

In The Name Of God

Group 8

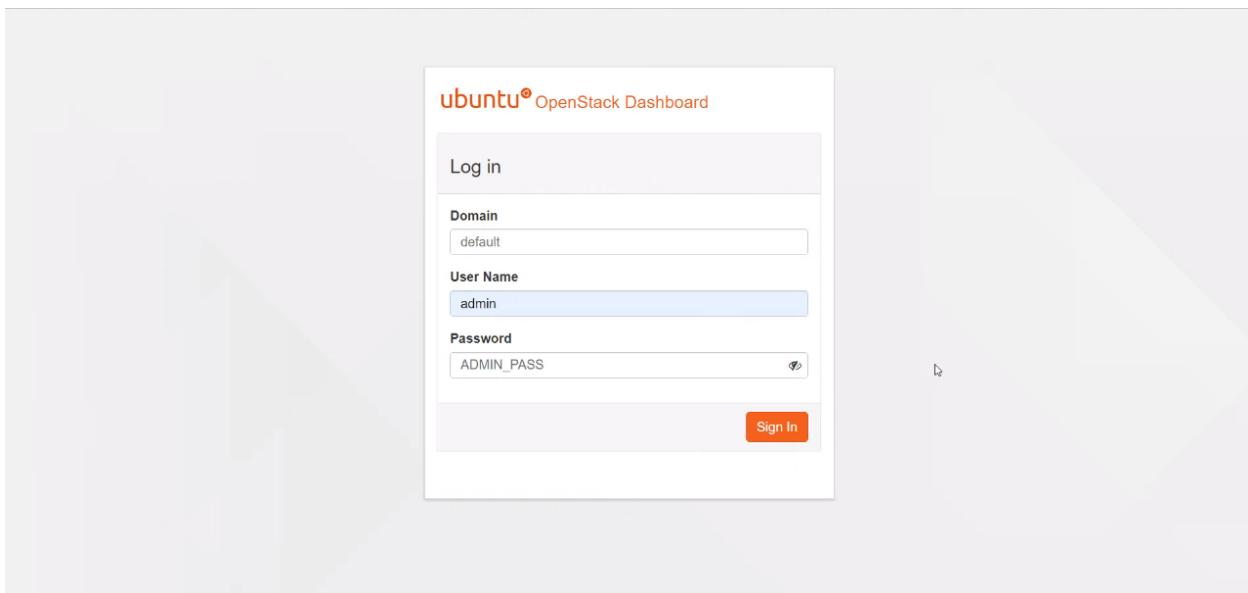
Amir Arsalan Yavari - 9830253 Mohammad Mahdi Barghi - 9818453

OPENSTACK HomeWork

January 08, 2023

Step 0:

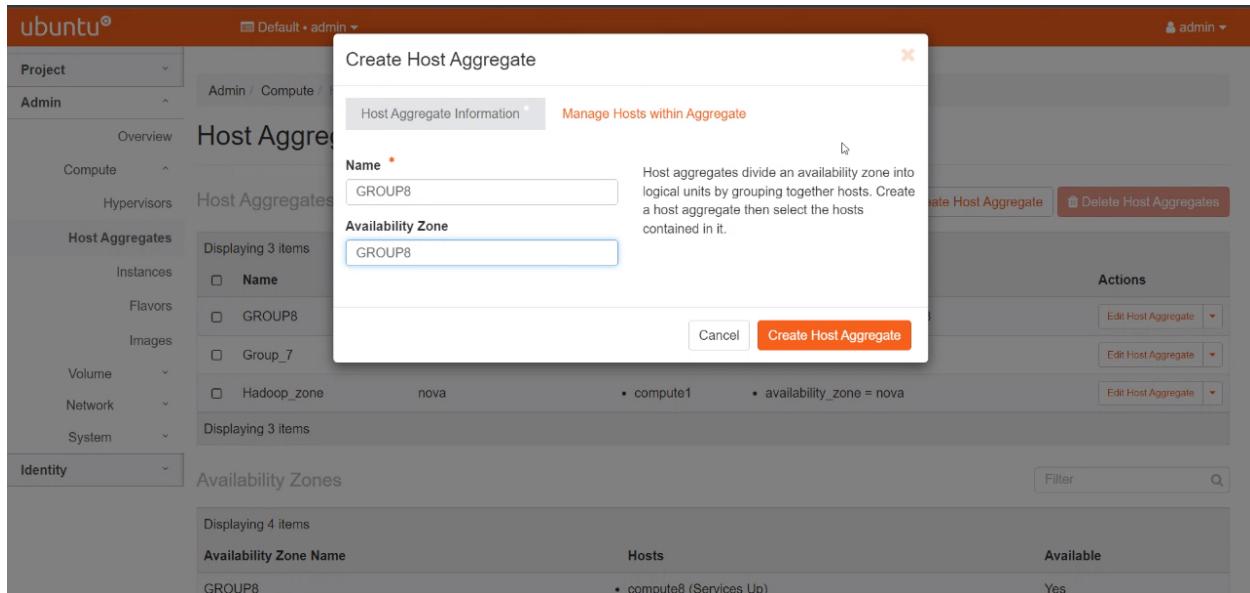
Login to dashboard:



