Microstack (Openstack) Ussuri Installation and Configuration Documentation

Aim: To install and configure Microstack (Open stack) Ussuri Cloud Stack on the Servers.

Pre-requisites:

At least two machines (one controller and one compute, the controller will also host VMs), each with 16GB RAM, a multi-core processor and at least 50GB of free disk space, connected to a Network, and running Ubuntu 18.04.5 LTS. Configure two machines each as Controller and Compute nodes. All the computers should be connected to the same organization network.

Install Microstack:

Installed MicroStack using the following command, it has to be run on the machine with Ubuntu Server 18.04.5 LTS with the latest patch available by Canonical Linux Distribution.

Controller Node:

Please follow the below commands for the successful installation of Microstack (Beta) Ussuri cloud.

Syntax: **$ ifconfig** – to check whether the ip address is available on the physical machine.

$ sudo apt-get update – to check for the latest packages installed on the physical machine.

```
controller@controller:~$ sudo apt-get update
[sudo] password for controller:
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
```

$ **sudo snap install microstack --beta --devmode** – to start the installation of cloud Stack Ussuri and services.

```
compute@serc:~$ sudo snap install microstack --beta --devmode
2021-02-15T08:55:21Z INFO Waiting for automatic snapd restart...
microstack (beta) ussuri from Canonical✓ installed
```

The version that would be displayed (here, Ussuri) matches the most recent stable Openstack release available with Microstack.

Note: **Microstack installed with the - -devmode flag will not receive updates.**

Initialize Microstack on the Control Machine:

- Run the following command on the machine which was set as the controller node and act accordingly.

Syntax: $ sudo microstack init - -auto - - control

```
controller@controller:~$ ls
controller@controller:~$ sudo snap install microstack --beta  --devmode
[sudo] password for controller:
snap "microstack" is already installed, see 'snap help refresh'
controller@controller:~$ sudo microstack init --auto --control
2021-02-15 08:38:23,377 - microstack_init - INFO - Configuring clustering ...
2021-02-15 08:38:23,646 - microstack_init - INFO - Setting up as a control node.
2021-02-15 08:38:26,716 - microstack_init - INFO - Configuring networking ...
2021-02-15 08:38:35,336 - microstack_init - INFO - Opening horizon dashboard up
to *
2021-02-15 08:38:36,225 - microstack_init - INFO - Waiting for RabbitMQ to start
 ...
Waiting for 10.2.59.42:5672
2021-02-15 08:38:42,963 - microstack_init - INFO - RabbitMQ started!
2021-02-15 08:38:42,963 - microstack_init - INFO - Configuring RabbitMQ ...
2021-02-15 08:38:43,878 - microstack_init - INFO - RabbitMQ Configured!
2021-02-15 08:38:43,941 - microstack_init - INFO - Waiting for MySQL server to s
tart ...
Waiting for 10.2.59.42:3306
2021-02-15 08:39:41,749 - microstack_init - INFO - Mysql server started! Creating databases ...
2021-02-15 08:39:46,275 - microstack_init - INFO - Configuring Keystone Fernet Keys ...
2021-02-15 08:42:24,068 - microstack_init - INFO - Bootstrapping Keystone ...
2021-02-15 08:42:35,398 - microstack_init - INFO - Creating service project ...
2021-02-15 08:42:41,071 - microstack_init - INFO - Keystone configured!
2021-02-15 08:42:41,138 - microstack_init - INFO - Configuring the Placement service...
2021-02-15 08:43:01,748 - microstack_init - INFO - Running Placement DB migrations...
2021-02-15 08:43:19,720 - microstack_init - INFO - Configuring nova control plane services ...
2021-02-15 08:43:31,625 - microstack_init - INFO - Running Nova API DB migrations (this may take a lot of time)...
2021-02-15 08:44:59,500 - microstack_init - INFO - Running Nova DB migrations (this may take a lot of time)...
Waiting for 10.2.59.42:8774
2021-02-15 08:54:42,702 - microstack_init - INFO - Creating default flavors...
2021-02-15 08:55:09,000 - microstack_init - INFO - Configuring nova compute hypervisor ...
2021-02-15 08:55:11,217 - microstack_init - INFO - Configuring the Spice HTML5 console service...
2021-02-15 08:55:12,015 - microstack_init - INFO - Configuring Neutron
Waiting for 10.2.59.42:9696
2021-02-15 09:01:50,645 - microstack_init - INFO - Configuring Glance ...
Waiting for 10.2.59.42:9292
2021-02-15 09:03:15,490 - microstack_init - INFO - Adding cirros image ...
2021-02-15 09:03:18,894 - microstack_init - INFO - Creating security group rules ...
2021-02-15 09:03:29,774 - microstack_init - INFO - Configuring the Cinder services...
2021-02-15 09:04:27,995 - microstack_init - INFO - Running Cinder DB migrations...
2021-02-15 09:05:33,386 - microstack_init - INFO - restarting libvirt and virtlogd ...
2021-02-15 09:05:55,111 - microstack_init - INFO - Complete. Marked microstack as initialized!
```

- It would take about 15-20 minutes to completely install and configure the physical machine (controller node) with Microstack Ussuri.
- On successful completion, get ready for compute node installation (refer to compute node section).
- To enable the networking between the physical machines and the VM's created on the controller and compute nodes, use the following commands:

$ microstack.openstack subnet set --dhcp external-subnet

```
controller@controller:~$ microstack.openstack subnet set --dhcp external-subnet
controller@controller:~$
```

$ microstack.openstack subnet set --dhcp test-subnet

```
controller@controller:~$ microstack.openstack subnet set --dhcp test-subnet
controller@controller:~$
```

$ microstack.openstack subnet set --dns-nameserver 8.8.8.8 external-subnet

```
controller@controller:~$ microstack.openstack subnet set --dns-nameserver 8.8.8.8 external-subnet
controller@controller:~$
```

$ microstack.openstack subnet set --dns-nameserver 8.8.8.8 test-subnet

```
controller@controller:~$ microstack.openstack subnet set --dns-nameserver 8.8.8.8 test-subnet
controller@controller:~$ 
```

$ microstack.openstack network set --share external

```
controller@controller:~$ microstack.openstack network set --share external
controller@controller:~$ 
```

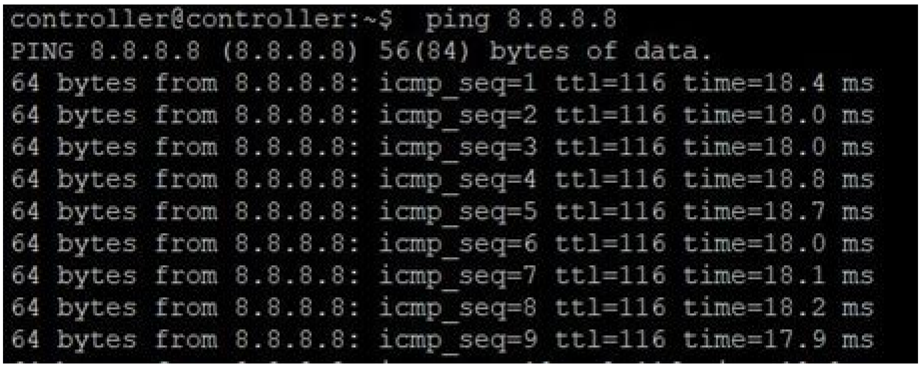
$ microstack.openstack network set --share test

```
controller@controller:~$ microstack.openstack network set --share test
```

$ microstack.openstack subnet show external-subnet

```
controller@controller:~$ microstack.openstack subnet show external-subnet
+----------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------+
| Field                | Value                                                                                                                                                  |
+----------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------+
| allocation_pools     | 10.20.20.2-10.20.20.254
| cidr                 | 10.20.20.0/24
| created_at           | 2021-02-15T09:01:33Z
| description          |
| dns_nameservers      | 8.8.8.8
| dns_publish_fixed_ip | None
| enable_dhcp          | True
| gateway_ip           | 10.20.20.1
| host_routes          |
| id                   | 1ad3f332-d807-426b-bb04-a3b26960a28a
| ip_version           | 4
| ipv6_address_mode    | None
| ipv6_ra_mode         | None
| location             | cloud='', project.domain_id=, project.domain_name='default', project.id='f92cd36e6acc48238ee9727e0cafafbf', project.name='admin', region_name='', zone= |
| name                 | external-subnet
| network_id           | 58eb6a65-bd71-476c-b316-0fdd5dc3405f
| prefix_length        | None
| project_id           | f92cd36e6acc48238ee9727e0cafafbf
| revision_number      | 2
| segment_id           | None
| service_types        |
| subnetpool_id        | None
| tags                 |
| updated_at           | 2021-02-15T09:13:39Z
+----------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------+
controller@controller:~$ 
```

$ ping 8.8.8.8



```
controller@controller:~$  ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=18.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=18.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=18.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=18.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=18.7 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=18.0 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=116 time=18.1 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=116 time=18.2 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=116 time=17.9 ms
```

The above commands would enable the networking **dhcp** service to communicate between the other physical machine along with VM and servers.

