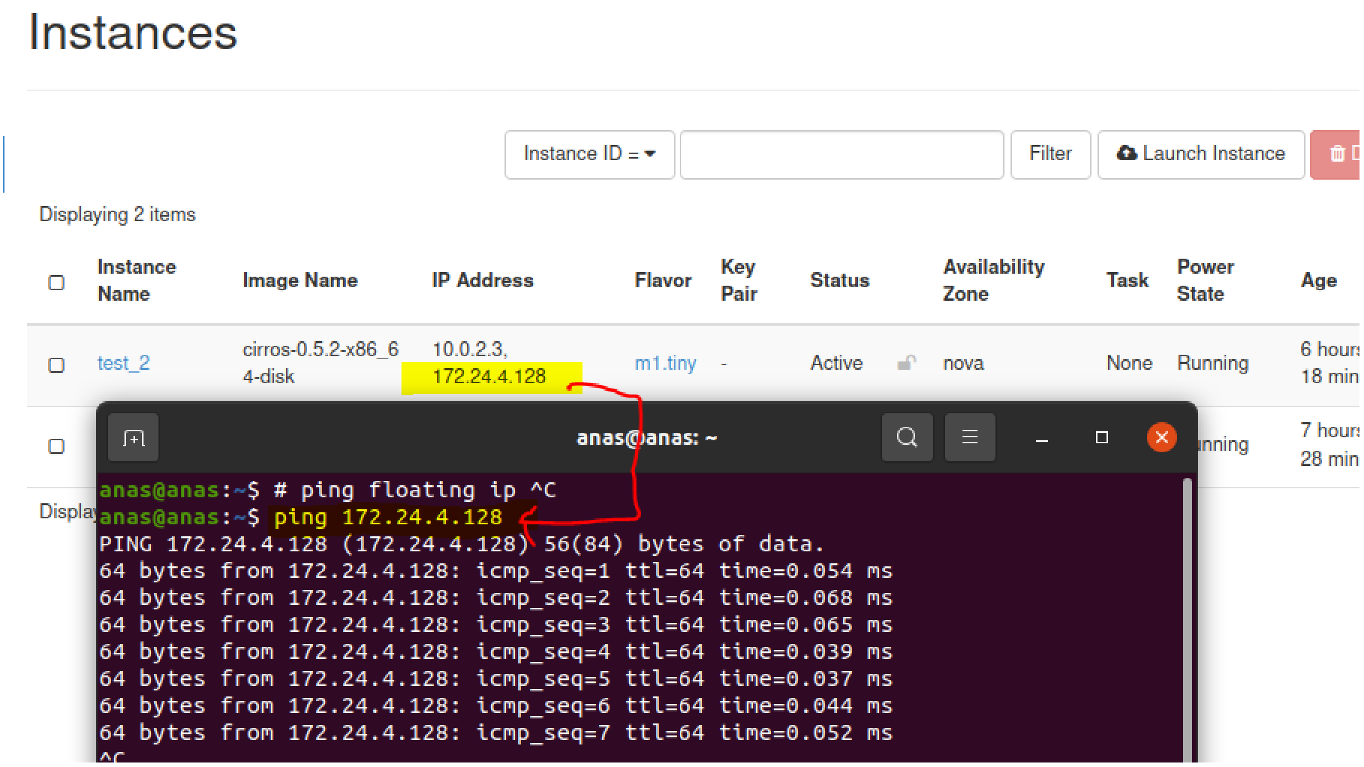
Floating ip =172.24.1.128 :



Displaying 2 items

□	Instance Name	Image Name	IP Address
□	test_2	cirros-0.5.2-x86_64-disk	10.0.2.3, 172.24.4.128
□	instance	cirros-0.5.2-x86_64-disk	192.168.233.49

Resualt of : ping 172.24.4.128



Instances

Displaying 2 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age
□	test_2	cirros-0.5.2-x86_64-disk	10.0.2.3, 172.24.4.128	m1.tiny	-	Active	nova	None	Running	6 hours 18 min
□									nning	7 hours 28 min

Display

anas@anas: ~

```
anas@anas:~$ # ping floating ip ^C
anas@anas:~$ ping 172.24.4.128
PING 172.24.4.128 (172.24.4.128) 56(84) bytes of data.
64 bytes from 172.24.4.128: icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from 172.24.4.128: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 172.24.4.128: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 172.24.4.128: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 172.24.4.128: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 172.24.4.128: icmp_seq=6 ttl=64 time=0.044 ms
64 bytes from 172.24.4.128: icmp_seq=7 ttl=64 time=0.052 ms
^C
```

All the tests have been successful. The Infrastructure as a Service environment can be used without hesitations for any type of deployment and application development.

6) Details and tests on the SaaS created

What is saas ?

SaaS is a method of software delivery that allows data to be accessed from any device with an internet connection and a web browser. In this web-based model, software vendors host and maintain the servers, databases, and the code that makes up an application.

OpenStack is, first and foremost, built to provide an Infrastructure-as-a-Service, however, its versatility has lent it to breaking out of that box and being used to build and support both PaaS and SaaS clouds.

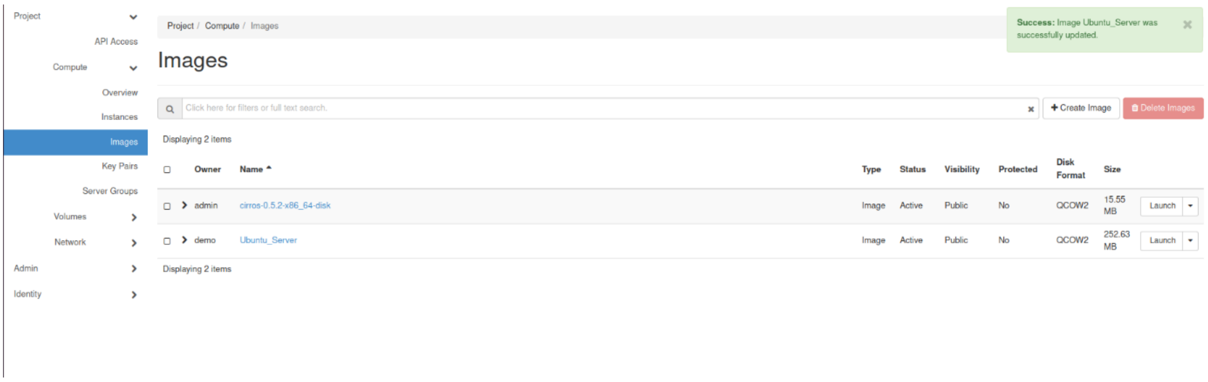
My implementation of saas :

Now i will install Ubuntu server as host and apache as software as service, if we can access and use apache then we can do the same with other applications.

So for that in this workshop we will imagine that apache is software.

➢ The creation of an Ubuntu Server :

- instance First of all we created an Ubuntu Server image since it does not exist on OpenStack



Project / Compute / Images

Success: Image Ubuntu_Server was successfully updated.

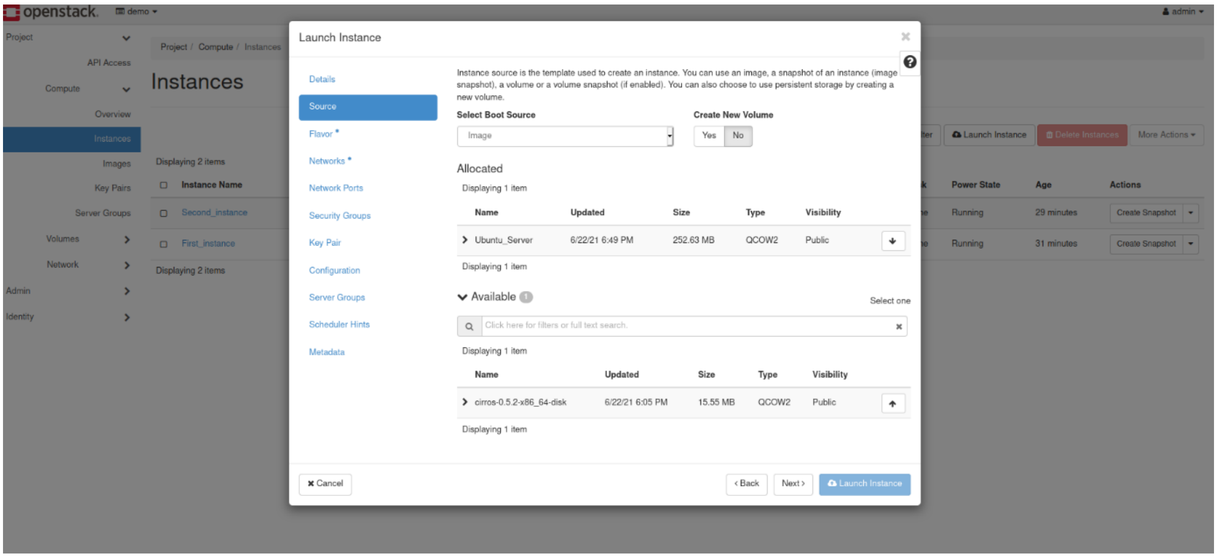
Images

Displaying 2 items

□	Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size	
□	admin	cirros-0.5.2-x86_64-disk	Image	Active	Public	No	QCOW2	15.55 MB	Launch
□	demo	Ubuntu_Server	Image	Active	Public	No	QCOW2	252.63 MB	Launch