2

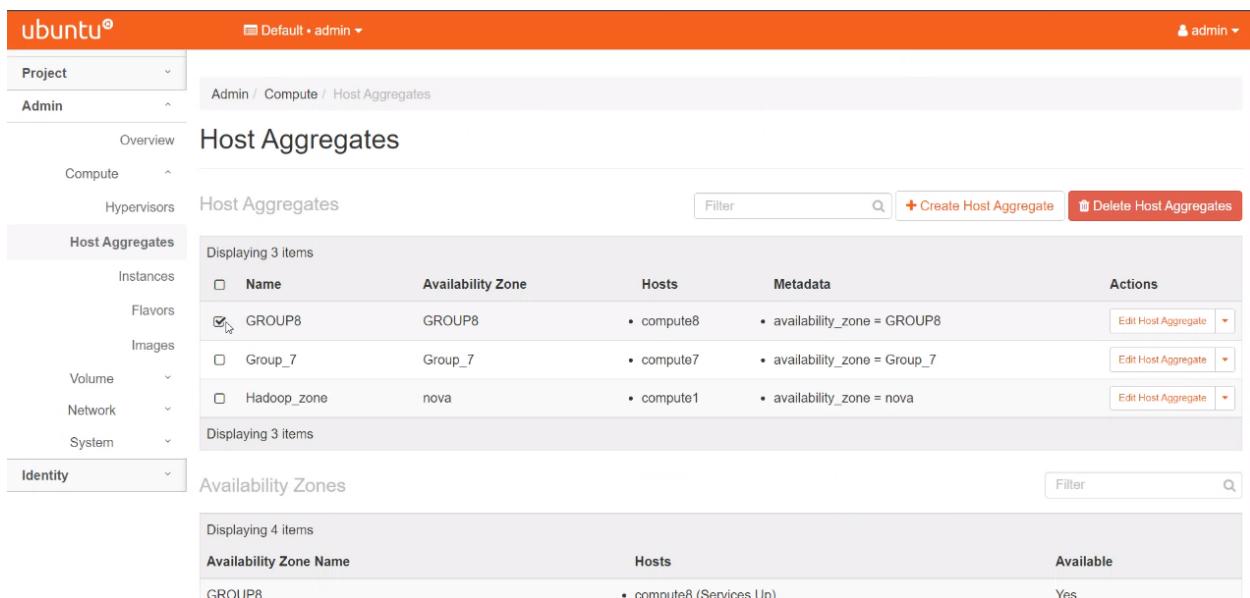
Step 1:

Creating zone: In the Admin tab; then in the compute section, select host aggregation then create a zone by naming 'GROUP8' and add compute8 to this one.



The screenshot shows the 'Create Host Aggregate' dialog box in the OpenStack Horizon interface. The 'Name' field is filled with 'GROUP8'. The 'Availability Zone' dropdown is set to 'GROUP8'. Below the dialog, the 'Host Aggregates' table shows three items: GROUP8, Group_7, and Hadoop_zone. The 'Availability Zones' table shows one item: GROUP8, which has one host, compute8, listed under it. The 'Available' column for GROUP8 is marked as 'Yes'.

The result like as below:

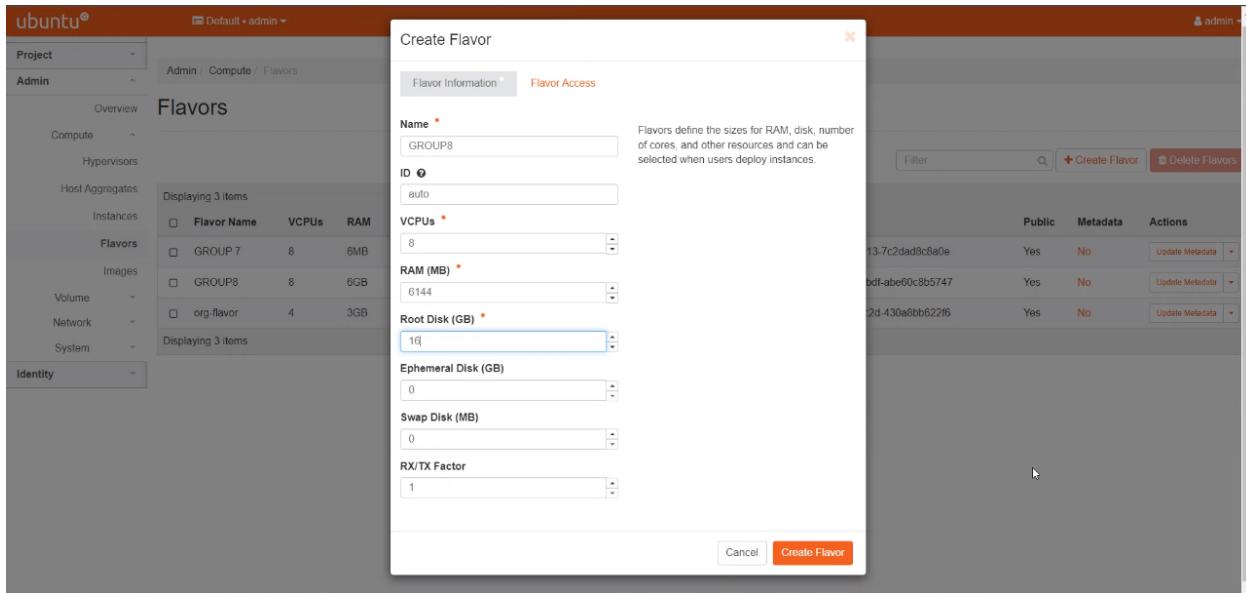


The screenshot shows the 'Host Aggregates' page in the OpenStack Horizon interface. The 'GROUP8' aggregate is selected, indicated by a checked checkbox in the 'Name' column. The table shows three host aggregates: GROUP8, Group_7, and Hadoop_zone. The 'Availability Zones' table shows one item: GROUP8, which has one host, compute8, listed under it. The 'Available' column for GROUP8 is marked as 'Yes'.