__Note:__ Ensure that the servers are talking to each other using ping command, please try to ping the ip address of the nodes.

Reference:
https://stackoverflow.com/questions/64320242/microstack-my-vms-cannot-access-the-internet

- Configure the IP tables as mentioned in the following command

$ sudo iptables -t nat -A POSTROUTING -s 10.20.20.1/24 ! -d 10.20.20.1/24 -j MASQUERADE

```
controller@controller:~$ sudo iptables -t nat -A POSTROUTING -s 10.20.20.1/24 ! -d 10.20.20.1/24 -j MASQUERADE
[sudo] password for controller:
controller@controller:~$
```

$ sudo sysctl net.ipv4.ip_forward=1

```
controller@controller:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
controller@controller:~$
```

<u>Note:</u> This would allow the machines to talk to the VM and other machines that connect externally to the controller.

To list available Openstack endpoints run the following command:

$ microstack.openstack catalog list – This would return the following:

```
+-----------+-----------+----------------------------------------+
| Name      | Type      | Endpoints                              |
+-----------+-----------+----------------------------------------+
| keystone  |  identity | microstack                             |
|           |           | public: http://10.20.20.1:5000/v3/     |
|           |           | microstack                             |
|           |           | internal: http://10.20.20.1:5000/v3/   |
|           |           | microstack                             |
|           |           | admin: http://10.20.20.1:5000/v3/      |
|           |           |                                        |
| glance    | image     | microstack                             |
|           |           |   admin: http://10.20.20.1:9292        |
|           |           | microstack                             |
|           |           |   public: http://10.20.20.1:9292       |
|           |           | microstack                             |
|           |           |   internal: http://10.20.20.1:9292     |
|           |           |                                        |
| neutron   | network   | microstack                             |
|           |           |   public: http://10.20.20.1:9696       |
|           |           | microstack                             |
|           |           |   admin: http://10.20.20.1:9696        |
|           |           | microstack                             |
|           |           |   internal: http://10.20.20.1:9696     |
|           |           |                                        |
| nova      | compute   | microstack                             |
|           |           |   public: http://10.20.20.1:8774/v2.1  |
|           |           | microstack                             |
|           |           |   admin: http://10.20.20.1:8774/v2.1   |
|           |           | microstack                             |
|           |           |   internal: http://10.20.20.1:8774/v2.1 |
|           |           |                                        |
| placement | placement | microstack                             |
|           |           |   public: http://10.20.20.1:8778       |
|           |           | microstack                             |
|           |           |   admin: http://10.20.20.1:8778        |
|           |           | microstack                             |
|           |           |   internal: http://10.20.20.1:8778     |
|           |           |                                        |
+-----------+-----------+----------------------------------------+
```

<u>Launch an Instance</u>

Launch an Instance to check whether the network is correctly configured and there is no error. These commands need to be used on the controller node..

The quickest way to launch the first openstack instance (or VM) command is as follows:

$ microstack launch cirros - -name test

```
controller@controller:~$ microstack launch cirros --name test
Creating local "microstack" ssh key at /home/controller/snap/microstack/common/.ssh/id_microstack
Launching server ...
Allocating floating ip ...
Server test launched! (status is BUILD)

Access it with `ssh -i /home/controller/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.34`
You can also visit the OpenStack dashboard at http://10.20.20.1:80
```

The resulting output provides the information you need to SSH to the instance:

Access it with `ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.123` or any IP dynamically assigned.

Note that the IP address of the instance may be different in your environment. In order to connect to the instance, run the 'ssh' command from the above output.

Note that the IP address of the instance may be different in your environment. In order to connect to the instance, run the 'ssh' command from the above output.

$ ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.123 or any ip address assigned by microstack.

```
Access it with `ssh -i /home/controller/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.34`
You can also visit the OpenStack dashboard at http://10.20.20.1:80
controller@controller:~$ ^C
controller@controller:~$ ssh -i /home/controller/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.34
The authenticity of host '10.20.20.34 (10.20.20.34)' can't be established.
ECDSA key fingerprint is SHA256:U1gYIKU9DLTD4Gr4a0cb6dLNoqbGhFJrdXXSAaRABjg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.20.20.34' (ECDSA) to the list of known hosts.
$
```

You are now connected to your first instance on your OpenStack cluster. You can start playing with it by executing various commands for example:

Syntax : **$ uptime** (use inside the created new instance)

```
uptime
15:17:19 up 5 min, 1 users, load average: 0.00, 0.00, 0.00
```

You can now try to connect to the network of the instance created by using the following commands:

$ ifconfig – to check the ip configuration

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:3E:8D:91
          inet addr:192.168.222.186  Bcast:192.168.222.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe3e:8d91/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1442  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15787 (15.4 KiB)  TX bytes:13958 (13.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

$ ping 8.8.8.8 – to connect to the external network from the new instances.

```
$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=114 time=21.041 ms
64 bytes from 8.8.8.8: seq=1 ttl=114 time=19.735 ms
64 bytes from 8.8.8.8: seq=2 ttl=114 time=18.328 ms
```

$ ping www.google.com to check that ping from the new instance to external network.

```
$ ping www.google.com
PING www.google.com (142.250.71.4): 56 data bytes
64 bytes from 142.250.71.4: seq=0 ttl=114 time=21.549 ms
64 bytes from 142.250.71.4: seq=1 ttl=114 time=19.942 ms
^C
```

If successful, you can exit the instance. And proceed for the installation and configuration on the keystone service on the controller Node, as per the following commands:

$ microstack.openstack hypervisor list

```
controller@controller:~$ microstack.openstack hypervisor list
+----+---------------------+-----------------+------------+-------+
| ID | Hypervisor Hostname | Hypervisor Type | Host IP    | State |
+----+---------------------+-----------------+------------+-------+
|  1 | controller          | QEMU            | 10.2.59.42 | up    |
|  2 | serc                | QEMU            | 10.2.59.41 | up    |
+----+---------------------+-----------------+------------+-------+
```

$ cd /var/snap/microstack/common/etc/horizon/local_settings.d/ (this is used to go to the horizon's external settings)

$ ls

$ sudo vi _05_snap_tweaks.py (Inside the file change it to the below) (instead of vi use vim)

```
root@serc-siren1:~# cd /var/snap/microstack/common/etc/horizon/local_settings.d/
root@serc-siren1:/var/snap/microstack/common/etc/horizon/local_settings.d# ls
_05_snap_tweaks.py
root@serc-siren1:/var/snap/microstack/common/etc/horizon/local_settings.d# vim _05_snap_tweaks.py
```

Under _05_snap_tweaks.py change the line to **ALLOWED_HOSTS ="*" or remove the split(",")** as shown below . i need to be pressed for inserting into the file and **:wq** can be used to save the file. (This is used for the interaction of the external machine to the controller node to launch the Horizon UI).

```
# Point us at keystone.
OPENSTACK_HOST = "10.20.20.1"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"

# Turn off external access for now.  (This should be turned on once we
# have hooks for setting a non default password.)
ALLOWED_HOSTS = "*".split(",")

# Use memcached as our caching backend.
CACHES = {
    'default': {
        #
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '10.20.20.1:11211',
    }
}
SESSION_ENGINE='django.contrib.sessions.backends.cache'
```

$ sudo snap set microstack config.network.ports.dashboard=5001

```
controller@controller:~$ sudo snap set microstack config.network.ports.dashboard=5001
controller@controller:~$
```

To check the all the lists properly installed on the controller with the Microstack please follow:

$ microstack.openstack server list

```
controller@controller:~$ microstack.openstack server list
+--------------------------------------+------+--------+----------------------------------+--------+---------+
| ID                                   | Name | Status | Networks                         | Image  | Flavor  |
+--------------------------------------+------+--------+----------------------------------+--------+---------+
| 17e6cc6b-00de-4a69-81a4-8f2675012cbe | test | ACTIVE | test=192.168.222.186, 10.20.20.34 | cirros | m1.tiny |
+--------------------------------------+------+--------+----------------------------------+--------+---------+
```

$ microstack.openstack network list

```
controller@controller:~$ microstack.openstack network list
+--------------------------------------+----------+--------------------------------------+
| ID                                   | Name     | Subnets                              |
+--------------------------------------+----------+--------------------------------------+
| 10cc93ed-f0cd-4ad7-a54c-b0b1dcbd0e14 | test     | c5a43dd5-91b4-4514-9d80-712e8ad0f209 |
| 58eb6a65-bd71-476c-b316-0fdd5dc3405f | external | 1ad3f332-d807-426b-bb04-a3b26960a28a |
+--------------------------------------+----------+--------------------------------------+
```

$ microstack.openstack flavor list

```
controller@controller:~$ microstack.openstack flavor list
+----+-----------+-------+------+-----------+-------+-----------+
| ID | Name      |   RAM | Disk | Ephemeral | VCPUs | Is Public |
+----+-----------+-------+------+-----------+-------+-----------+
| 1  | m1.tiny   |   512 |    1 |         0 |     1 | True      |
| 2  | m1.small  |  2048 |   20 |         0 |     1 | True      |
| 3  | m1.medium |  4096 |   20 |         0 |     2 | True      |
| 4  | m1.large  |  8192 |   20 |         0 |     4 | True      |
| 5  | m1.xlarge | 16384 |   20 |         0 |     8 | True      |
+----+-----------+-------+------+-----------+-------+-----------+
```