Displaying 2 items

Then we create an Ubuntu Server instance with a pair key to be able to access the server using ssh :



Launch Instance

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: Yes [✓ No]

Allocated

Displaying 1 item

Name	Updated	Size	Type	Visibility
Ubuntu_Server	6/22/21 6:49 PM	252.63 MB	QCOW2	Public

Displaying 1 item

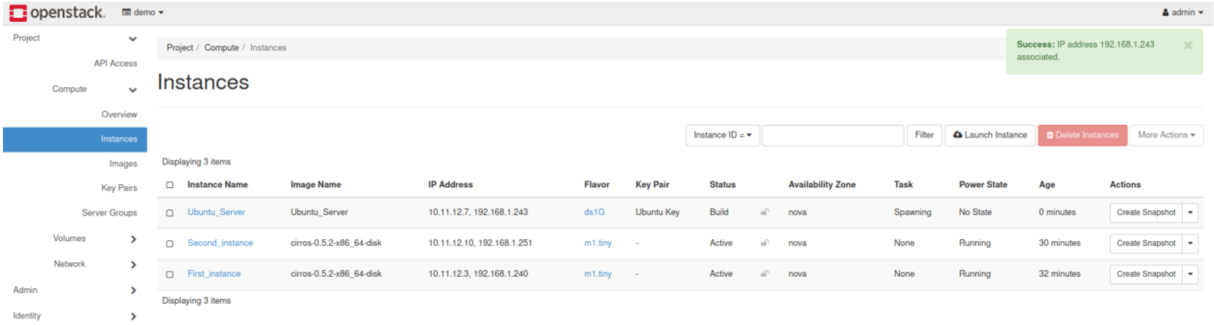
Available 1

Select one

Displaying 1 item

Name	Updated	Size	Type	Visibility
cirros-0.5.2-x86_64-disk	6/22/21 6:05 PM	15.55 MB	QCOW2	Public

Displaying 1 item



Project / Compute / Instances

Success: IP address 192.168.1.243 associated.

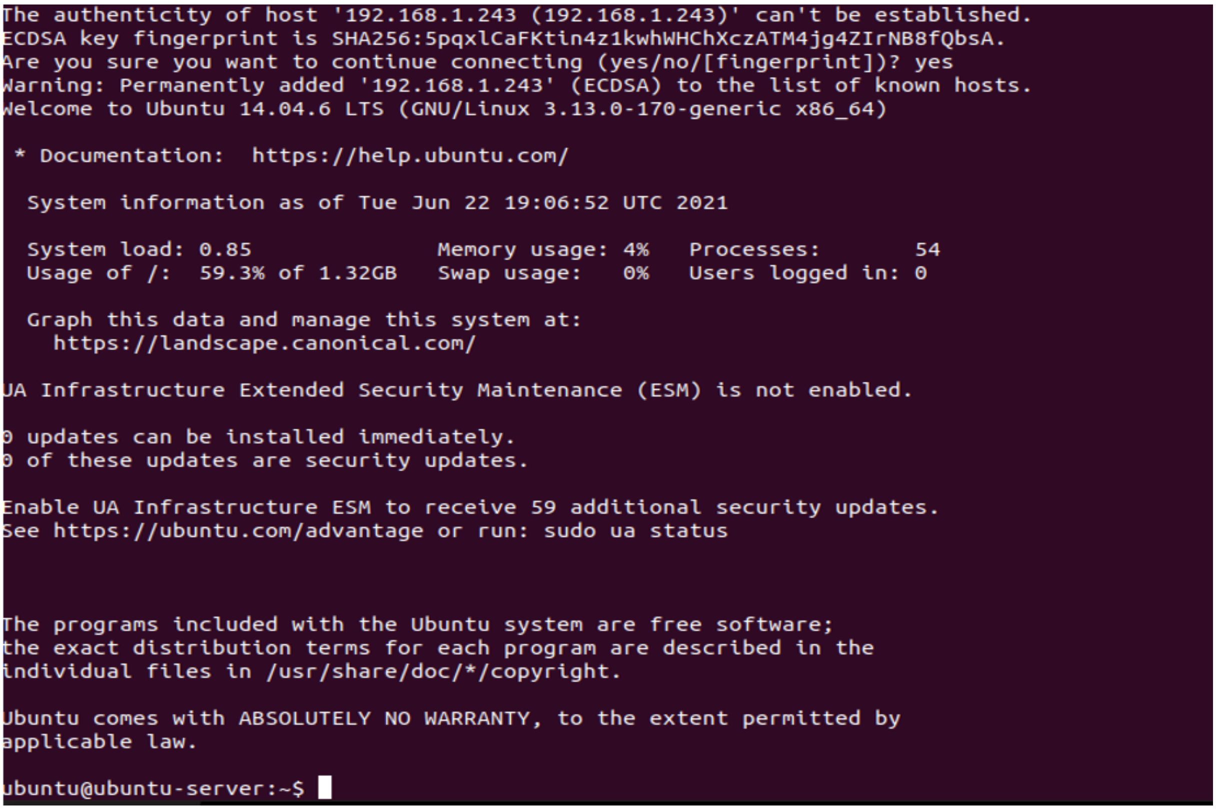
Instances

Displaying 3 items

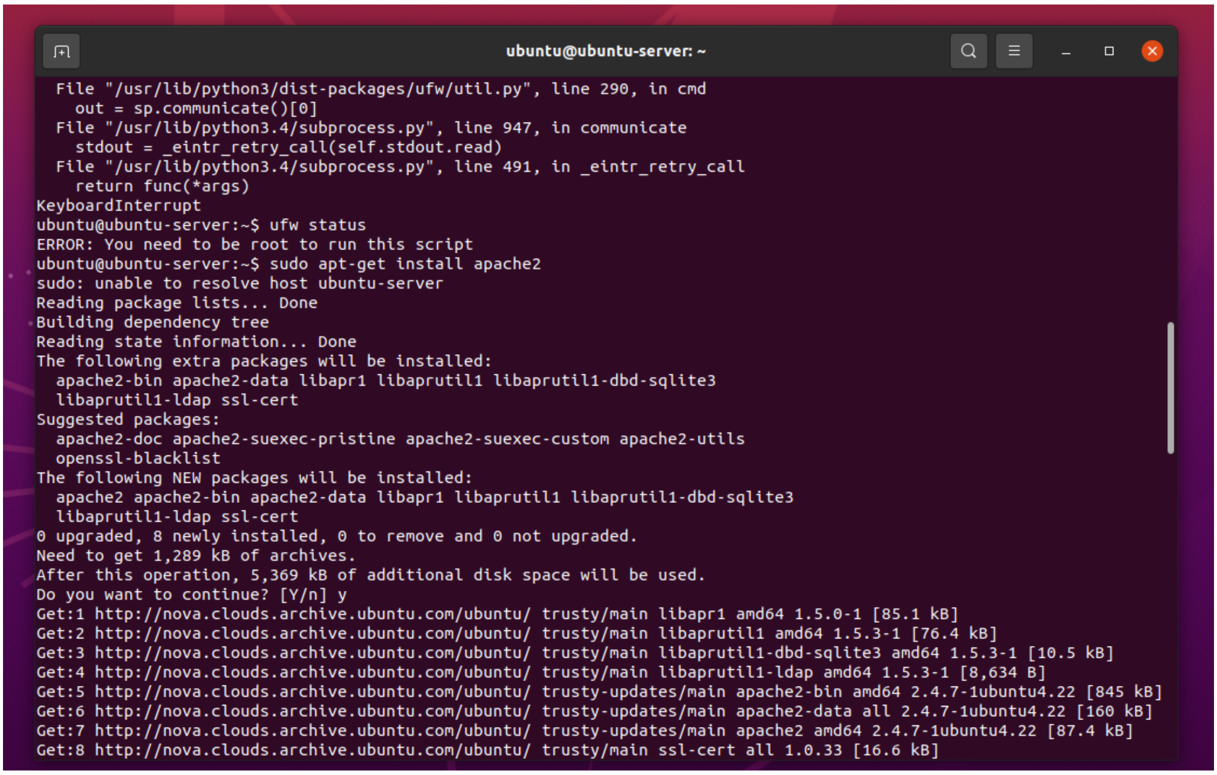
□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
□	Ubuntu_Server	Ubuntu_Server	10.11.12.7, 192.168.1.243	ds1G	Ubuntu Key	Build	nova	Spawning	No State	0 minutes	Create Snapshot
□	Second_instance	cirros-0.5.2-x86_64-disk	10.11.12.10, 192.168.1.251	m1.tiny	-	Active	nova	None	Running	30 minutes	Create Snapshot
□	First_instance	cirros-0.5.2-x86_64-disk	10.11.12.3, 192.168.1.240	m1.tiny	-	Active	nova	None	Running	32 minutes	Create Snapshot

