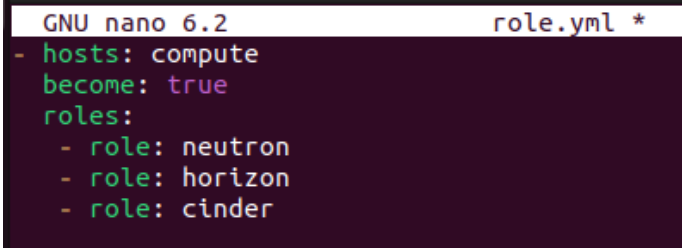


<b>Name:</b> Andaya, Lyka C.	<b>Date Performed:</b> December 5, 2023
<b>Course/Section:</b> CPE31S4	<b>Date Submitted:</b> December 8, 2023
<b>Instructor:</b> Dr. Taylar	<b>Semester and SY:</b> 2023-2024
<b>Activity 15: OpenStack Installation (Neutron, Horizon, Cinder)</b>	
<b>1. Objectives</b>	
Create a workflow to install OpenStack using Ansible as your Infrastructure as Code (IaC).	
<b>2. Intended Learning Outcomes</b>	
<ol style="list-style-type: none"> <li>1. Analyze the advantages and disadvantages of cloud services</li> <li>2. Evaluate different Cloud deployment and service models</li> <li>3. Create a workflow to install and configure OpenStack base services using Ansible as documentation and execution.</li> </ol>	
<b>3. Resources</b>	
<p>Oracle VirtualBox (Hypervisor)</p> <p>1x Ubuntu VM or Centos VM</p>	
<b>4. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a new repository for this activity.</li> <li>2. Create a playbook that converts the steps in the following items in <a href="https://docs.openstack.org/install-guide/">https://docs.openstack.org/install-guide/</a> <ol style="list-style-type: none"> <li>a. Neutron</li> <li>b. Horizon</li> <li>c. Cinder</li> <li>d. Create different plays in installing per server type (controller, compute etc.) and identify it as a group in the Inventory file.</li> <li>e. Add, commit and push it to your GitHub repo.</li> </ol> </li> </ol>	
<b>5. Output</b> (screenshots and explanations)	
<b>INPUT</b>	 <pre> GNU nano 6.2                                role.yml * - hosts: compute   become: true   roles:     - role: neutron     - role: horizon     - role: cinder </pre>

```

GNU nano 6.2                                main.yml *
- name: installing the components for neutron
  apt:
    name: neutron-linuxbridge-agent
    when: ansible_distribution == "Ubuntu"

- name: configuring RabbitMQ message queue access
  copy:
    dest: /etc/neutron/neutron.conf
    content: |
      [DEFAULT]
      transport_url = rabbit://openstack:1234@controller

- name: configuring Identity service access (1)
  copy:
    dest: /etc/neutron/neutron.conf
    content: |
      [DEFAULT]
      auth_strategy = keystone

      [keystone_authtoken]
      www_authenticate_uri = http://controller:5000
      auth_url = http://controller:5000
      memcached_servers = controller:11211
      auth_type = password
      project_domain_name = default
      user_domain_name = default
      project_name = service
      username = neutron
      password = 1234

- name: configuring the lock path
  copy:
    dest: /etc/neutron/neutron.conf
    content: |
      [oslo_currency]
      lock_path = /var/lib/neutron/tmp

```

```

- name: configuring the lock path
  copy:
    dest: /etc/neutron/neutron.conf
    content: |
      [oslo_currency]
      lock_path = /var/lib/neutron/tmp

- name: configuring the access parameters
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [neutron]
      auth_url = http://controller:5000
      auth_type = password
      project_domain_name = default
      user_domain_name = default
      region_name = RegionOne
      project_name = service
      username = neutron
      password = 1234

- name: restarting the compute service
  shell: service nova-compute restart
  when: ansible_distribution == "Ubuntu"

- name: restarting the linux bridge agent
  shell: service neutron-linuxbridge-agent restart
  when: ansible_distribution == "Ubuntu"