$ microstack.openstack keypair list

```
controller@controller:~$ microstack.openstack keypair list
+------------+-------------------------------------------------+
| Name       | Fingerprint                                     |
+------------+-------------------------------------------------+
| microstack | 73:1a:00:d7:9c:cc:3c:8f:24:73:9d:fc:06:23:8c:76 |
+------------+-------------------------------------------------+
```

$ microstack.openstack image list

```
controller@controller:~$ microstack.openstack image list
+--------------------------------------+--------+--------+
| ID                                   | Name   | Status |
+--------------------------------------+--------+--------+
| 4b0c3d24-096a-4445-9011-28500bcef89c | cirros | active |
+--------------------------------------+--------+--------+
controller@controller:~$
```

$ microstack.openstack security group rule list

```
controller@controller:~$ microstack.openstack security group rule list
+--------------------------------------+-------------+-----------+-----------+------------+--------------------------------------+--------------------------------------+
| ID                                   | IP Protocol | Ethertype | IP Range  | Port Range | Remote Security Group                | Security Group                       |
+--------------------------------------+-------------+-----------+-----------+------------+--------------------------------------+--------------------------------------+
| 0c71b289-d7cc-492a-87ab-9839ed15db6c | None        | IPv4      | 0.0.0.0/0 |            | None                                 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| 1665a68d-42e4-493f-817d-08972c007b4d | None        | IPv4      | 0.0.0.0/0 |            | None                                 | 2c1a52cb-80e6-4aea-b496-9e8691fd405e |
| 1c61577e-a3ac-4a07-9d30-af2a3ad3469e | None        | IPv6      | ::/0      |            | None                                 | 2c1a52cb-80e6-4aea-b496-9e8691fd405e |
| 40b0c389-4dca-4082-b261-6f32164c1bc2 | None        | IPv6      | ::/0      |            | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 |
| 5dd079ce-f289-4058-ac33-906cddd77111 | None        | IPv4      | 0.0.0.0/0 |            | None                                 | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 |
| 64daf691-2147-47af-bdef-aca01214675b | None        | IPv4      | 0.0.0.0/0 |            | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| 705064ae-043a-49e7-a6e0-052a8d194152 | tcp         | IPv4      | 0.0.0.0/0 | 22:22      | None                                 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| a0cce96e-3426-42dc-a118-e2efcf4b2c9f | icmp        | IPv4      | 0.0.0.0/0 |            | None                                 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| a4e44cdc-e612-42e8-8b8c-ae4f05d7b4c8 | None        | IPv6      | ::/0      |            | None                                 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| d04442ae-cc59-4a25-9a48-136311bb74b3 | None        | IPv6      | ::/0      |            | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 | 91da22d5-0c73-4e99-8ebc-f7244f0d5404 |
| ec143fc5-a6e3-4693-81eb-bd352d10dff7 | None        | IPv4      | 0.0.0.0/0 |            | 2c1a52cb-80e6-4aea-b496-9e8691fd405e | 2c1a52cb-80e6-4aea-b496-9e8691fd405e |
| f1f0c6d1-bf2c-4451-9965-f76afaaae1a2 | None        | IPv6      | ::/0      |            | 2c1a52cb-80e6-4aea-b496-9e8691fd405e | 2c1a52cb-80e6-4aea-b496-9e8691fd405e |
| f72e10e7-9f0b-4570-ba9c-e6714f5d7e8a | None        | IPv4      | 0.0.0.0/0 |            | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 |
| ffe21f19-9b6e-4b3a-85b7-1b5a2098f14f | None        | IPv6      | ::/0      |            | None                                 | 8190b165-4ea4-4492-9d35-7ab1db1ea5f2 |
+--------------------------------------+-------------+-----------+-----------+------------+--------------------------------------+--------------------------------------+
controller@controller:~$
```

<u>Installation and Configuration of Compute Node:</u>

Run the following command on a machine with Ubuntu Server 18.04.5 LTS with the latest patch available by Canonical Linux Distribution.

<u>Compute Node:</u>

Please follow the below commands for the successful installation of Microstack (Beta) ussuri cloud.

Syntax : **$ ifconfig** – to check whether the ip address is available on the physical machine.

```
compute@serc:~$ ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.2.59.41  netmask 255.255.252.0  broadcast 10.2.59.255
        inet6 fe80::8eec:4bff:fe6e:d84e  prefixlen 64  scopeid 0x20<link>
        ether 8c:ec:4b:6e:d8:4e  txqueuelen 1000  (Ethernet)
        RX packets 2807  bytes 233326 (233.3 KB)
        RX errors 0  dropped 1056  overruns 0  frame 0
        TX packets 133  bytes 18441 (18.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 103  bytes 8756 (8.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 103  bytes 8756 (8.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether fc:01:7c:6a:c6:13  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

$ sudo apt-get update – to check for the latest packages installed on the physical machine.

```
controller@controller:~$ sudo apt-get update
[sudo] password for controller:
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
```

$ sudo snap install microstack --beta --devmode – To start the installation of cloud archive Ussuri and services.

```
compute@serc:~$ sudo snap install microstack --beta --devmode
2021-02-15T08:55:21Z INFO Waiting for automatic snapd restart...
microstack (beta) ussuri from Canonical✓ installed
```

<u>Generate a compute node connection string</u>

Run the following command on the **controller node or controller machine**. It will generate an encoded string that will be needed when adding a compute node in the next step:

$ sudo microstack add-compute (This command should be running on the controller machine only.).

Copy the assigned key to the below given command. Controller node generates a connection string that needs to be mentioned on the compute node and is valid only for 10-20 min. Regenerate the code if the compute node is not able to recognize the connection-string at the controller-node end.

```
controller@controller:~$ sudo microstack add-compute
[sudo] password for controller:
Use the following connection string to add a new compute node to the cluster (valid for 20 minutes from this moment):
[illegible]
controller@controller:~$
```

Note: The above step needs to be used in the **controller node** or main machine with controller node configured and use the command mentioned above and copy the connection string using the below command:

Run the following command on all machines you want to act as compute nodes. Each will require a unique connection string from the control node:

$ sudo microstack init --auto --compute --join <connection-string> (This command would be running on compute node only). Copy the generated connection string from the controller node.

Note :

1. The connection string is only valid for 10-20 mins.
2. Regenerate the connection string if the time is elapsed.
3. In case of any errors, regenerate the connection- string.
4. Please check if the controller and compute nodes are talking to each other by pinging the ip address to each other nodes.