Displaying 3 items

➢ Execution of : Ubuntu Server et apache



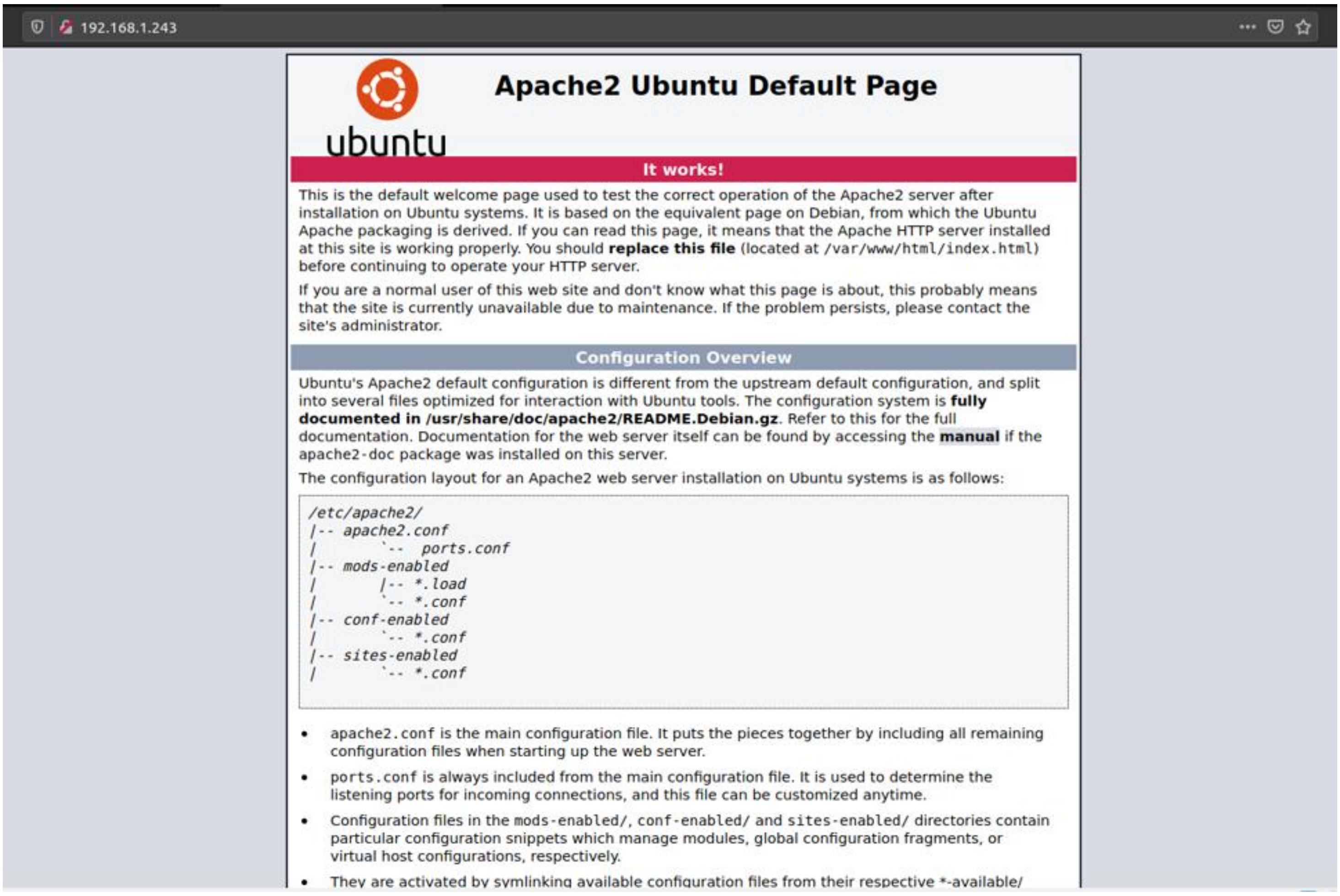
```
The authenticity of host '192.168.1.243 (192.168.1.243)' can't be established.
ECDSA key fingerprint is SHA256:5pqxlCaFKtin4z1kwhWHChXczATM4jg4ZIrNB8fQbsA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.243' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Jun 22 19:06:52 UTC 2021

  System load: 0.85               Memory usage: 4%   Processes:       54
  Usage of /:  59.3% of 1.32GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

UA Infrastructure Extended Security Maintenance (ESM) is not enabled.

0 updates can be installed immediately.
0 of these updates are security updates.

Enable UA Infrastructure ESM to receive 59 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu-server:~$ 
```



```
  File "/usr/lib/python3/dist-packages/ufw/util.py", line 290, in cmd
    out = sp.communicate()[0]
  File "/usr/lib/python3.4/subprocess.py", line 947, in communicate
    stdout = _eintr_retry_call(self.stdout.read)
  File "/usr/lib/python3.4/subprocess.py", line 491, in _eintr_retry_call
    return func(*args)
KeyboardInterrupt
ubuntu@ubuntu-server:~$ ufw status
ERROR: You need to be root to run this script
ubuntu@ubuntu-server:~$ sudo apt-get install apache2
sudo: unable to resolve host ubuntu-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap ssl-cert
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,289 kB of archives.
After this operation, 5,369 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty/main libapr1 amd64 1.5.0-1 [85.1 kB]
Get:2 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1 amd64 1.5.3-1 [76.4 kB]
Get:3 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1-dbd-sqlite3 amd64 1.5.3-1 [10.5 kB]
Get:4 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1-ldap amd64 1.5.3-1 [8,634 B]
Get:5 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty-updates/main apache2-bin amd64 2.4.7-1ubuntu4.22 [845 kB]
Get:6 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty-updates/main apache2-data all 2.4.7-1ubuntu4.22 [160 kB]
Get:7 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty-updates/main apache2 amd64 2.4.7-1ubuntu4.22 [87.4 kB]
Get:8 http://nova.clouds.archive.ubuntu.com/ubuntu/ trusty/main ssl-cert all 1.0.33 [16.6 kB]
```

➢ TEST:

Here after all the installations are finished we test the access to the apache interface here are the results:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

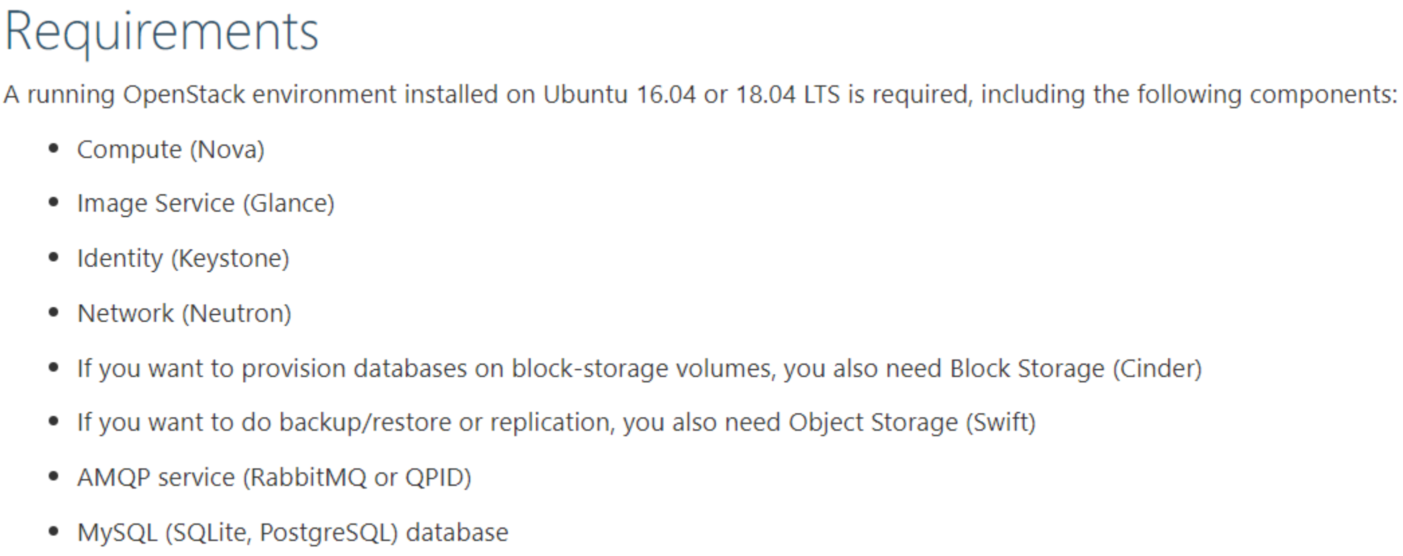
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/