```

```

GNU nano 6.2 main.yml *
- name: Installing the packages for horizon
  apt:
    name: openstack-dashboard
    when: ansible_distribution == "Ubuntu"

- name: configuring the dashboard to use Openstack services
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_HOST = "controller"
    when: ansible_distribution == "Ubuntu"

- name: Allowing all hosts to access dashboard
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      ALLOWED_HOSTS = ['*']
    when: ansible_distribution == "Ubuntu"

- name: configure the memcached session storage service
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

      CACHES = {
        'default': {
          'BACKEND': 'django.core.cache.backends.memcached.>
        }
      }
    when: ansible_distribution == "Ubuntu"

- name: enable the identity API version 3
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTA>
    when: ansible_distribution == "Ubuntu"

- name: enable support for domains
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
    when: ansible_distribution == "Ubuntu"

- name: configure API versions
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_API_VERSIONS = {
        "identity": 3,
        "image": 2,
        "volume": 3,
      }
    when: ansible_distribution == "Ubuntu"

- name: configure default as the default domain for users that y>
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
    when: ansible_distribution == "Ubuntu"

```

```

- name: configure user as the default role for users that your c>
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
  when: ansible_distribution == "Ubuntu"

- name: if you choose network 1, disable support for layer-3 net>
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      OPENSTACK_NEUTRON_NETWORK = {
        'enable_router': False,
        'enable_quotas': False,
        'enable_ipv6': False,
        'enable_distributed_router': False,
        'enable_ha_router': False,
        'enable_fip_topology_check': False
      }
  when: ansible_distribution == "Ubuntu"

- name: add the following line if not added yet
  copy:
    dest: /etc/openstack-dashboard/local_settings.py
    content: |
      CACHES = {
        'default': {
          'BACKEND': 'django.core.cache.backends.memcached.M>
        },
      }
      SESSION_ENGINE = "django.contrib.sessions.backends.cache"
      OPENSTACK_HOST = "192.168.56.113"
      OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTA>

      TIME_ZONE = "Asia/Tokyo"

```

```

      TIME_ZONE = "Asia/Tokyo"

      OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
      OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
      OPENSTACK_API_VERSIONS = {
        "identity": 3,
        "volume": 3,
        "compute": 2,
      }
      WSGIApplicationGroup %{GLOBAL}
  when: ansible_distribution == "Ubuntu"

  name: install apache2
  apt:
    name: apache2
  when: ansible_distribution == "Ubuntu"

  name: finalize installation by starting the apache2 service
  shell: sudo systemctl reload apache2.service
  when: ansible_distribution == "Ubuntu"

  name: finalize installation by reloading the apache2 service
  shell: sudo systemctl reload apache2.service
  when: ansible_distribution == "Ubuntu"

```

```

GNU nano 6.2 main.yml *
- name: Install Cinder
  apt:
    name: cinder-api
    when: ansible_distribution == "Ubuntu"

- name: install cinder scheduler
  shell: sudo apt install cinder-scheduler
  when: ansible_distribution == "Ubuntu"

- name: configure database access
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [database]
      connection = mysql+pymysql://cinder:1234@controller/cinder
  when: ansible_distribution == "Ubuntu"

- name: configure RabbitMQ message queue access
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
      transport_url = rabbit://openstack:1234@controller
  when: ansible_distribution == "Ubuntu"

- name: configure identity services access
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
      auth_strategy = keystone

      [keystone_authtoken]
      www_authenticate_uri = http://controller:5000
      auth_url = http://controller:5000
      memcached_servers = controller:11211

```

```

GNU nano 6.2 main.yml *
www_authenticate_uri = http://controller:5000
auth_url = http://controller:5000
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = 1234
when: ansible_distribution == "Ubuntu"

- name: configure my_ip option to use the management interface IP add of controller node
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
      my_ip = 192.168.56.106
  when: ansible_distribution == "Ubuntu"

- name: configure the lock path
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [oslo_concurrency]
      lock_path = /var/lib/cinder/tmp
  when: ansible_distribution == "Ubuntu"

- name: populate the block storage database
  shell: su -s /bin/sh -c "cinder-manage db sync" cinder
  when: ansible_distribution == "Ubuntu"

- name: adding the following line for block storage
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [cinder]