```
compute@serc:~$ sudo microstack init --auto --compute --join hKhob3N0bmFtZaoxMC4
yLjU5LjQyq2ZpbmdlcnByaW50xCAschZPFtMTs1rcFlazLxvTssLJg8FZuNeFYtIWpGp8SKJpZNkgZjV
lNjc0Zjg4YmVjNDhkOTk2MTU4NTBmZWUwZjJhMzKmc2VjcmV02SBCbkZxUXd6OTdjcDh3T0N4dFI2alI
tUXBGaTlYLUp3bw==
[sudo] password for compute:
2021-02-15 10:53:26,162 - microstack_init - INFO - Configuring clustering ...
2021-02-15 10:53:26,480 - microstack_init - INFO - Setting up as a compute node.
2021-02-15 10:53:30,119 - microstack_init - INFO - Configuring networking ...
2021-02-15 10:53:36,450 - microstack_init - INFO - Opening horizon dashboard up to *
2021-02-15 10:53:37,312 - microstack_init - INFO - Disabling local rabbit ...
2021-02-15 10:53:38,170 - microstack_init - INFO - Disabling local MySQL ...
2021-02-15 10:53:38,892 - microstack_init - INFO - Disabling the Placement service...
2021-02-15 10:53:39,653 - microstack_init - INFO - Disabling nova control plane services ...
2021-02-15 10:53:42,404 - microstack_init - INFO - Configuring nova compute hypervisor ...
2021-02-15 10:53:44,459 - microstack_init - INFO - Configuring the Spice HTML5 console service...
2021-02-15 10:53:57,152 - microstack_init - INFO - Creating security group rules ...
2021-02-15 10:54:02,816 - microstack_init - INFO - Disabling Cinder services...
2021-02-15 10:54:08,479 - microstack_init - INFO - restarting libvirt and virtlogd ...
2021-02-15 10:54:38,418 - microstack_init - INFO - Complete. Marked microstack as initialized!
compute@serc:~$
```

Note:

- Please ensure both the machines are communicating with each other or else the setup on the compute node would fail.
- Please copy the exact string to the above said command from the controller.

Interaction with MicroStack

You can interact with your OpenStack either via the web GUI or the CLI.

Web GUI:

To interact with your OpenStack via the web GUI visit http://10.20.20.1/ and log in with the ‘admin' user. The password is obtained in this way:

$ sudo snap get microstack config.credentials.keystone-password

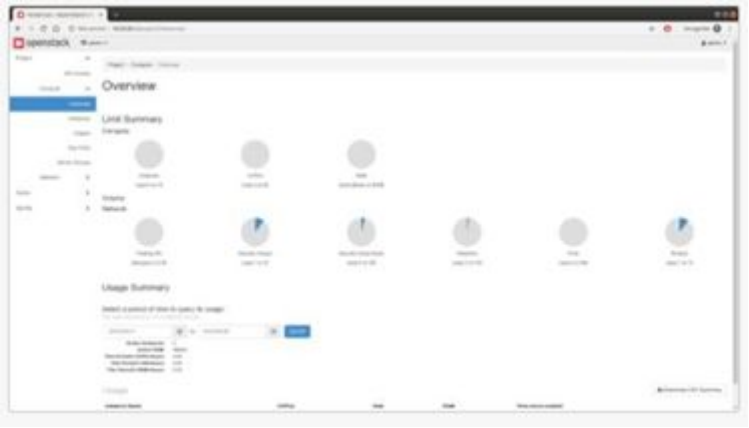
```
--- 10.20.20.71 ping statistics ---
154 packets transmitted, 154 received, 0% packet loss, time 156489ms
rtt min/avg/max/mdev = 0.122/0.415/3.751/0.476 ms
controller@controller:~$ sudo snap get microstack config.credentials.keystone-password
```

Note the default username is admin and to generate the password please use the above command on the controller node.

Type the credentials and press the ‘Sign In’ button:

Output screen:

You should now see the Open stack Dashboard



You can start playing with your local private cloud (i.e. create additional users, launch instances, etc.).

Troubleshooting the horizon UI if you are not able to see the above output.

If you are not able to run or see the Open stack dashboard with the IP http://10.20.20.1/ then please follow the below steps and prerequisites:

Pre requisites:

- Ensure that you have a Linux laptop/desktop connected to the same network and have the terminal access in the laptop.

Configuration Procedure:

Accessing Horizon on a remote server

If you've installed Microstack on a remote server you can use SSH local port forwarding to access Horizon:

$ sudo ssh -i <ssh-key> -N -L 8001:10.20.20.1:80 <user>@<server-ip> or <user>@<controller-node-ip>

Or use the following commands on a ubuntu laptop with the same network that is connected to the controller and compute nodes.

Note: These commands need to be used in the external machine (laptop/computer) connected to the same network.

$ ssh-keygen (To generate the ssh key for the laptop which connects to the horizon UI / controller node)

$ ssh-copy-id <username>@<your-system-ip-address> (This command is to copy the ssh-key to the controller node system using the controller machines - username along with the ip address.)

$ sudo ssh -I <ssh-key-generated> -N -L 8001:10.20.20.1:5001 <username of the controller node>@<ip-address>

```
ganesh@ganesh:~$ sudo su
[sudo] password for ganesh:
root@ganesh:/home/ganesh# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:svucpxNyAmlIeC4cJhp/uAAdjoNcrl59WX6G0oWXFkE root@ganesh
The key's randomart image is:
+---[RSA 2048]----+
| o.o      .E.    |
|B=*       . o    |
|XBo+ .   o =     |
|+o* *   = =      |
| + = o.+S+ o     |
|. o   +o+ o      |
| .    .+ .       |
|       o...      |
|      ..=+       |
root@ganesh:/home/ganesh# ssh -i SHA256:svucpxNyAmlIeC4cJhp/uAAdjoNcrl59WX6G0oWXFkE -N -L 8001:10.20.20.1:5001 controller@10.2.59.42
Warning: Identity file SHA256:svucpxNyAmlIeC4cJhp/uAAdjoNcrl59WX6G0oWXFkE not accessible: No such file or directory.
controller@10.2.59.42's password:
```

.

Hence, you could try accessing the horizon UI by using the said ip address in the browser.

http://10.20.20.1:5001 or http://localhost:8001

Note: If you are not able to follow please refer the link: https://microstack.run/docs/pro-tips

You can save system resources by disabling MicroStack when it's not in use. Please use these commands on the controller node.

$ sudo snap disable microstack

To re-enable :

$ sudo snap enable microstack

Installation Steps for deploying an image and creating an instance on the Openstack on the controller node

- First enable microstack using the following command:
  - **$ sudo snap enable microstack** (Not required if the microstack is enabled).
- Then use the command to check the installed vm's on the microstack

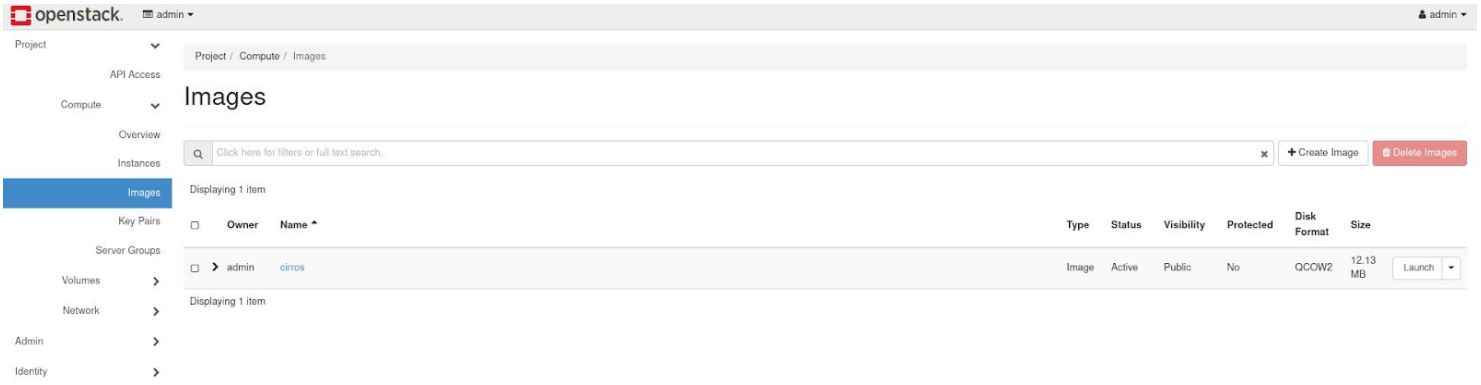
$ microstack.openstack server list (Check for the server list (VM's already created)).