7) DaaS: Trove project (Database as a service) : steps

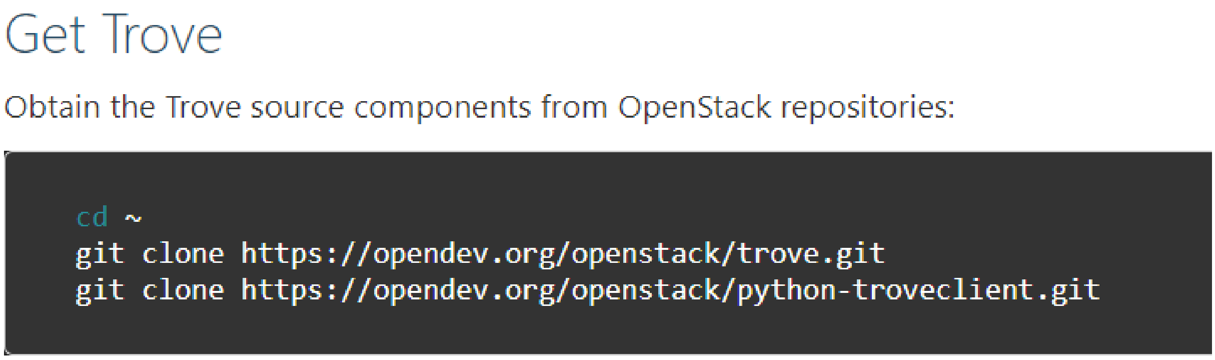


Requirements

A running OpenStack environment installed on Ubuntu 16.04 or 18.04 LTS is required, including the following components:

- Compute (Nova)
- Image Service (Glance)
- Identity (Keystone)
- Network (Neutron)
- If you want to provision databases on block-storage volumes, you also need Block Storage (Cinder)
- If you want to do backup/restore or replication, you also need Object Storage (Swift)
- AMQP service (RabbitMQ or QPID)
- MySQL (SQLite, PostgreSQL) database

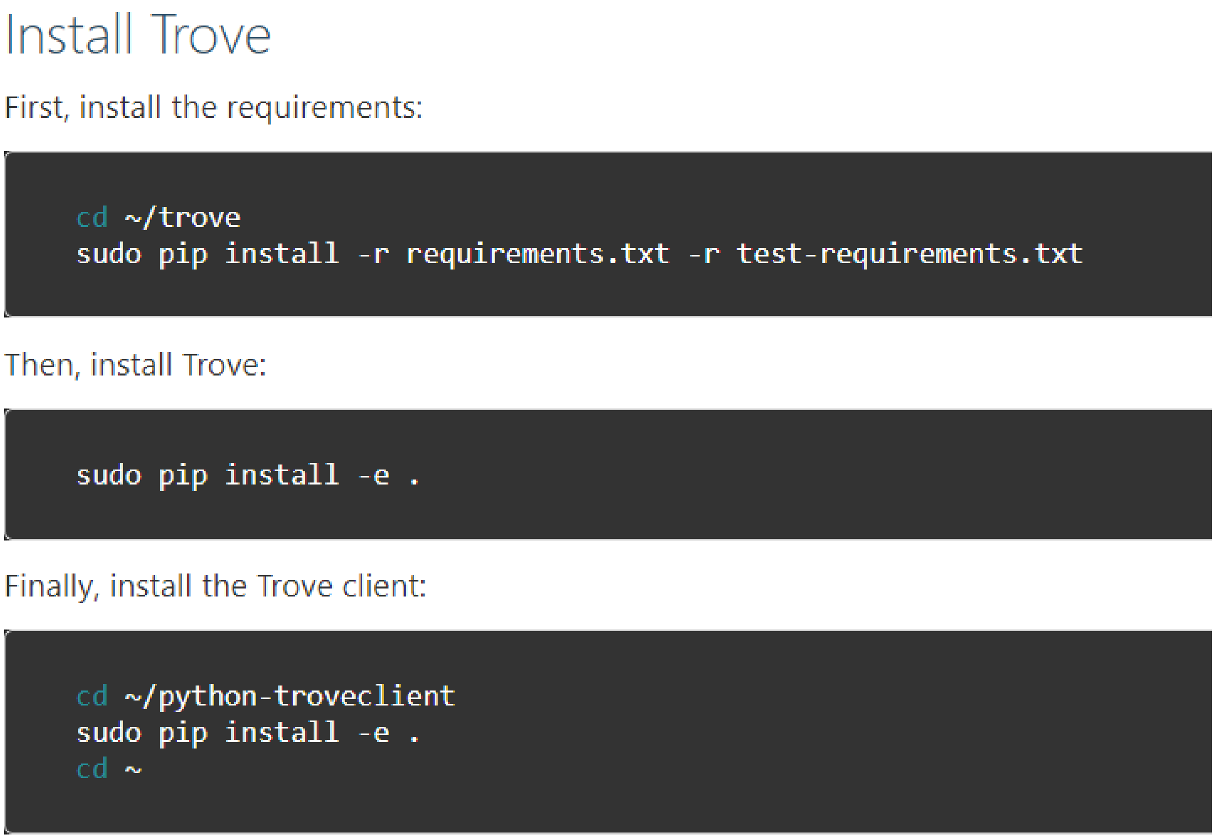
➢ **Commands used to install Trove (Database as a service) :**



Get Trove

Obtain the Trove source components from OpenStack repositories:

```
cd ~
git clone https://opendev.org/openstack/trove.git
git clone https://opendev.org/openstack/python-troveclient.git
```



Install Trove

First, install the requirements:

```
cd ~/trove
sudo pip install -r requirements.txt -r test-requirements.txt
```

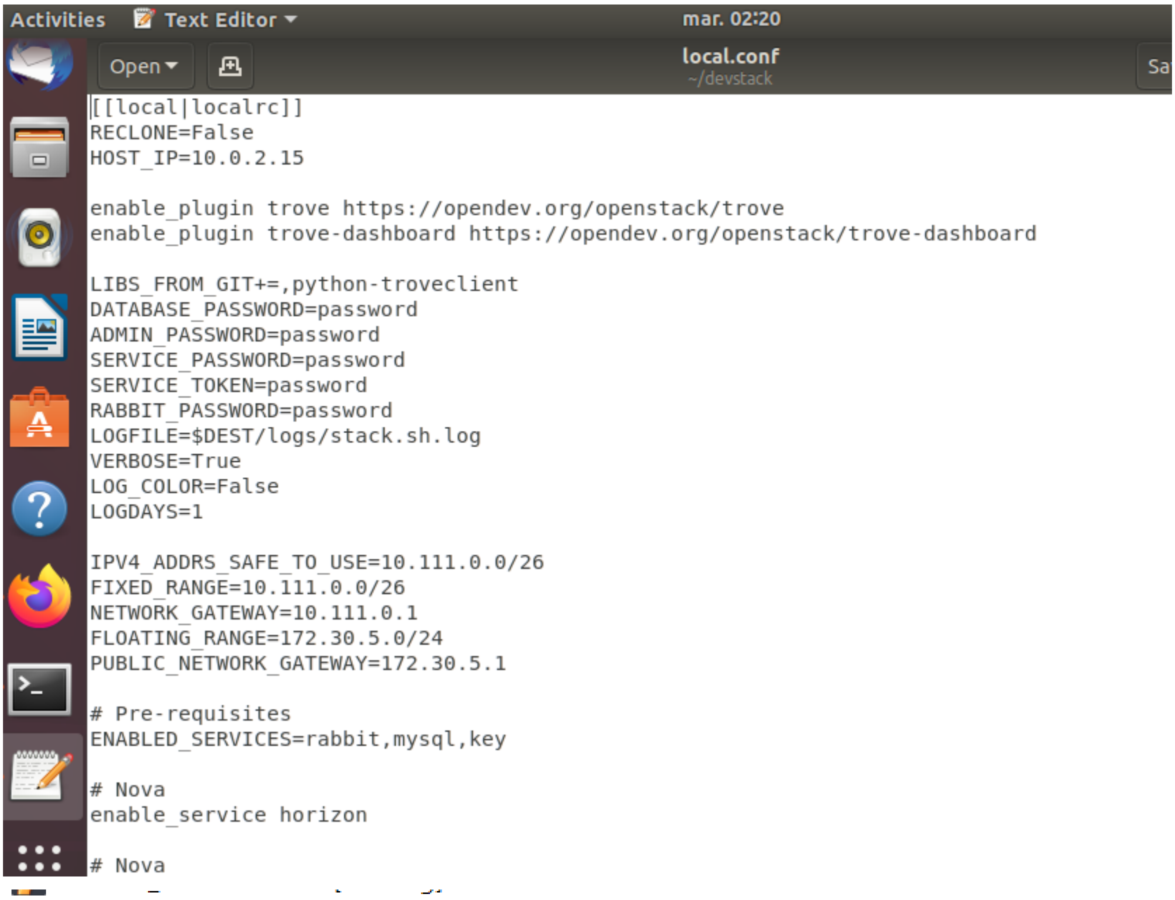
Then, install Trove:

```
sudo pip install -e .
```