```

```

GNU nano 6.2 main.yml *
- name: adding the following line for block storage
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [cinder]
        os_region_name = RegionOne
  when: ansible_distribution == "Ubuntu"

- name: install nova-api
  shell: sudo apt install nova-api
  when: ansible_distribution == "Ubuntu"

- name: restarting the compute API service
  shell: service nova-api start
  when: ansible_distribution == "Ubuntu"

- name: restart the block storage services (1)
  shell: service cinder-scheduler start
  when: ansible_distribution == "Ubuntu"

- name: restart the block storage services (2)
  shell: sudo systemctl start apache2
  when: ansible_distribution == "Ubuntu"

- name: install the supporting utility packages
  apt:
    name:
      - lvm2
      - thin-provisioning-tools
  when: ansible_distribution == "Ubuntu"

- name: create the LVM volume group cinder-volume
  shell: sudo touch cinder-volumes /dev/sdb
  when: ansible_distribution == "Ubuntu"

- name: install the packages for cinder (2) storage node

- name: install the packages for cinder (2) storage node
  apt:
    name:
      - cinder-volume
      - tgt
  when: ansible_distribution == "Ubuntu"

- name: configure the LVM backend with the LVM driver
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [lvm]
        volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
        volume_group = cinder-volumes
        target_protocol = iscsi
        target_helper = tgtadm
  when: ansible_distribution == "Ubuntu"

- name: enabling the LVM backend
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
        enabled_backends = lvm
  when: ansible_distribution == "Ubuntu"

- name: configuring the location of the image service API
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
        glance_api_servers = http://controller:9292
  when: ansible_distribution == "Ubuntu"

```

```

- name: configuring the lock path
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [oslo_concurrency]
      lock_path = /var/lib/cinder/tmp
  when: ansible_distribution == "Ubuntu"

- name: Restarting the block storage volume service including its dependencies (1)
  shell: service tgt restart
  when: ansible_distribution == "Ubuntu"

- name: Restarting the block storage volume service including its dependencies (2)
  shell: service cinder-volume restart
  when: ansible_distribution == "Ubuntu"

- name: install the packages for cinder (3) backup service
  apt:
    name: cinder-backup
  when: ansible_distribution == "Ubuntu"

- name: configuring the backup options
  copy:
    dest: /etc/cinder/cinder.conf
    content: |
      [DEFAULT]
      backup_driver = cinder.backup.drivers.swift.SwiftBackupDriver
      backup_swift_url = SWIFT_URL
  when: ansible_distribution == "Ubuntu"

- name: restart the block storage backup service
  shell: service cinder-backup start
  when: ansible_distribution == "Ubuntu"

```

**Explanation:** The server 1 in linux will install the Neutron, Horizon, and Cinder

## PROCESS

```
lykaandaya@managenode:~/H0A15.1$ ansible-playbook --ask-become-pass role.yml
BECOME password:

PLAY [compute] *****
*****
**

TASK [Gathering Facts] *****
*****
**
ok: [192.168.56.113]

TASK [neutron : installing the components for neutron] *****
*****
**
ok: [192.168.56.113]

TASK [neutron : configuring RabbitMQ message queue access] *****
*****
**
changed: [192.168.56.113]

TASK [neutron : configuring Identity service access (1)] *****
*****
**
changed: [192.168.56.113]

TASK [neutron : configuring the lock path] *****
*****
**
changed: [192.168.56.113]

TASK [neutron : configuring the access parameters] *****
*****
**
ok: [192.168.56.113]

TASK [neutron : restarting the compute service] *****
```



```
TASK [neutron : restarting the compute service] *****
*****
**
changed: [192.168.56.113]

TASK [neutron : restarting the linux bridge agent] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : Installing the packages for horizon] *****
*****
**
ok: [192.168.56.113]

TASK [horizon : configuring the dashboard to use Openstack services] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : Allowing all hosts to access dashboard] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : configure the memcached session storage service] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : enable the identity API version 3] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : enable support for domains] *****
*****
```

```
TASK [horizon : enable support for domains] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : configure API versions] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : configure default as the default domain for users
that y>] *****
**
changed: [192.168.56.113]

TASK [horizon : configure user as the default role for users that
your c>] *****
**
changed: [192.168.56.113]

TASK [horizon : if you choose network 1, disable support for laye
r-3 net>] *****
**
changed: [192.168.56.113]

TASK [horizon : add the following line if not added yet] *****
*****
**
changed: [192.168.56.113]

TASK [horizon : install apache2] *****
*****
**
ok: [192.168.56.113]

TASK [horizon : finalize installation by starting the apache2 ser
vice] *****
**
```

```
TASK [horizon : finalize installation by starting the apache2 service] *****
**
changed: [192.168.56.113]

TASK [horizon : finalize installation by reloading the apache2 service] *****
**
changed: [192.168.56.113]

TASK [cinder : Install Cinder] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : install cinder scheduler] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : configure database access] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : configure RabbitMQ message queue access] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : configure identity services access] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : configure my_ip option to use the management interface IP add of controller node] *****
**
changed: [192.168.56.113]
```

```
TASK [cinder : configure the lock path] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : populate the block storage database] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : adding the following line for block storage] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : install nova-api] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : restarting the compute API service] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : restart the block storage services (1)] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : restart the block storage services (2)] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : install the supporting utility packages] *****
*****
**
ok: [192.168.56.113]
```

```
TASK [cinder : create the LVM volume group cinder-volume] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : install the packages for cinder (2) storage node]
*****
**
changed: [192.168.56.113]

TASK [cinder : configure the LVM backend with the LVM driver] ***
*****
**
changed: [192.168.56.113]

TASK [cinder : enabling the LVM backend] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : configuring the location of the image service API]
*****
**
changed: [192.168.56.113]

TASK [cinder : configuring the lock path] *****
*****
**
changed: [192.168.56.113]

TASK [cinder : Restarting the block storage volume service includ
ing its dependencies (1)] *****
**
changed: [192.168.56.113]

TASK [cinder : Restarting the block storage volume service includ
ing its dependencies (2)] *****
**
changed: [192.168.56.113]
```