```
controller@controller:~$ microstack.openstack server list
+--------------------------------------+------+--------+-----------------------------------+--------+---------+
| ID                                   | Name | Status | Networks                          | Image  | Flavor  |
+--------------------------------------+------+--------+-----------------------------------+--------+---------+
| 17e6cc6b-00de-4a69-81a4-8f2675012cbe | test | ACTIVE | test=192.168.222.186, 10.20.20.34 | cirros | m1.tiny |
+--------------------------------------+------+--------+-----------------------------------+--------+---------+
```

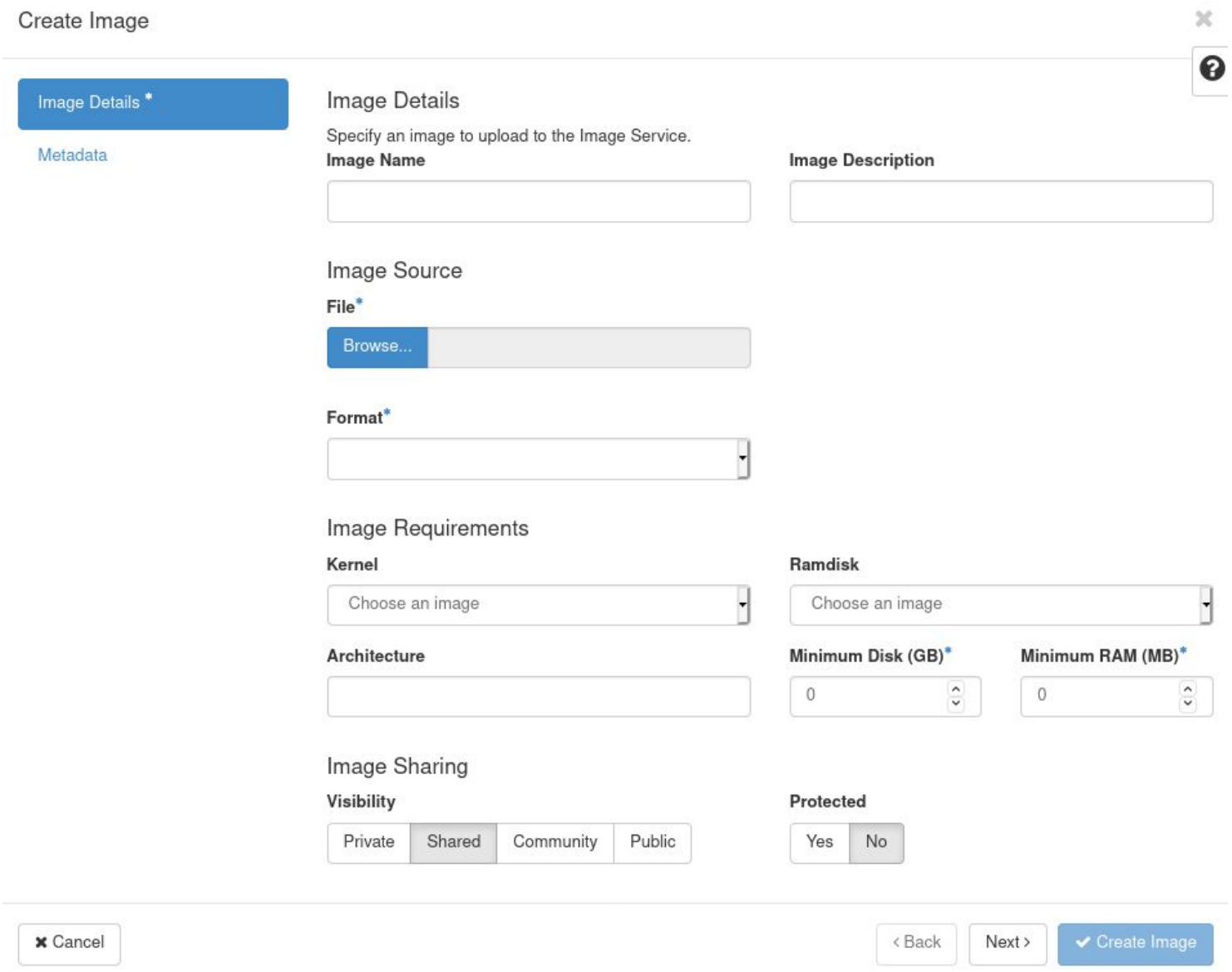
Image creation using Horizon GUI Interface :

To create to the new image on the GUI, please follow the steps given below:

- Login into the horizon UI, on the left side pane, there would be project → Compute → Images



- Click on the create Image button on the right side of the window pane.
- There fill the particular details.



- For the file section in the given above image, use the link to download the Cloud Image source : http://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img

- After download, please select the file from the downloaded location from your local machine.

Create Image

Image Details

Metadata

Image Details

Specify an image to upload to the Image Service.

Image Name

doc1

Image Description

docTest1

Image Source

File*

Browse... bionic-server-cloudimg-amd64.img

Format*

QCOW2 - QEMU Emulator

Image Requirements

Kernel

Choose an image

Ramdisk

Choose an image

Architecture

Minimum Disk (GB)*

0

Minimum RAM (MB)*

0

Image Sharing

Visibility

Private | Shared | Community | Public

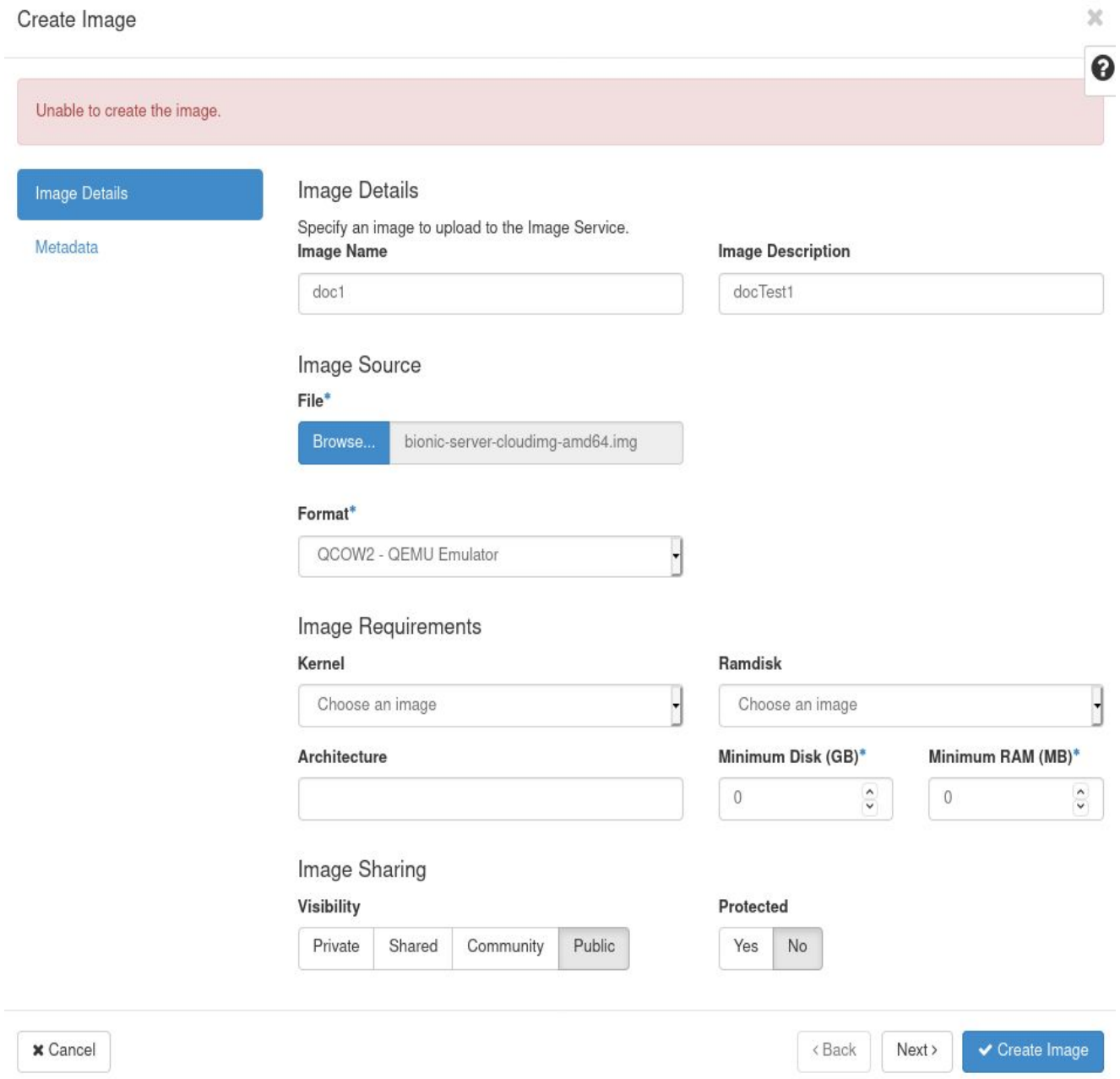
Protected

Yes | No

Cancel

Back | Next | Create Image