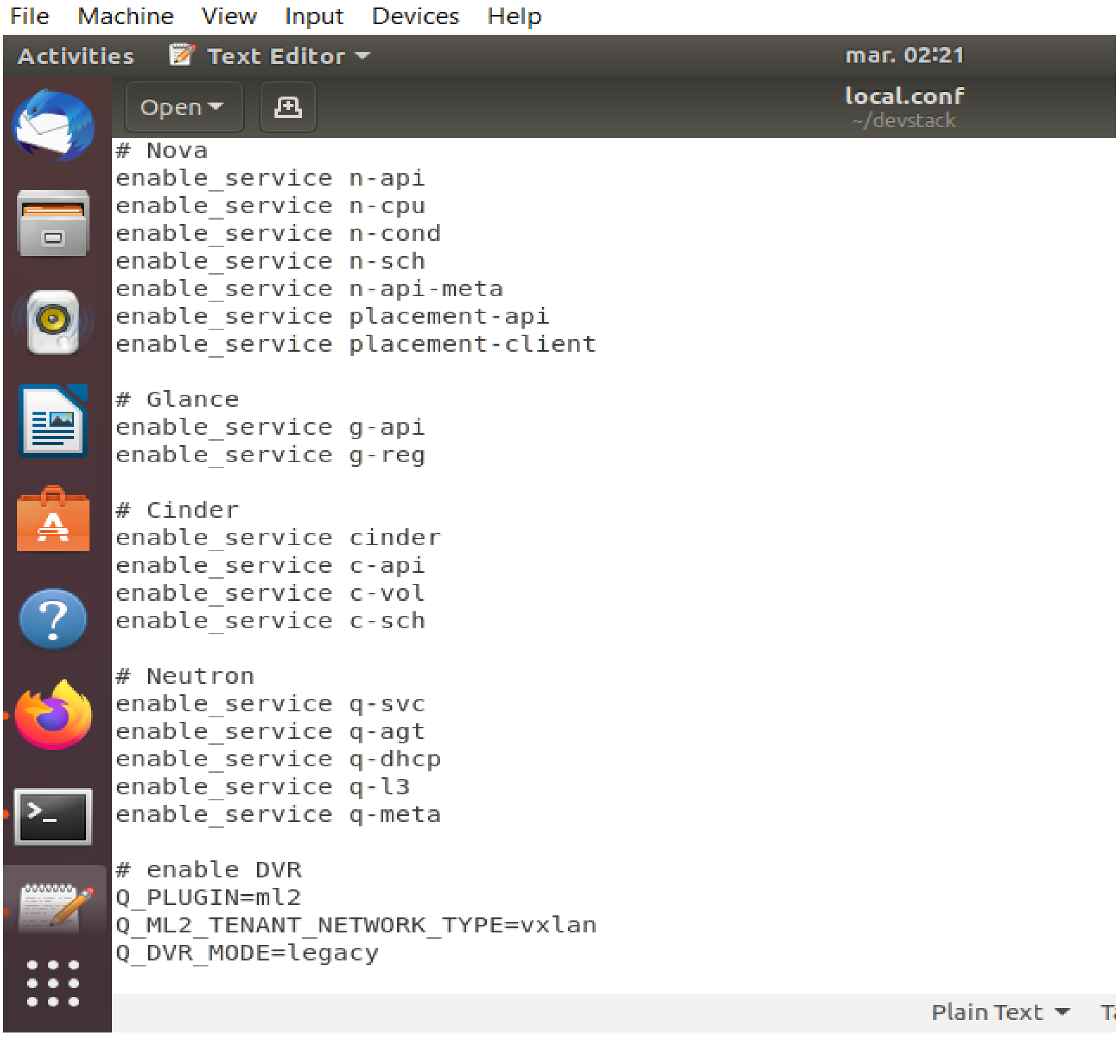
Finally, install the Trove client:

```
cd ~/python-troveclient
sudo pip install -e .
cd ~
```

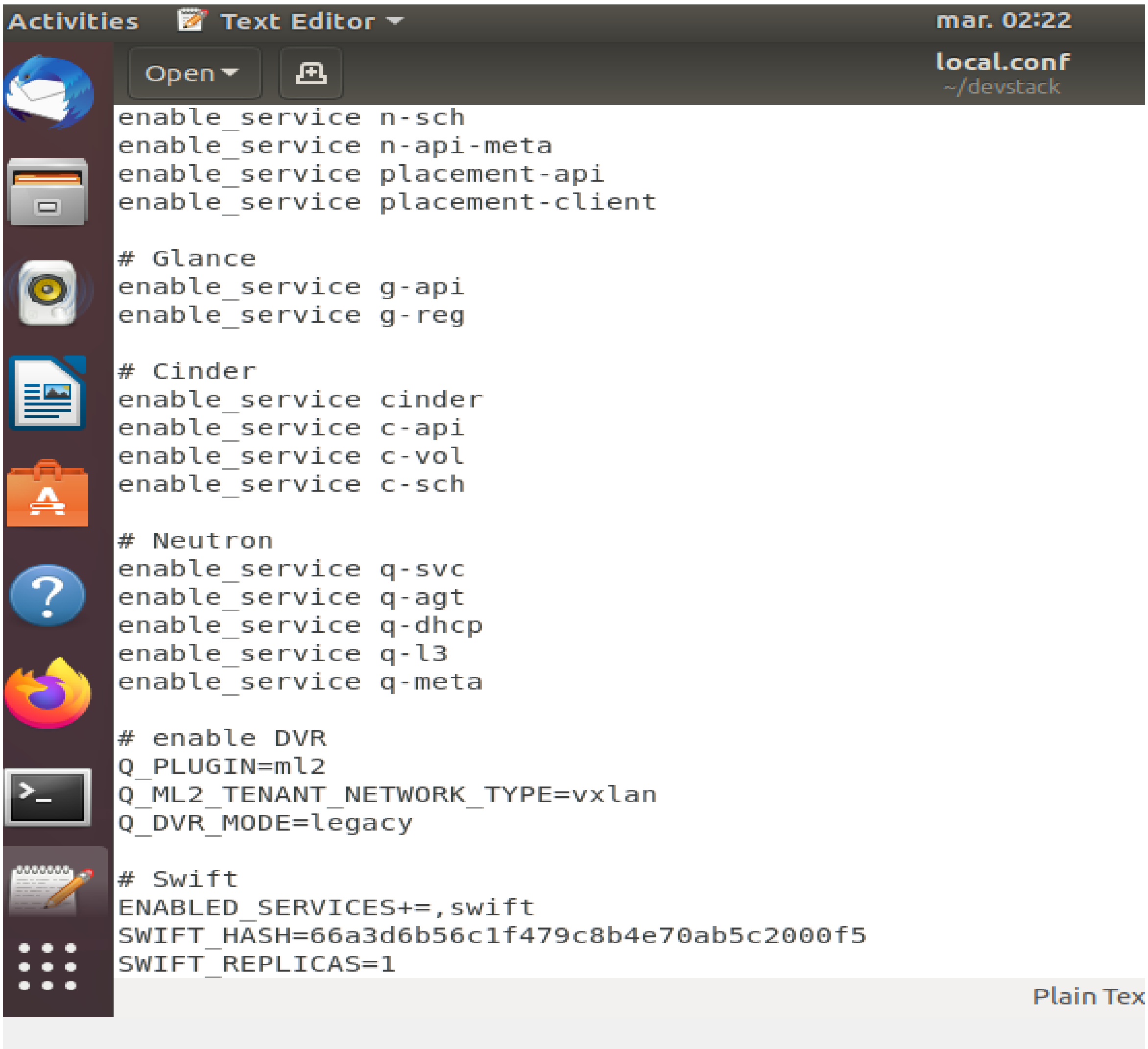
- **My Local.conf of Trove installation :**



```
[[local|localrc]]
RECLONE=False
HOST_IP=10.0.2.15

enable_plugin trove https://opendev.org/openstack/trove
enable_plugin trove-dashboard https://opendev.org/openstack/trove-dashboard

LIBS_FROM_GIT+=,python-troveclient
DATABASE_PASSWORD=password
ADMIN_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
RABBIT_PASSWORD=password
LOGFILE=$DEST/logs/stack.sh.log
VERBOSE=True
LOG_COLOR=False
LOGDAYS=1

IPV4_ADDRS_SAFE_TO_USE=10.111.0.0/26
FIXED_RANGE=10.111.0.0/26
NETWORK_GATEWAY=10.111.0.1
FLOATING_RANGE=172.30.5.0/24
PUBLIC_NETWORK_GATEWAY=172.30.5.1

# Pre-requisites
ENABLED_SERVICES=rabbit,mysql,key

# Nova
enable_service horizon

# Nova
```