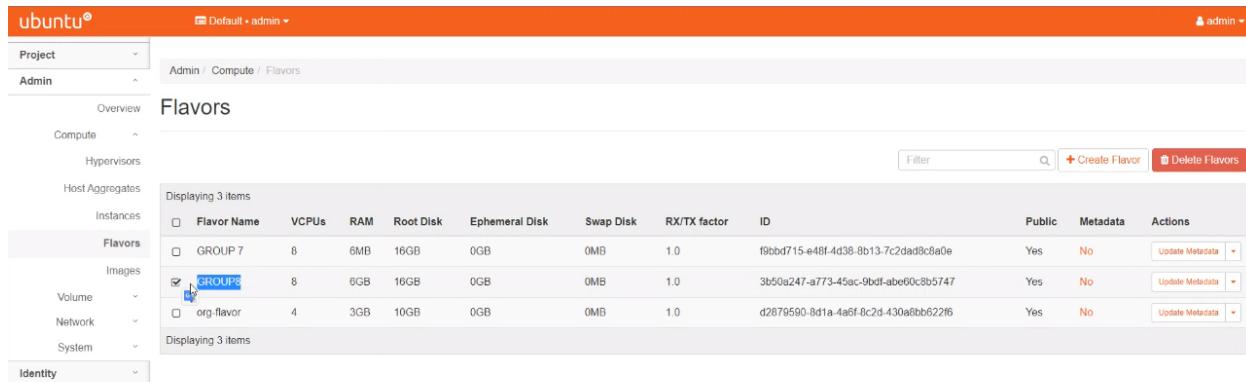
3

Step 2:

Creating flavor: In the Admin tab; then in the compute section, select flavors then create a new flavor by naming 'GROUP8' with 6 cores of VCPU, 8Gb of RAM and 16Gb of disk.



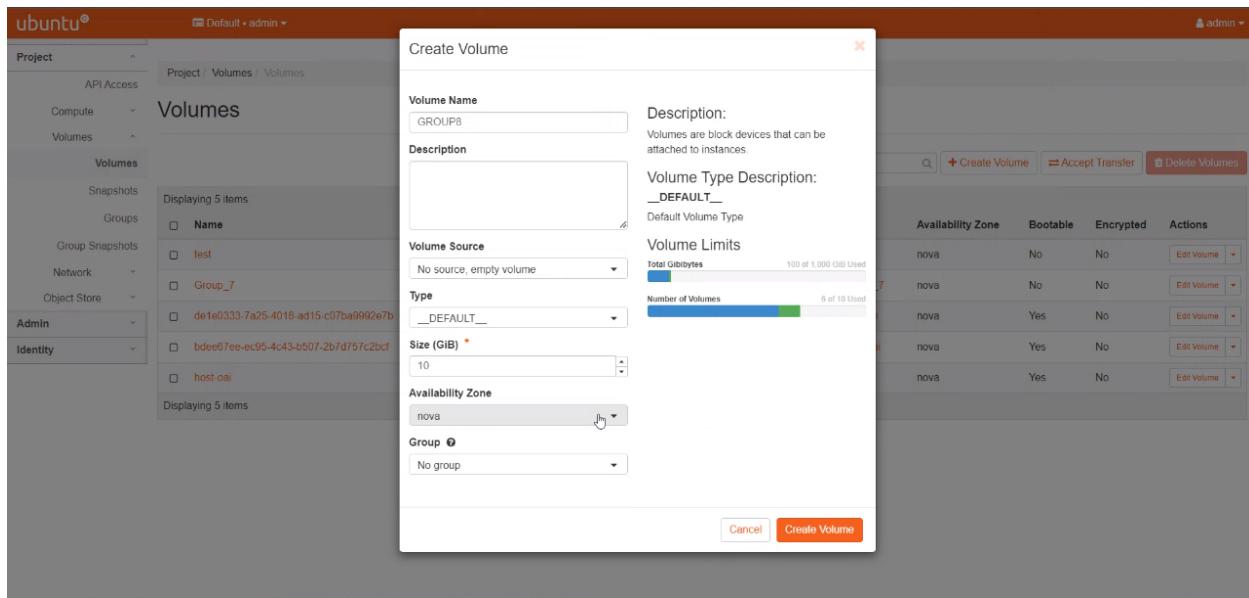
The result like as below:



4

Step 3:

Create volume: In the project tab; then in the volume section, then create a new volume by naming 'GROUP8' with size 10 GB.