- After clicking on the create image, wait for the image to be created. If you are not able to create the image, then move to the controller node for CLI based image creation.



<u>Image creation using CLI on the controller node:</u>

Download the image file of Ubuntu in controller node (controller machine/server)

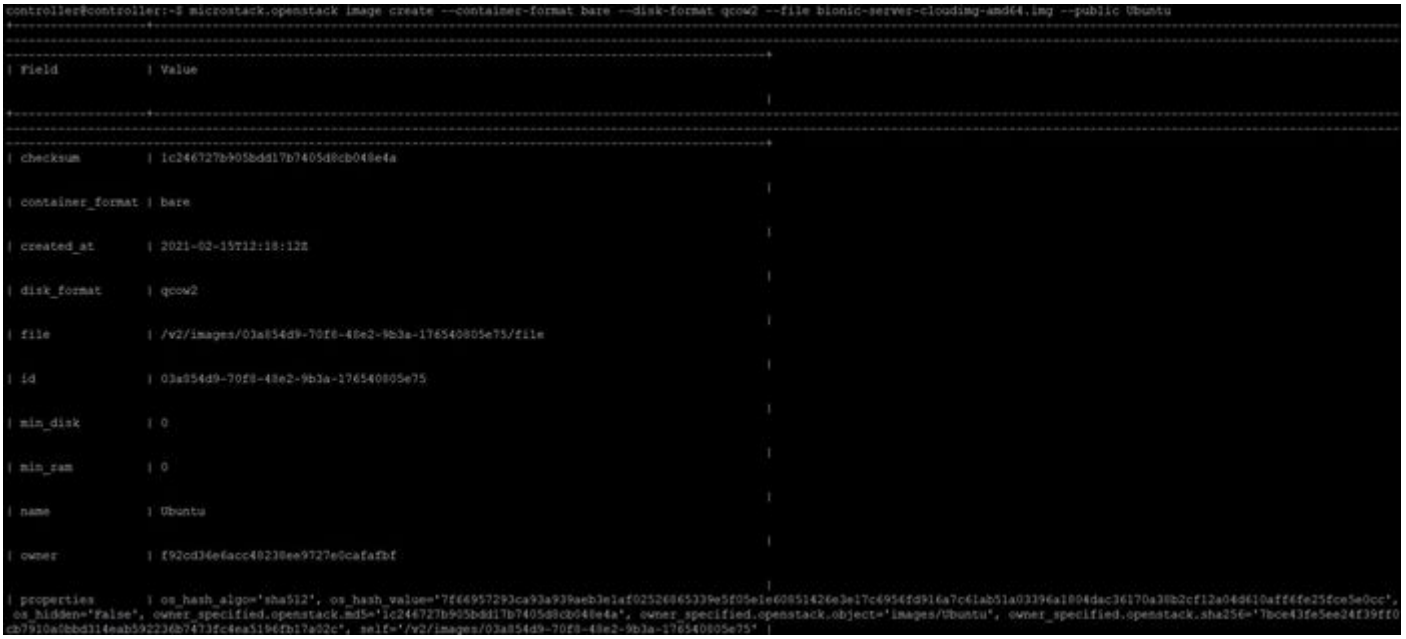
Below are the commands for CLI (command line interface) use wget

§ wget http://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img

Use the below command for the generation of the image in the CLI as given below: Use without root privileges:

- **$microstack.openstack image create --container-format bare --disk-format qcow2 --file bionic-server-cloudimg-amd64.img --public Ubuntu**

  - **Microstack.openstack image create :** To initiate the image creation on the controller node.
  - **--container-format bare :** The instance/container would be assigned empty.
  - **--disk-format qcow2 -** File format for the disk created from the image.
  - **--file bionic-server-cloudimg-amd64.img -** Image file to deploy an image
  - **--public Ubuntu -** Name of the image



· Use the below command on the controller node, to check for image deployment on the controller node and horizon UI

- **$microstack.openstack image list** (Shows the list of Images)

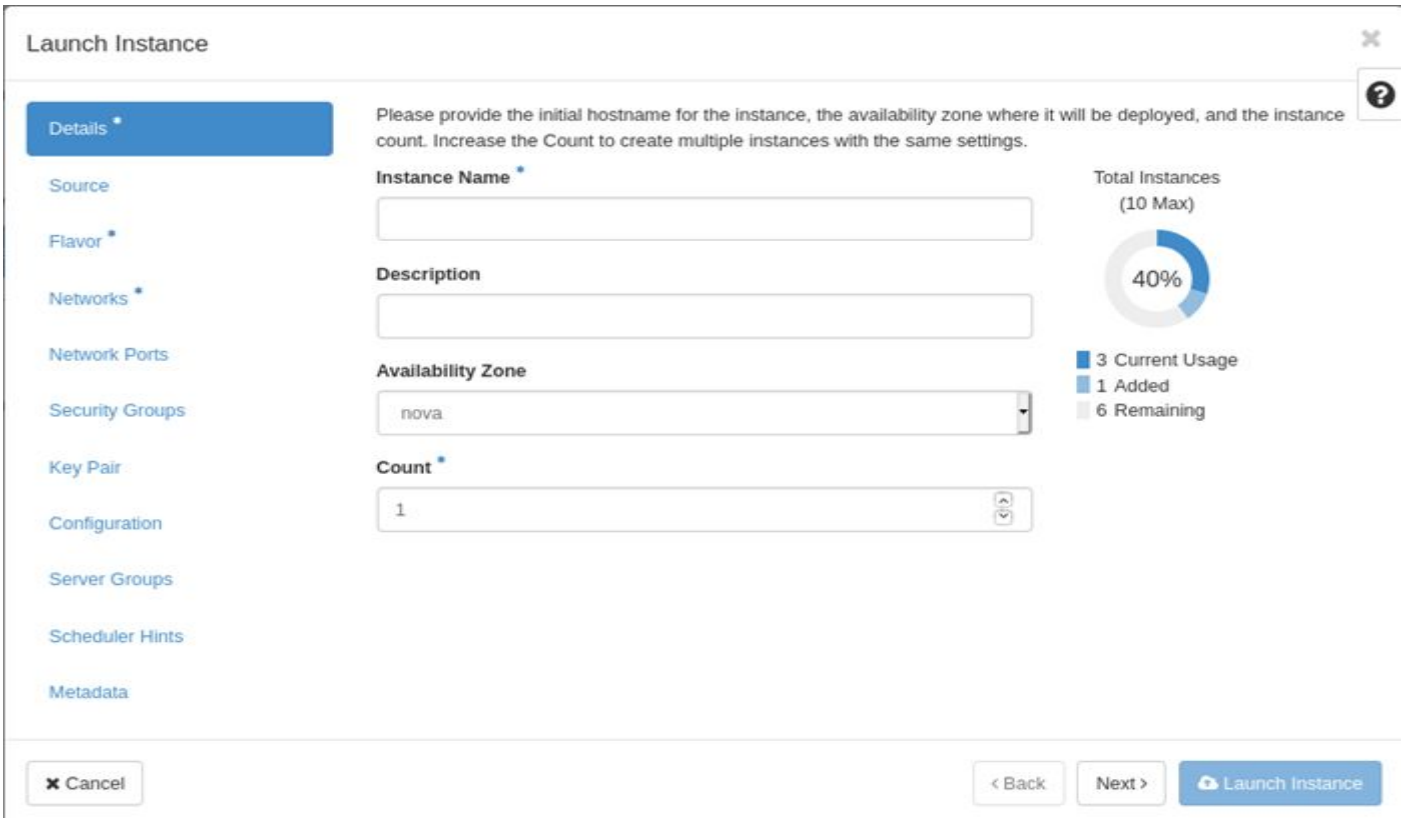
```
controller@controller:~$ microstack.openstack image list
+--------------------------------------+--------+--------+
| ID                                   | Name   | Status |
+--------------------------------------+--------+--------+
| 03a854d9-70f8-48e2-9b3a-176540805e75 | Ubuntu | active |
| 4b0c3d24-096a-4445-9011-28500bcef89c | cirros | active |
+--------------------------------------+--------+--------+
controller@controller:~$
```

<u>Horizon User Interface for the image list : Project → Compute → Images</u>