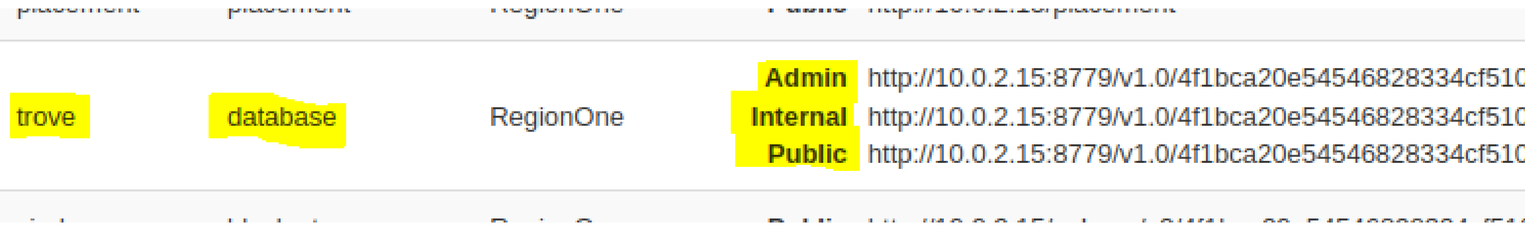


```
# Nova
enable_service n-api
enable_service n-cpu
enable_service n-cond
enable_service n-sch
enable_service n-api-meta
enable_service placement-api
enable_service placement-client

# Glance
enable_service g-api
enable_service g-reg

# Cinder
enable_service cinder
enable_service c-api
enable_service c-vol
enable_service c-sch

# Neutron
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta

# enable DVR
Q_PLUGIN=ml2
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_DVR_MODE=legacy
```



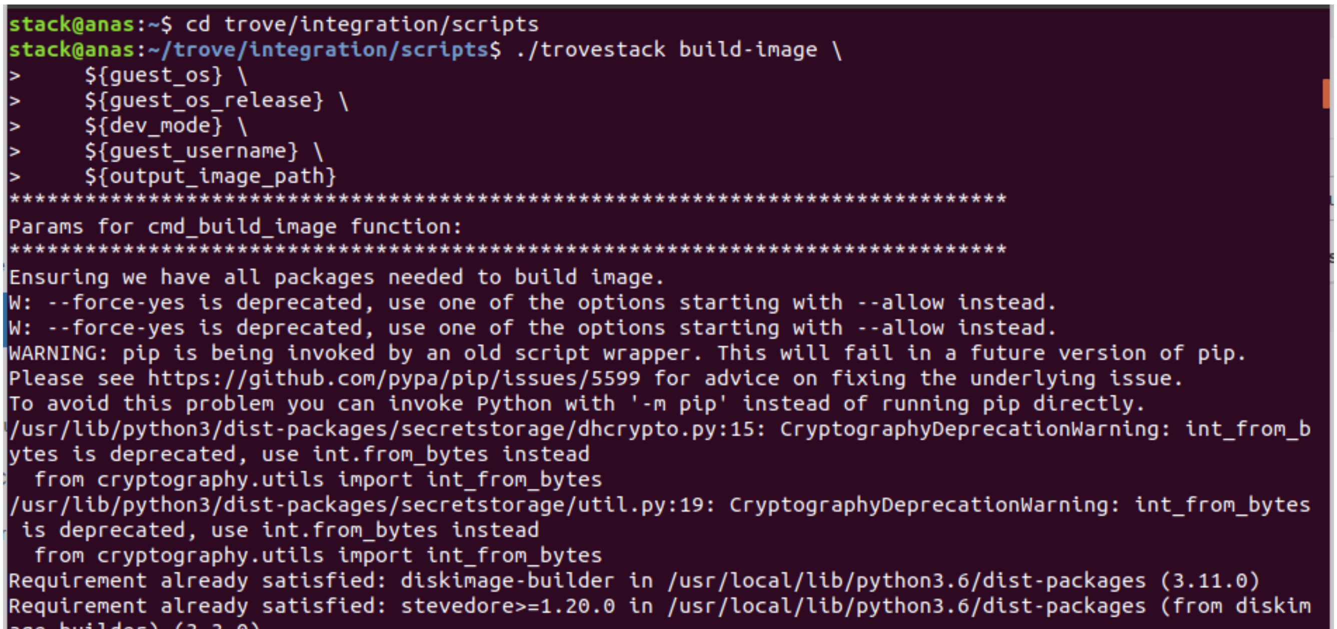
```
enable_service n-sch
enable_service n-api-meta
enable_service placement-api
enable_service placement-client

# Glance
enable_service g-api
enable_service g-reg

# Cinder
enable_service cinder
enable_service c-api
enable_service c-vol
enable_service c-sch

# Neutron
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta

# enable DVR
Q_PLUGIN=ml2
Q_ML2_TENANT_NETWORK_TYPE=vxlan
Q_DVR_MODE=legacy

# Swift
ENABLED_SERVICES+=,swift
SWIFT_HASH=66a3d6b56c1f479c8b4e70ab5c2000f5
SWIFT_REPLICAS=1
```

➢ Trove system info :



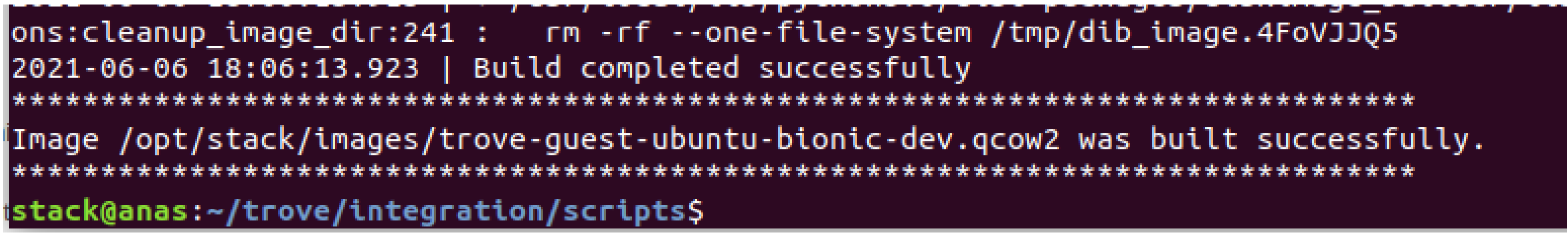
placement	placement	RegionOne	Public	http://10.0.2.15/placement
trove	database	RegionOne	Admin Internal Public	http://10.0.2.15:8779/v1.0/4f1bca20e54546828334cf510 http://10.0.2.15:8779/v1.0/4f1bca20e54546828334cf510 http://10.0.2.15:8779/v1.0/4f1bca20e54546828334cf510

➢ Trove install :