	<pre> TASK [cinder : install the packages for cinder (3) backup service ] ***** ** changed: [192.168.56.113]  TASK [cinder : configuring the backup options] ***** ***** ** changed: [192.168.56.113]  TASK [cinder : restart the block storage backup service] ***** ***** ** changed: [192.168.56.113]  PLAY RECAP ***** ***** ** 192.168.56.113      : ok=47  changed=41  unreachable=0 failed=0    skipped=0    rescued=0    ignored=0 </pre>
<b>Explanation:</b> It shows that it executed the tasks that I created in the playbook	
<b>OUTPUT</b>	
<b>UBUNTU</b>	<pre> lykaandaya@controlnode2:~\$ sudo systemctl status neutron-linuxbridge-agent ● neutron-linuxbridge-agent.service - Openstack Neutron Linux Bridge Agent    Loaded: loaded (/lib/systemd/system/neutron-linuxbridge-agent.service; enabled; vendor preset: enabled)    Active: active (running) since Tue 2023-12-05 05:49:17 PST; 744ms ago      Process: 115156 ExecStartPre=/bin/mkdir -p /var/lock/neutron /var/log/neutron (code=exited, status=0/SUCCESS)      Process: 115157 ExecStartPre=/bin/chown neutron:neutron /var/lock/neutron /var/log/neutron (code=exited, status=0/SUCCESS)      Process: 115158 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS)     Main PID: 115160 (neutron-linuxbr)       Tasks: 1 (limit: 1131)      Memory: 18.3M         CPU: 201ms     CGroup: /system.slice/neutron-linuxbridge-agent.service             └─115160 /usr/bin/python3 /usr/bin/neutron-linuxbridge-agent --config-file=/etc/neutron/neutron.conf  Dec 05 05:49:17 controlnode2 systemd[1]: Starting Openstack Neutron Linux Bridge Agent: neutron-linuxbridge-agent.service. Dec 05 05:49:17 controlnode2 systemd[1]: Started Openstack Neutron Linux Bridge Agent: neutron-linuxbridge-agent.service. lines 1-15/15 (END) </pre>

```

lykaandaya@controlnode2:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-12-08 23:31:28 PST; 1min 1s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 1080 (apache2)
      Tasks: 85 (limit: 1131)
    Memory: 9.0M
       CPU: 1.062s
    CGroup: /system.slice/apache2.service
            └─1080 /usr/sbin/apache2 -k start
               └─1129 "(wsgi:cinder-wsgi" -k start
                  └─1130 "(wsgi:cinder-wsgi" -k start
                     └─1137 "(wsgi:cinder-wsgi" -k start
                        └─1138 "(wsgi:cinder-wsgi" -k start
                           └─1139 "(wsgi:cinder-wsgi" -k start
                              └─1149 "(wsgi:horizon) " -k start
                                 └─1150 "(wsgi:horizon) " -k start
                                    └─1151 "(wsgi:horizon) " -k start
                                       └─1152 "(wsgi:keystone-pu" -k start
                                          └─1153 "(wsgi:keystone-pu" -k start
                                             └─1154 "(wsgi:keystone-pu" -k start
                                                └─1155 "(wsgi:keystone-pu" -k start
                                                   └─1156 "(wsgi:keystone-pu" -k start
                                                      └─1157 /usr/sbin/apache2 -k start
                                                         └─1158 /usr/sbin/apache2 -k start
                                                            └─1159 /usr/sbin/apache2 -k start
                                                               └─1160 /usr/sbin/apache2 -k start
                                                                  └─1161 /usr/sbin/apache2 -k start

Dec 08 23:31:27 controlnode2 systemd[1]: Starting The Apache HTTP Server:
Dec 08 23:31:28 controlnode2 apachectl[930]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 10.0.0.1 as fallback
Dec 08 23:31:28 controlnode2 systemd[1]: Started The Apache HTTP Server:
lines 1-32/32 (END)