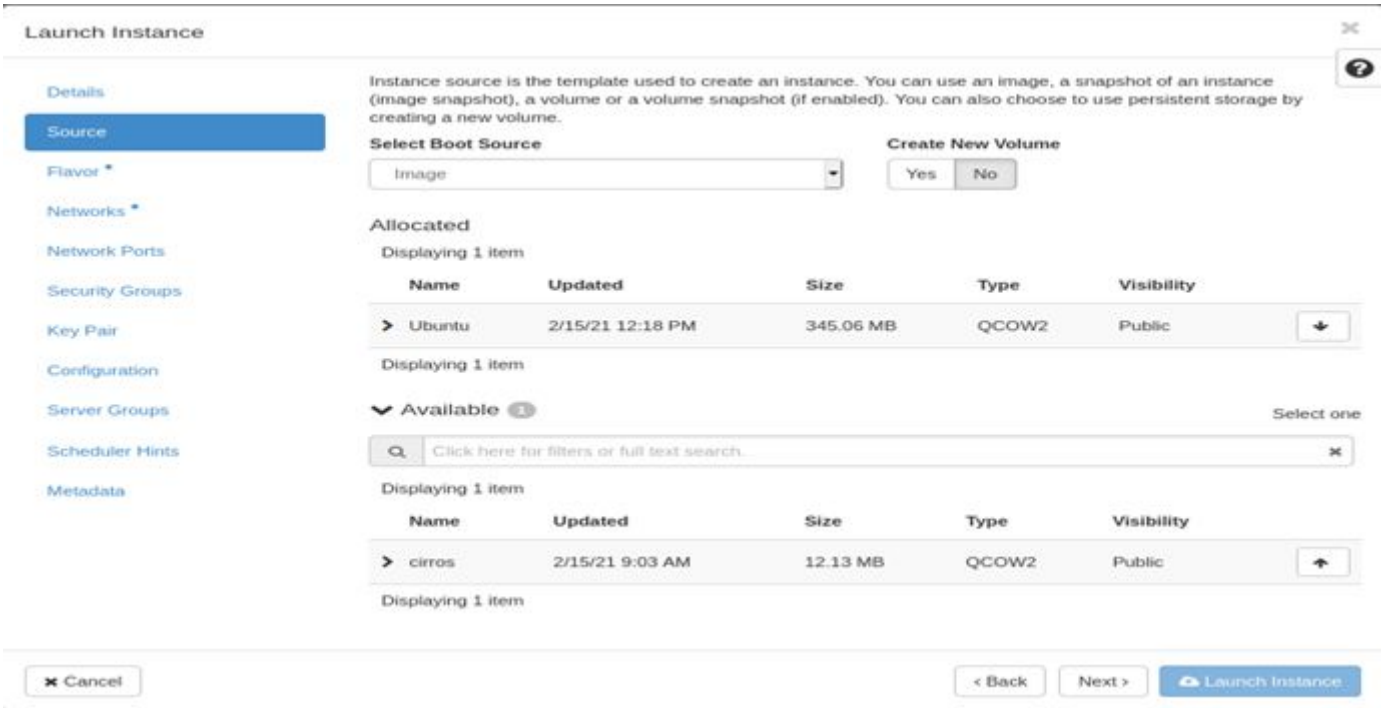
After creating the images, please go to the Project Section → Compute → Instances → Launch Instances.

Under Launch instances:

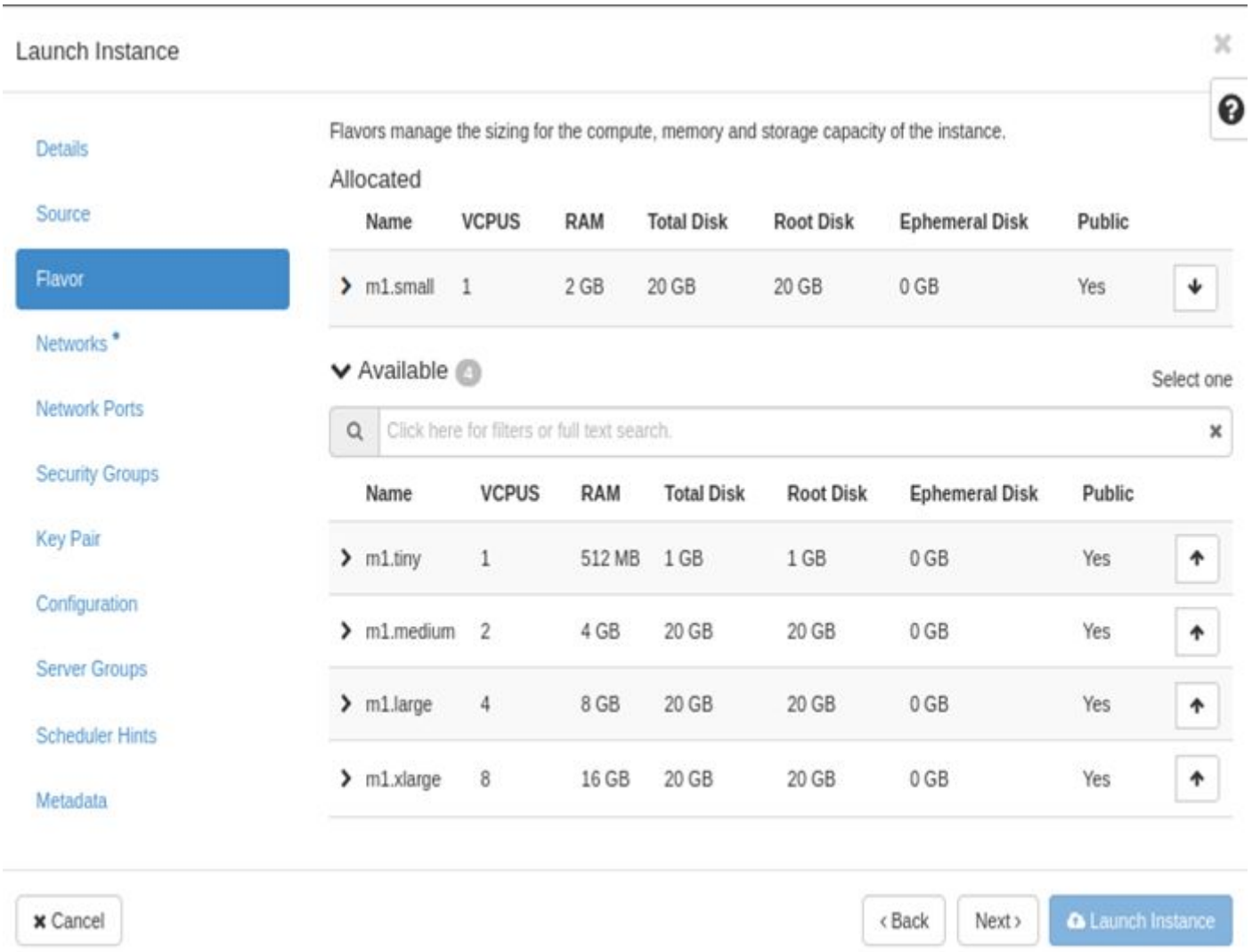
Step 1: Enter the particular details as asked to and mentioned in the image.



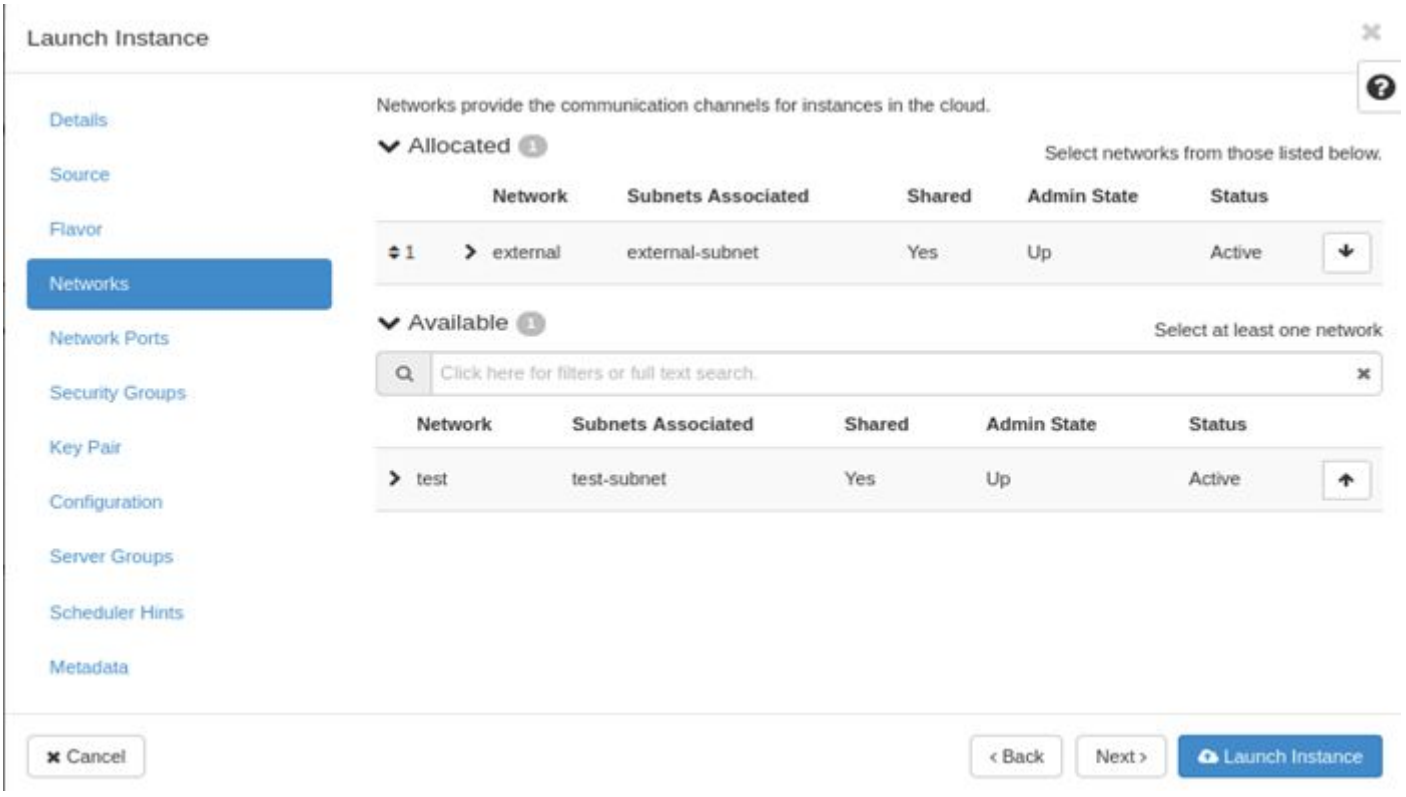
Step 2: After filling the details, please click on Next and select the necessary Image for instance creation.



Step 3: Select the flavor (hard disk size and ram capacity) for the instance.

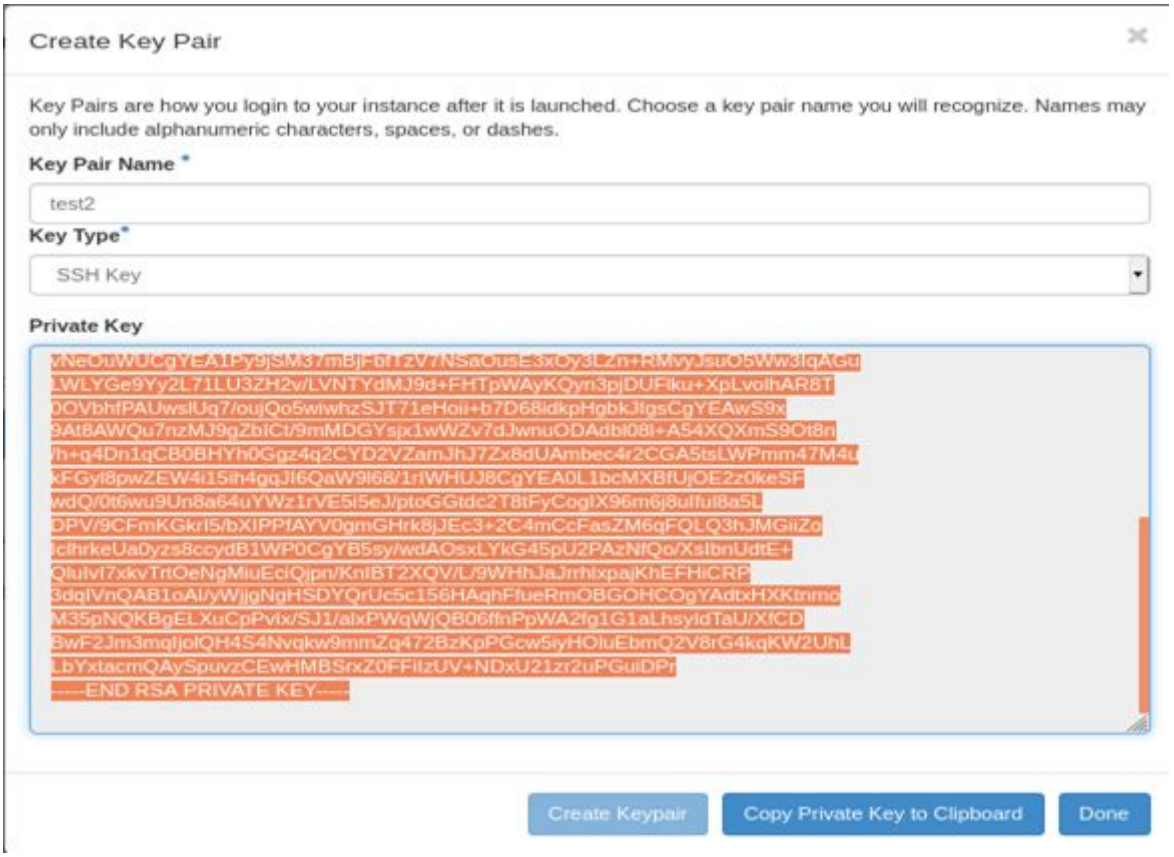


Step 4: Select the network you want to for the instance. (Default – External)

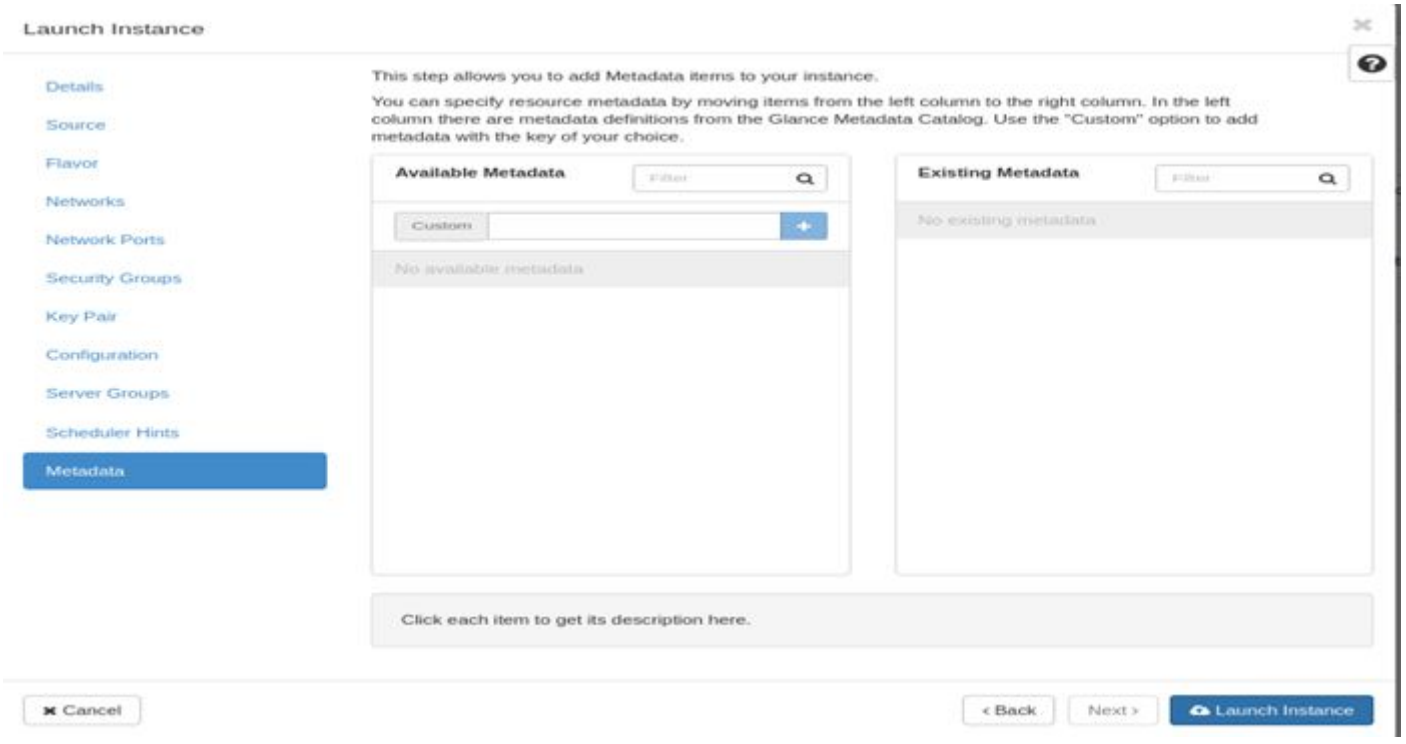


Step 5: After selection of the network, now we need to generate the ssh key pair to connect to the instance from the external or local machines. To generate the key pairs under launch instances, select the keypair and Add New pair and use the following details

By using the ssh key. The private key would be automatically generated, use the key and save it as .pem file on your local machine.



Step 6; After creating the file and the keypair, click done and move to launch instance button. This would create the instance with the specific requirements as mentioned in above steps.



Step 7: After the click, please wait for 1-2 minutes to let the instance be created and once created please make a note of the ip address assigned to the instance.

Step 8: After creating the instance, open a terminal and copy the saved private key to the controller using ssh paired with the ip address. Follow the commands to copy the pem file to the controller.

Local machine terminal commands:

$ ssh-keygen

Bind the keygen using the following command

$ sudo ssh -i <ssh-key> -N -L 8001:10.20.20.1:5001 username@ip-address.