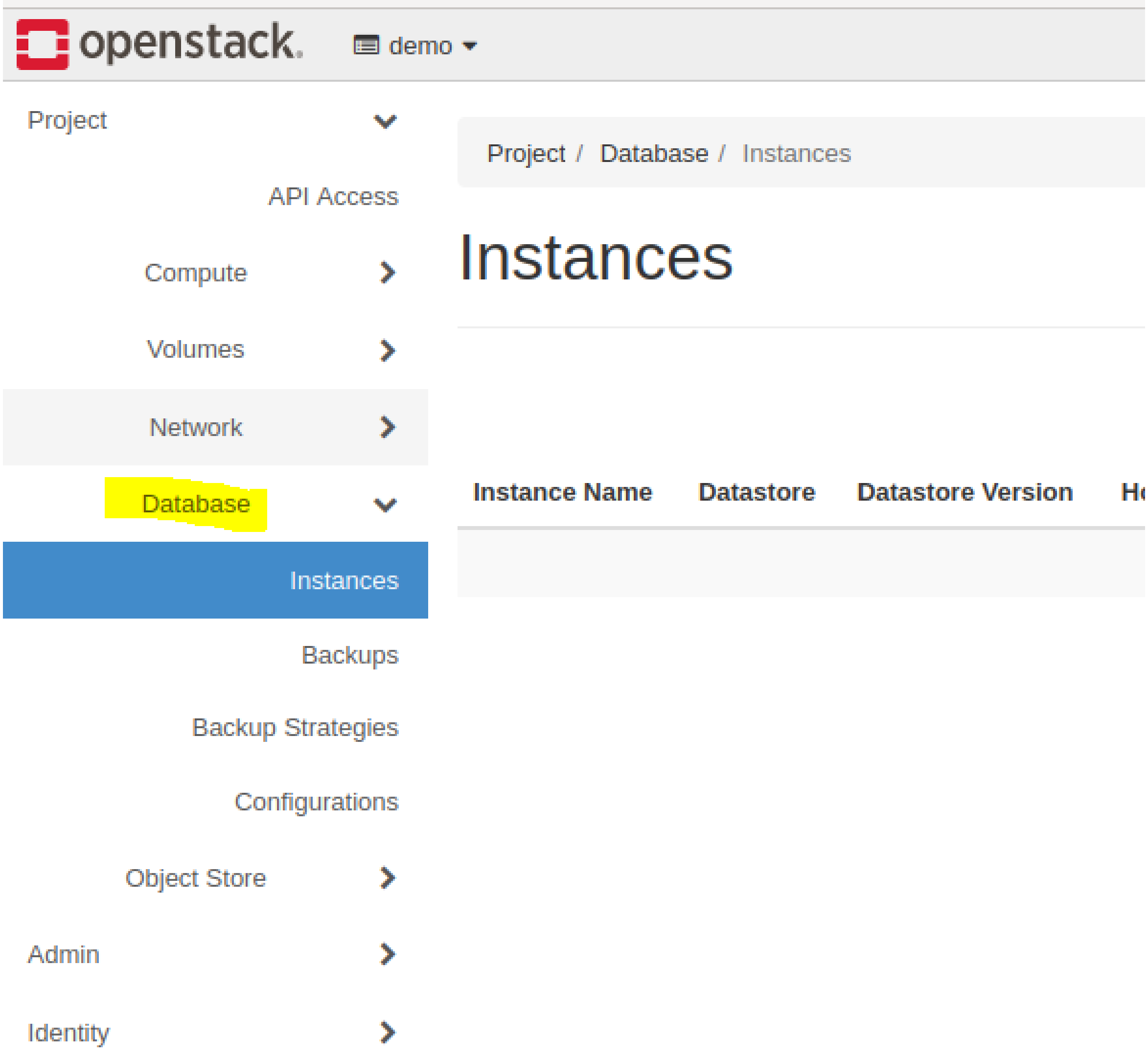
```
stack@anas:~$ cd trove/integration/scripts
stack@anas:~/trove/integration/scripts$ ./trovestack build-image \
>     ${guest_os} \
>     ${guest_os_release} \
>     ${dev_mode} \
>     ${guest_username} \
>     ${output_image_path}
*******************************************************************************
Params for cmd_build_image function:
*******************************************************************************
Ensuring we have all packages needed to build image.
W: --force-yes is deprecated, use one of the options starting with --allow instead.
W: --force-yes is deprecated, use one of the options starting with --allow instead.
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
/usr/lib/python3/dist-packages/secretstorage/dhcrypto.py:15: CryptographyDeprecationWarning: int_from_b
ytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/usr/lib/python3/dist-packages/secretstorage/util.py:19: CryptographyDeprecationWarning: int_from_bytes
 is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Requirement already satisfied: diskimage-builder in /usr/local/lib/python3.6/dist-packages (3.11.0)
Requirement already satisfied: stevedore>=1.20.0 in /usr/local/lib/python3.6/dist-packages (from diskim
age-builder) (3.3.0)
```

- **Trove installed successfully :**

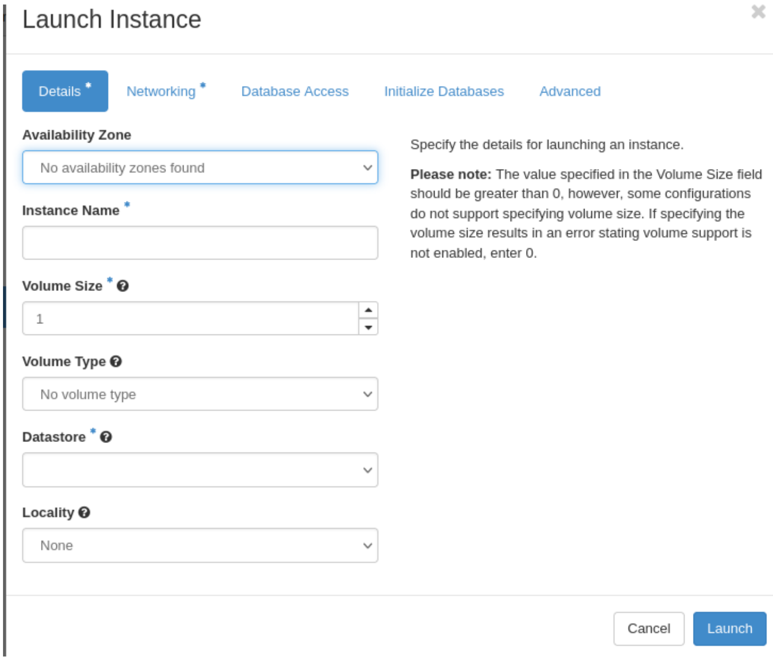


```
ons:cleanup_image_dir:241 :   rm -rf --one-file-system /tmp/dib_image.4FoVJJQ5
2021-06-06 18:06:13.923 | Build completed successfully
*******************************************************************************
Image /opt/stack/images/trove-guest-ubuntu-bionic-dev.qcow2 was built successfully.
*******************************************************************************
stack@anas:~/trove/integration/scripts$
```

- **Trove project (Database as a service) Interface:**



- **Lunch database & Backup Database :**


Launch Instance

Availability Zone: No availability zones found

Instance Name *

Volume Size * : 1

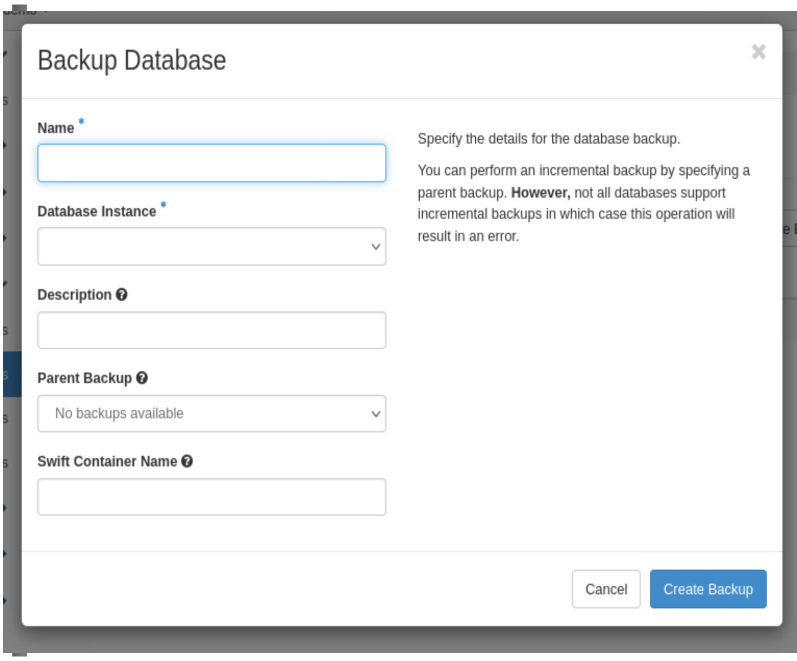
Volume Type: No volume type

Datastore *

Locality: None

Specify the details for launching an instance.

Please note: The value specified in the Volume Size field should be greater than 0, however, some configurations do not support specifying volume size. If specifying the volume size results in an error stating volume support is not enabled, enter 0.