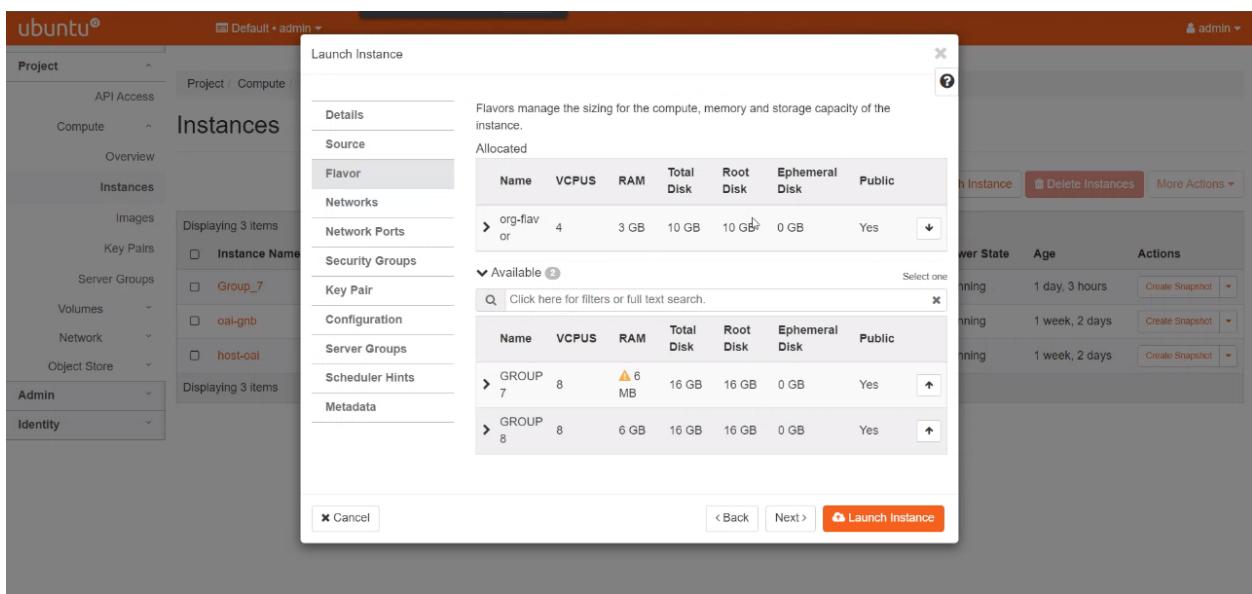
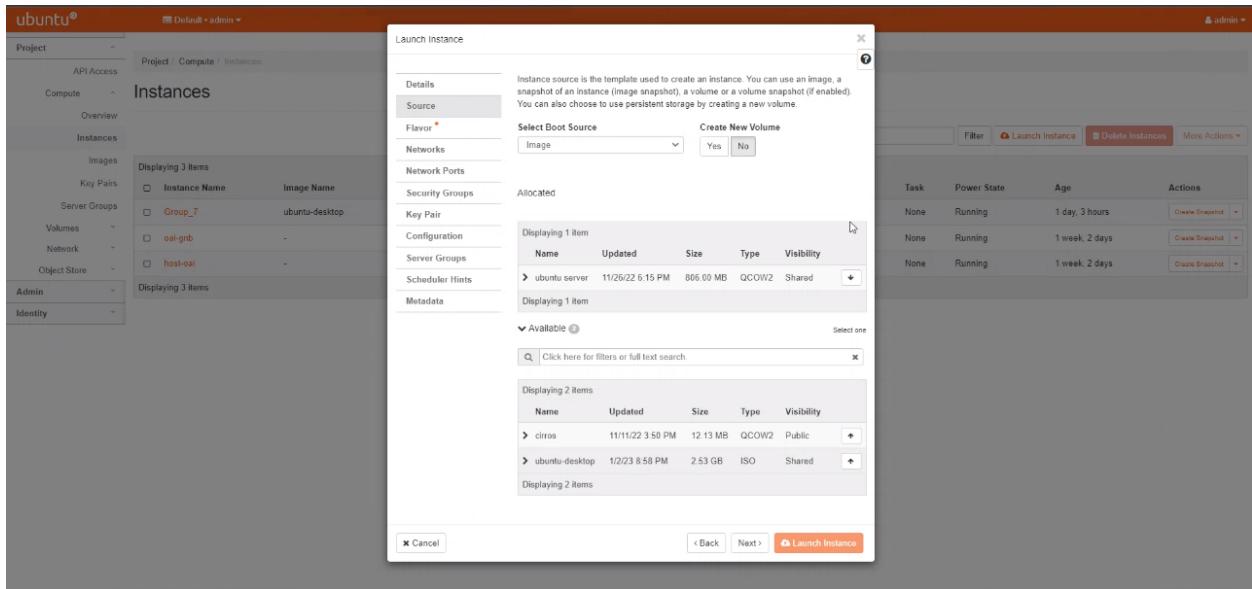
The result like as below:

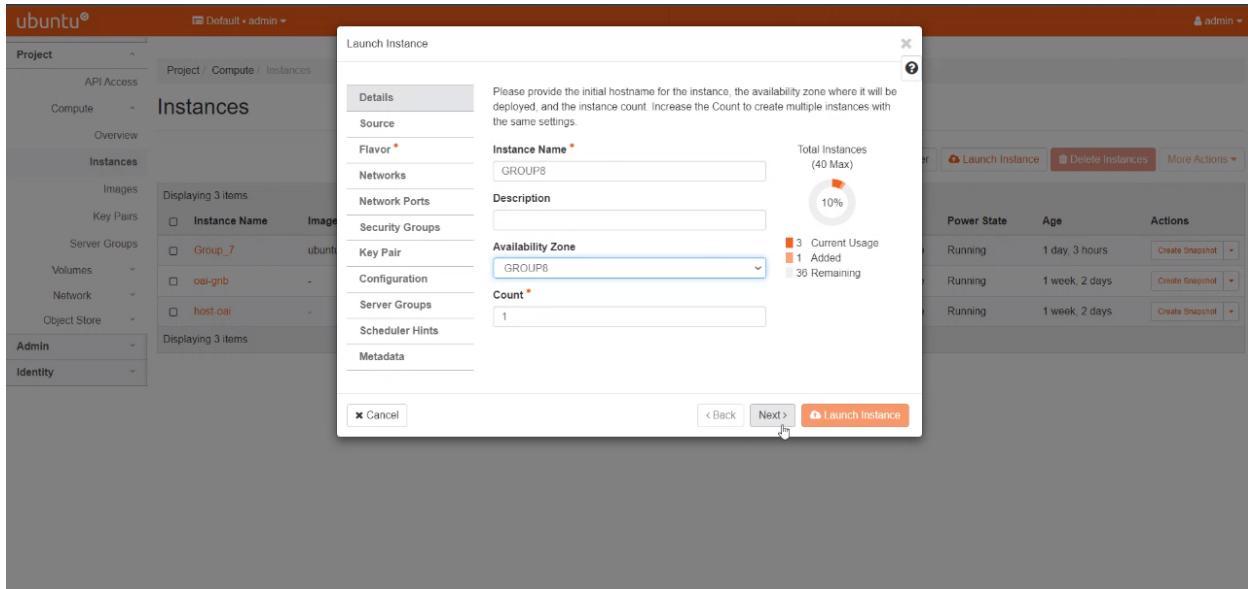
Volume	Name	Description	Size	Status	Group	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
	GROUP8	-	10GiB	Available	-	_DEFAULT_		nova	No	No	<button>Edit Volume</button>
	test	-	10GiB	Available	-	_DEFAULT_		nova	No	No	<button>Edit Volume</button>
	Group_7	-	10GiB	In-use	-	_DEFAULT_	/dev/vda on Group_7	nova	No	No	<button>Edit Volume</button>
	de1e0333-7a25-4018-ad15-c07ba9992e7b	-	20GiB	In-use	-	_DEFAULT_	/dev/vda on oai-gnb	nova	Yes	No	<button>Edit Volume</button>
	bdee67ee-ec95-4c43-b507-2b7d757c2bcf	-	20GiB	In-use	-	_DEFAULT_	/dev/vda on host-oai	nova	Yes	No	<button>Edit Volume</button>
	host-oai	-	20GiB	Available	-	_DEFAULT_		nova	Yes	No	<button>Edit Volume</button>

5

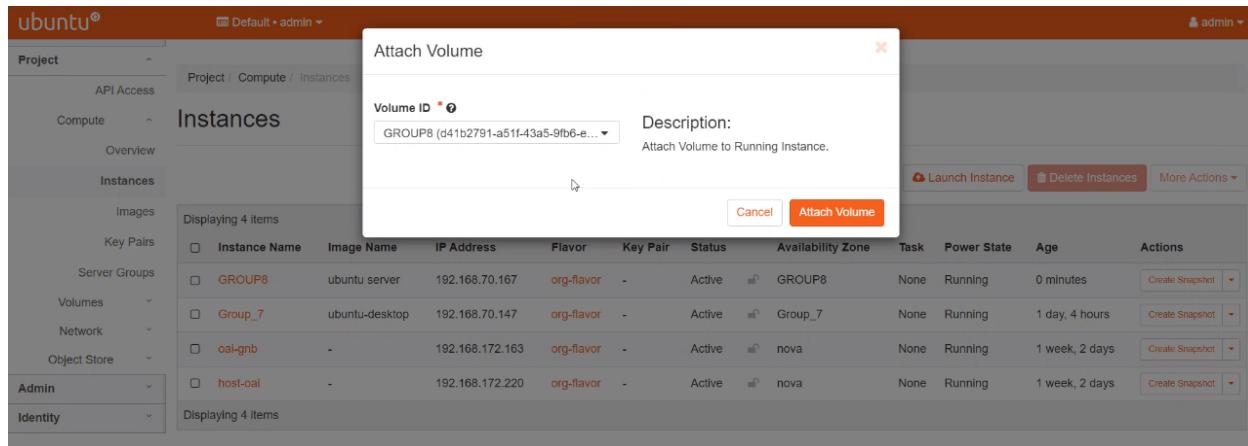
Step 4:

Create Instance: In the project tab; then in the compute section, select instances and press launch instance; fill it like below configs:

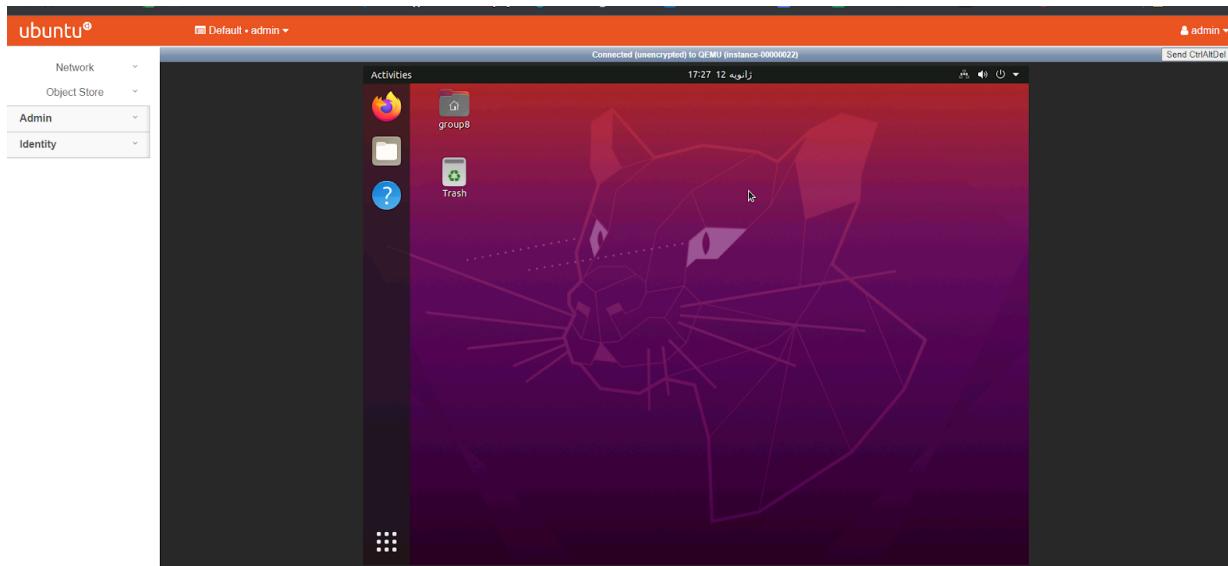




Then attach volume to the created instance:



Then installing Ubuntu...



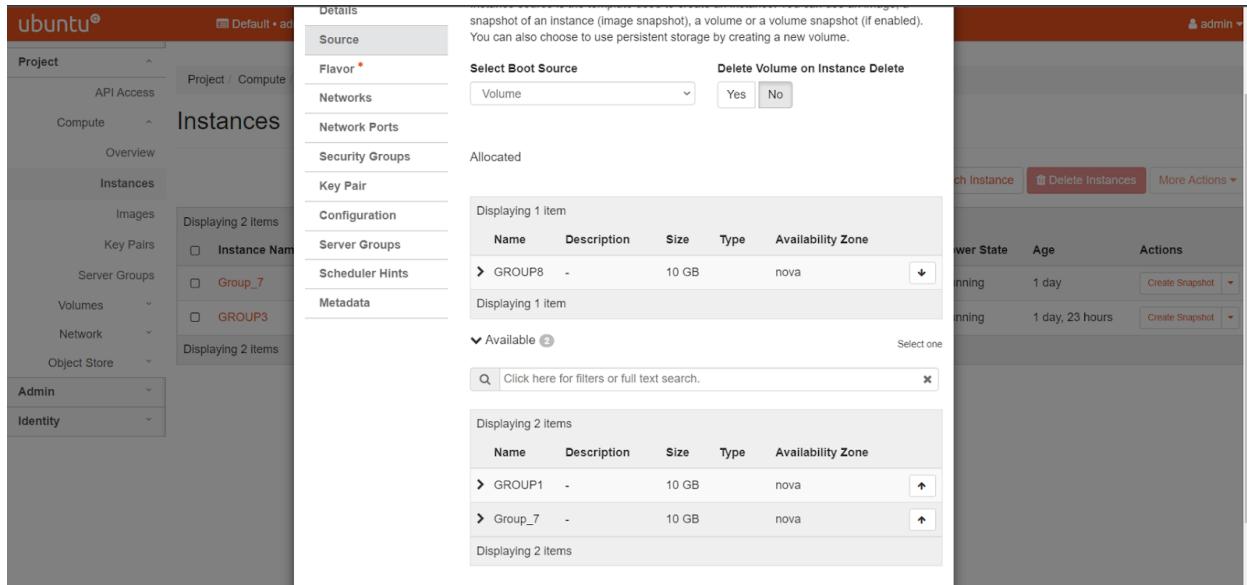
Step 5:

Detach volume:

The screenshot shows the OpenShift web console under the "Instances" tab. A modal dialog box titled "Detach Volume" is open, prompting for a "Volume ID" (set to "GROUP8 (043a1db3-8c24-46da-a9e9...)") and a "Description" ("Detach Volume from Running Instance"). The main table lists four instances: "Group_7_fromVol", "GROUP1", "GROUP3", and "GROUP8", each with its status, IP address, and other details.