```
sudo ssh -i SHA256:zsp9pqhb+OZjP1xsA2RaDZRbwfrZc44eSXdeH87inpc -N -L 8001:10.20.20.1:5001 raghu@10.4.25.15
```

Then using your machine terminal copy the pem key to the said ssh key to the username@ip -address

$ scp "/path to the pem file" <ssh-key> -i username@ipaddress:

```
scp "/home/ganesh/Desktop/test2.pem" -i SHA256:yx03eEeI7fh//XMyrLJVr21wdTtdIgv7JTdhWaUH6yA controller@10.2.59.42:
```

Using the above command, you can access the new instance created as shown in the example below:

```
ganesh@ganesh:~$ scp "/home/ganesh/Desktop/test2.pem" -i SHA256:yx03eEeI7fh//XMyrLJVr21wdTtdIgv7JTdhWaUH6yA controller@10.2.59.42:
The authenticity of host '10.2.59.42 (10.2.59.42)' can't be established.
ECDSA key fingerprint is SHA256:pFnCEj9KghostGufvNE6HDp61uGYo8pGaMr9hQsZ7A0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.2.59.42' (ECDSA) to the list of known hosts.
controller@10.2.59.42's password:
test2.pem                                                                    100% 1675   490.8KB/s   00:00
-i: No such file or directory
ssh: Could not resolve hostname sha256: Name or service not known
ganesh@ganesh:~$ sudo ssh -i SHA256:yx03eEeI7fh//XMyrLJVr21wdTtdIgv7JTdhWaUH6yA -N -L 8001:10.20.20.1:5001 controller@10.2.59.42
Warning: Identity file SHA256:yx03eEeI7fh//XMyrLJVr21wdTtdIgv7JTdhWaUH6yA not accessible: No such file or directory.
controller@10.2.59.42's password:
```

While accessing the controller, please use the below commands for successful entry into the new instances.

$ chmod 400 file.pem (any file name) This command should be running on both controller and local machine.

$ ssh -i "file.pem" instance-name@ instance – ip-address (use the ssh-key to connect to the newly created instance in the horizon from an external network.)

```
ssh -i ubuntu@10.20.20.27
ls
chmod 400 test2.pem
ssh -i "test2.pem" ubuntu@10.20.20.222
history
```

If the setup is successful, the instance would be launched.

Note: These above commands would only work on the controller node and compute node which have micro stack and other services configured.

Note: Please ensure that the below codes should not be used at all times.

$ sudo lsof -i -P -n | grep LISTEN

```
$ sudo ss -tulwn

$ netstat -tulpn | grep 3000

$ netstat -tulpn | grep 8000

$ netstat -tulpn | grep 5001

$ netstat -tulpn | grep 8001

$ netstat

$ sudo lsof -i -P -n | grep LISTEN

$ sudo ss -tulw

$ ps aux | grep ssh

$ netstat -tulpn | grep 8001

$ netstat -tulpn | grep 5001

$ killall ssh

$ pkill ssh

$ systemctl status sshd

$ systemctl start sshd

$ systemctl status sshd

$ ps aux | grep ssh
```

Cheat Codes:

To disable Microstack.

$ sudo snap disable microstack

To re-enable:

$ sudo snap enable microstack.