Backup Database

Name *

Database Instance *

Description

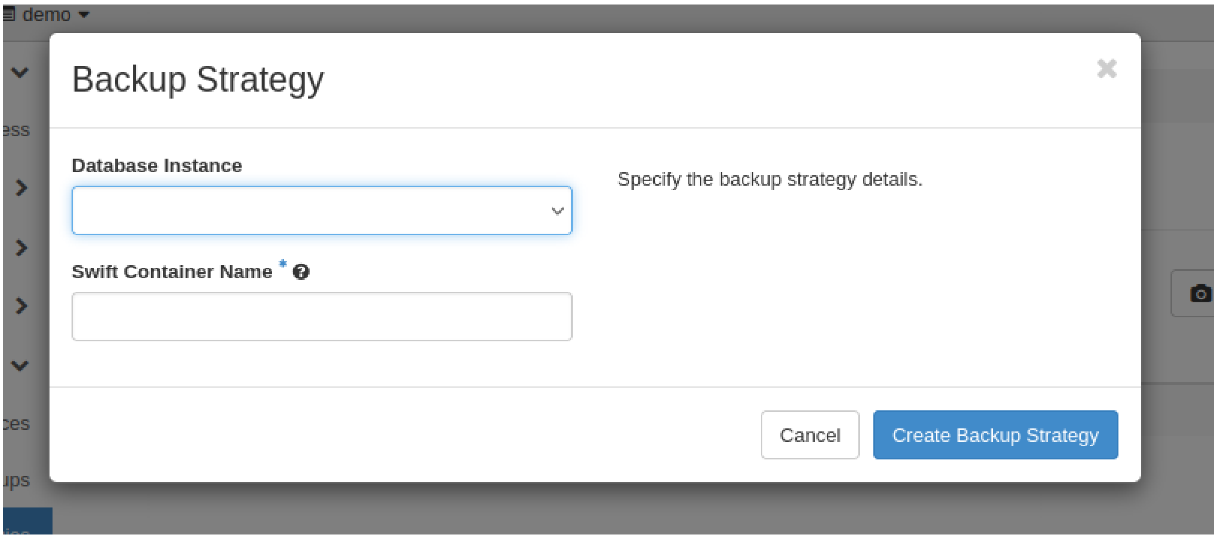
Parent Backup: No backups available

Swift Container Name

Specify the details for the database backup.

You can perform an incremental backup by specifying a parent backup. **However,** not all databases support incremental backups in which case this operation will result in an error.

➢ Backup strategy :

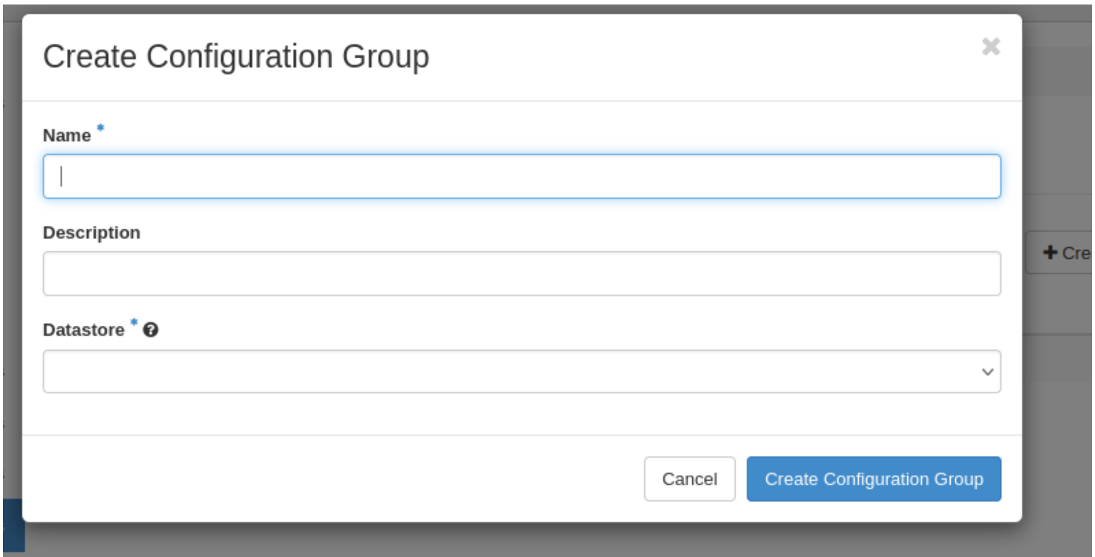

Backup Strategy

Database Instance

Swift Container Name *

Specify the backup strategy details.

➢ Configuration Groups :

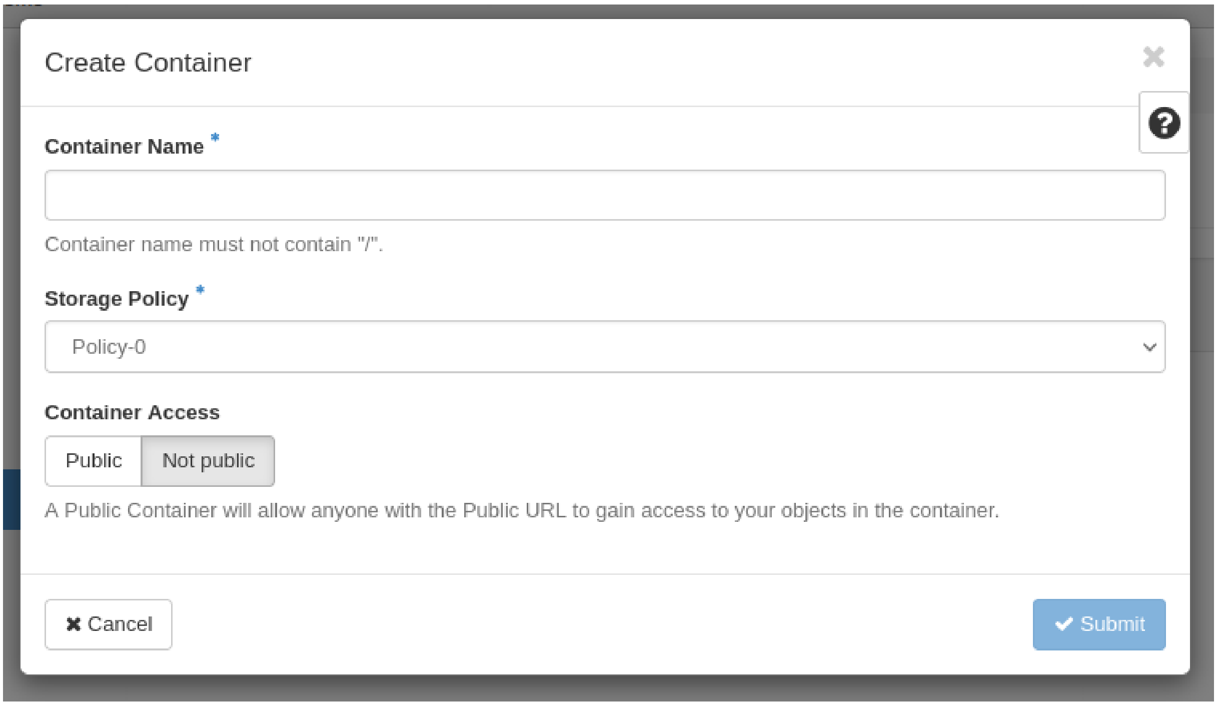

Create Configuration Group

Name *

Description

Datastore *

➢ Container :



Create Container

Container Name *

Container name must not contain "/".

Storage Policy *

Policy-0

Container Access

Public | Not public

A Public Container will allow anyone with the Public URL to gain access to your objects in the container.

Conclusion :

The implementation presented here proved that the users can actually access OpenStack resources and code to improve it or change it according to their needs and then share the results with the entire community for verification and further utilization.

The OpenStack-based system developed here can be a useful choice for users that want to start experiencing the cloud world. It can also be used for small enterprises that want to get a competitive and at the same time affordable cloud platform.

-AMAYOU ANAS-

References :

- https://docs.openstack.org/
- How to install OpenStack on your local machine using Devstack. [Online]. Available from: https://www.mirantis.com/blog/how-to-install-openstack-onyour-local-machine-using-devstack/ 2019.02.15
- All-In-One Single Machine. [Online]. Available from:https://docs.openstack.org/devstack/latest/guides/singlemachine.html 2019.02.15
- Loosely coupled architecture of OpenStack. [Online]. Available from: https://www.researchgate.net/figure/Looselycoupled-architecture-of-OpenStack_fig1_305297793 2019.02.15
- How to Install Single Node OpenStack on CentOS 7. [Online]. Available from: https://www.alibabacloud.com/blog/how-to-install-singlenode-openstack-on-centos-7_594048 2019.02.18
- How to install OpenStack on Ubuntu Server with DevStack. [Online].