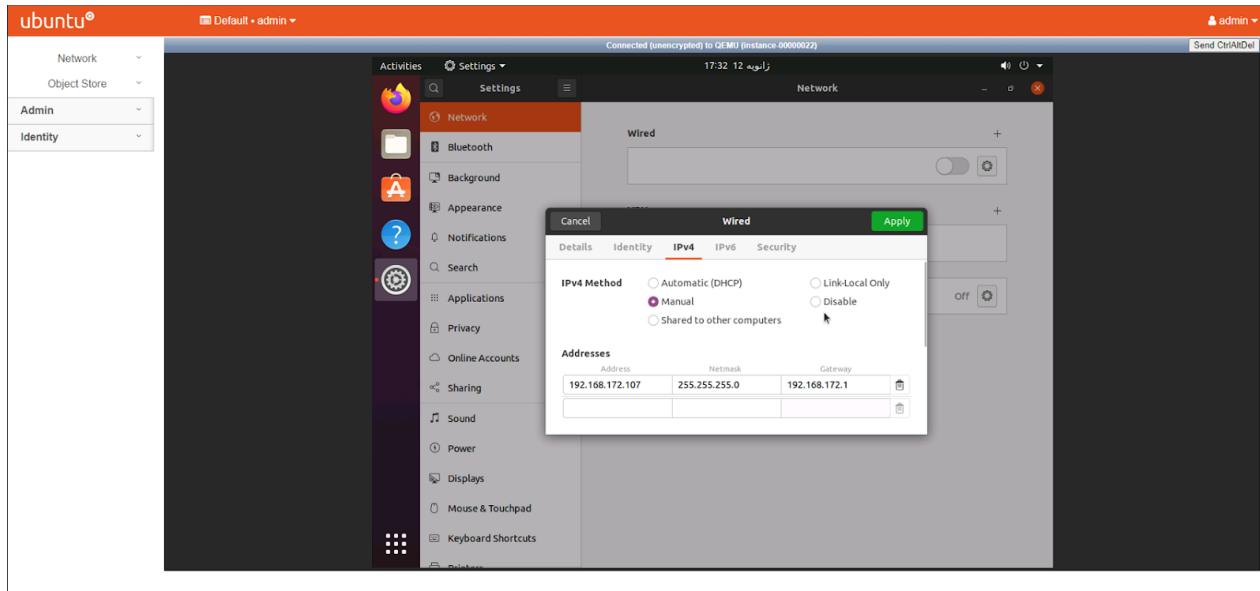
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
Group_7_fromVol	-	192.168.70.176	org-flavor	-	Active	Group_7	None	Running	9 minutes	<button>Create Snapshot</button>
GROUP1	ubuntu-desktop	192.168.70.187	org-flavor	-	Active	GROUP1	None	Running	21 hours, 59 minutes	<button>Create Snapshot</button>
GROUP3	-	192.168.70.146	org-flavor	-	Active	GROUP3	None	Running	3 days, 23 hours	<button>Create Snapshot</button>
GROUP8	-	192.168.70.161	org-flavor	-	Active	GROUP8	None	Running	1 week	<button>Create Snapshot</button>

Create new instance:



Step 5.5:

A) Change network setting and IP:



B) we wait for other team but no team create IP access which our team can ping that .

C) Snapshot has been created:

	Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
<input type="checkbox"/>	> admin	5g	Snapshot	Active	Private	No	QCOW2	0 bytes
<input type="checkbox"/>	> admin	GROUP8	Snapshot	Active	Private	No	QCOW2	0 bytes
<input type="checkbox"/>	> admin	ubuntu server	Image	Active	Shared	No	QCOW2	806.00 MB
<input type="checkbox"/>	> admin	ubuntu-desktop	Image	Active	Shared	No	ISO	2.53 GB

Step 6:

First install requirement Dictionaries:

```
● ● ●
pip install wheel
pip install openstacksdk
python -m openstack version
```

After that download and change 'cloud.yml' file as we send it for you.

And also try to code for checking different layers as we send it for you by name 'main.py'.

output is like this:

```
whitch option?  
1) Proxy layer  
2) Cloud layer  
3) resource layer  
>> |
```

Proxy layer:

```
● ● ●  
  
def proxy_layer():  
    openstack.enable_logging(debug=True)  
    conn = openstack.connect(cloud='openstack')  
    for server in conn.compute.servers():  
        print(server.to_dict())
```

Output:

The output like the picture in next page 😊

```
>> 1
{'access_ipv4': '',
 'access_ipv6': '',
 'addresses': {'net1': [{('OS-EXT-IPS-MAC:mac_addr': 'fa:16:3e:28:78:76',
                  'OS-EXT-IPS:type': 'fixed',
                  'addr': '192.168.70.176',
                  'version': 4)}]},
 'admin_password': None,
 'attached_volumes': [{('attachment_id': None,
                      'bdm_id': None,
                      'delete_on_termination': False,
                      'device': None,
                      'id': 'af690e4f-bdb0-42e4-859f-78e5ebdca6ce',
                      'location': None,
                      'name': None,
                      'tag': None,
                      'volume_id': None})],
 'availability_zone': 'Group_7',
 'block_device_mapping': None,
 'compute_host': 'compute7',
 'config_drive': '',
 'created_at': '2023-01-19T14:50:44Z',
 'description': None,
 'disk_config': 'AUTO',
 'fault': None,
 'flavor': {'description': None,
            'disk': 10,
            'ephemeral': 0,
            'extra_specs': {},
            'id': 'org-flavor',
            'is_disabled': None,
            'is_public': True,
            'location': None,
            'name': 'org-flavor',
            'original_name': 'org-flavor',
            'ram': 3072,
            'rxtx_factor': None,
            'swap': 0,
            'vcpus': 4},
 'flavor_id': None,
 'has_config_drive': '',
 'host_id': 'f89b26e5998fac2fd3a9323118e74ed555b0f2745f78760be3b5e377',
 'host_status': 'UP',
 'hostname': 'group-7-fromvol',
 'hypervisor_hostname': 'compute7',
 'id': 'aec8a260-78e3-4d6b-9769-2bfc597edf00',
 'image': {'architecture': None,
           'checksum': None,
           'container_format': None,
           'created_at': None,
           'direct_url': None,
           'disk_format': None,
           'file': None,
           'has_auto_disk_config': None,
           'hash_algo': None,
           'hash_value': None,
           'hw_cpu_cores': None,
           'hw_cpu_policy': None,
           'hw_cpu_sockets': None,
           'hw_cpu_thread_policy': None,
           'hw_cpu_threads': None,
           'hw_disk_bus': None,
           'hw_machine_type': None,
           'hw_qemu_guest_agent': None},
```

Cloud layer:

```
def cloud_layer():

    openstack.enable_logging(debug=True)
    conn = openstack.connect(cloud='openstack')
    for server in conn.list_servers():
        print(server.to_dict())
```

Output:

The output like the picture in next page 😊

```
d be attempting to list floating ips on neutron, it is possible to control the b  
is setting, you will need a clouds.yaml file. For more information, please see h  
{'access_ipv4': '192.168.70.176',  
 'access_ipv6': '',  
 'addresses': {'net1': [{['OS-EXT-IPS-MAC:mac_addr': 'fa:16:3e:28:78:76',  
   'OS-EXT-IPS:type': 'fixed',  
   'addr': '192.168.70.176',  
   'version': 4}]},  
 'admin_password': None,  
 'attached_volumes': [{['attachment_id': None,  
   'bdm_id': None,  
   'delete_on_termination': False,  
   'device': None,  
   'id': 'af690e4f-bdb0-42e4-859f-78e5ebdca6ce',  
   'location': None,  
   'name': None,  
   'tag': None,  
   'volume_id': None}],  
 'availability_zone': 'Group_7',  
 'block_device_mapping': None,  
 'compute_host': 'compute7',  
 'config_drive': '',  
 'created_at': '2023-01-19T14:50:44Z',  
 'description': None,  
 'disk_config': 'AUTO',  
 'fault': None,  
 'flavor': {'['description': None,  
   'disk': 10,  
   'ephemeral': 0,  
   'extra_specs': {},  
   'id': 'org-flavor',  
   'is_disabled': None,  
   'is_public': True,  
   'location': None,  
   'name': 'org-flavor',  
   'original_name': 'org-flavor',  
   'ram': 3072,  
   'rxtx_factor': None,  
   'swap': 0,  
   'vcpus': 4},  
 'flavor_id': None,  
 'has_config_drive': '',  
 'host_id': 'f89b26e5998fac2fd3a9323118e74ed555b0f2745f78760be3b5e377',  
 'host_status': 'UP',  
 'hostname': 'group-7-fromvol',  
 'hypervisor_hostname': 'compute7',  
 'id': 'aec8a260-78e3-4d6b-9769-2bfc597edf00',  
 'image': {'['architecture': None,  
   'checksum': None,  
   'container_format': None,  
   'created_at': None,  
   'direct_url': None,  
   'disk_format': None,  
   'file': None,  
   'has_auto_disk_config': None,  
   'hash_algo': None,  
   'hash_value': None,  
   'hw_cpu_cores': None,  
   'hw_cpu_policy': None,  
   'hw_cpu_sockets': None,  
   'hw_cpu_thread_policy': None,  
   'hw_cpu_threads': None,  
   'hw_disk_bus': None,  
   'hw_machine_type': None,
```

Resource layer:

```
● ● ●

def resource_layer():

    openstack.enable_logging(debug=True)
    conn = openstack.connect(cloud='openstack')
    for server in
openstack.compute.v2.server.Server.list(session=conn.compute):
    print(server.to_dict())
```

Output:

The output like the picture in next page 😊

```
3) resource layer
>> 3
{'access_ipv4': None,
 'access_ipv6': None,
 'addresses': None,
 'admin_password': None,
 'attached_volumes': [],
 'availability_zone': None,
 'block_device_mapping': None,
 'compute_host': None,
 'config_drive': None,
 'created_at': None,
 'description': None,
 'disk_config': None,
 'fault': None,
 'flavor': None,
 'flavor_id': None,
 'has_config_drive': None,
 'host_id': None,
 'host_status': None,
 'hostname': None,
 'hypervisor_hostname': None,
 'id': 'aec8a260-78e3-4d6b-9769-2bfc597edf00',
 'image': None,
 'image_id': None,
 'instance_name': None,
 'interface_ip': '',
 'is_locked': None,
 'kernel_id': None,
 'key_name': None,
 'launch_index': None,
 'launched_at': None,
 'links': [{"href": "http://controller:8774/v2.1/servers/aec8a260-78e3-4d6b-9769-2bfc597edf00",
            'rel': 'self'},
           {"href": "http://controller:8774/servers/aec8a260-78e3-4d6b-9769-2bfc597edf00",
            'rel': 'bookmark'}],
 'location': {'cloud': 'openstack',
              'project': {'domain_id': None,
                          'domain_name': None,
                          'id': '68100fa976c643a3bf4317ff983af65b',
                          'name': 'admin'},
              'region_name': 'RegionOne',
              'zone': None},
 'max_count': None,
 'metadata': None,
 'min_count': None,
 'name': 'Group_7_fromVol',
 'networks': None,
 'power_state': None,
 'private_v4': '',
 'private_v6': '',
 'progress': None,
 'project_id': None,
 'public_v4': '',
 'public_v6': '',
 'ramdisk_id': None,
 'reservation_id': None,
 'root_device_name': None,
 'scheduler_hints': None,
 'security_groups': None,
 'server_groups': None,
 'status': None,
 'tags': [],
 'task_state': None},
```

Compare different layers type:

Proxy layer: This layer is mostly used to provide information about all the compute servers such as volumes, compute flavor, images detail on that compute node, hostnames and network setting for all openStack compute servers.

Cloud layer: This is a higher level access to the openstack SDK where we can check the list of connection servers that have been set up for our openstack SDK project in "cloud.yml".

This one shows more details about different configurations on specific servers about different openstack modules such as Network module e.g Neutron.

Resource layer: In this layer we follow more details of low-level layers such as CRUD (create, read, update and delete) operations, as well as information about the main building blocks of other layers.

In this layer we have less info about some network configs and flavors. And most fields fill up by 'None' value.

Step 7:

NOTE: The code is placed next to the document file 'create_cluster.py'.