```

```

lykaandaya@controlnode2:~$ sudo systemctl status cinder-volume
● cinder-volume.service - OpenStack Cinder Volume
   Loaded: loaded (/lib/systemd/system/cinder-volume.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-12-05 05:55:12 PST; 951ms ago
     Docs: man:cinder-volume(1)
    Main PID: 118545 (cinder-volume)
      Tasks: 1 (limit: 1131)
    Memory: 29.5M
       CPU: 280ms
    CGroup: /system.slice/cinder-volume.service
            └─118545 /usr/bin/python3 /usr/bin/cinder-volume --config-file=/etc/cinder/cinder.conf

Dec 05 05:55:12 controlnode2 systemd[1]: Stopped OpenStack Cinder Volume.
Dec 05 05:55:12 controlnode2 systemd[1]: cinder-volume.service: Consumed 1.329s CPU time.
Dec 05 05:55:12 controlnode2 systemd[1]: Started OpenStack Cinder Volume.
lines 1-14/14 (END)

```

**Explanation:** In the linux server 1 it shows that the apache2(HTTP, horizon), neutron and cinder are installed and the service is currently active and running.



```

lykaandaya@managenode:~/HOA15.1$ git add *
lykaandaya@managenode:~/HOA15.1$ git commit -m "HOA15"
[main 293fcc5] HOA15
27 files changed, 21130 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 inventory
create mode 100644 role.yml
create mode 100644 roles/cinder/cinder.conf
create mode 100644 roles/cinder/handlers/main.yml
create mode 100644 roles/cinder/handlers/tasks/configure.yml
create mode 100644 roles/cinder/handlers/tasks/install.yml
create mode 100644 roles/cinder/handlers/tasks/main.yml
create mode 100644 roles/cinder/handlers/tasks/prereq.yml
create mode 100644 roles/cinder/nova.conf
create mode 100644 roles/horizon/handlers/main.yml
create mode 100644 roles/horizon/handlers/tasks/configure.yml
create mode 100644 roles/horizon/handlers/tasks/install.yml
create mode 100644 roles/horizon/handlers/tasks/main.yml
create mode 100644 roles/horizon/local_settings
create mode 100644 roles/horizon/openstack-dashboard.conf
create mode 100644 roles/neutron/dhcp_agent.ini
create mode 100644 roles/neutron/linuxbridge_agent.ini
create mode 100644 roles/neutron/metadata_agent.ini
create mode 100644 roles/neutron/ml2_conf.ini
create mode 100644 roles/neutron/neutron.conf
create mode 100644 roles/neutron/nova.conf
create mode 100644 roles/neutron/tasks/configure.yml
create mode 100644 roles/neutron/tasks/install.yml
create mode 100644 roles/neutron/tasks/main.yml
create mode 100644 roles/neutron/tasks/network1.yml
create mode 100644 roles/neutron/tasks/prereq.yml
lykaandaya@managenode:~/HOA15.1$ git push
Enumerating objects: 38, done.
Counting objects: 100% (38/38), done.
Compressing objects: 100% (37/37), done.
Writing objects: 100% (37/37), 109.64 KiB | 445.00 KiB/s, done.
Total 37 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
To github.com:andayalyka/HOA15.1.git
 f0f9c45..293fcc5  main -> main

```

**Explanation:** It add push all the file created and has a commit message HOA15.

### Reflections:

Answer the following:

1. Describe Neutron, Horizon and Cinder services



- Neutron, Horizon, and Cinder play vital roles within OpenStack, a cloud computing platform that operates on open-source principles. Neutron functions as the networking service, delivering on-demand, scalable, and adaptable network resources to other OpenStack services, facilitating communication among various components. Horizon acts as the web-based dashboard interface, providing users with a graphical tool to oversee and control their cloud resources, streamlining intricate tasks for an improved user experience. Simultaneously, Cinder serves as the block storage service, empowering users to connect and manage scalable block storage devices for their virtual machines. This facilitates seamless integration and effective administration of storage resources in the OpenStack environment, accommodating diverse storage backends and configurations. Collectively, Neutron, Horizon, and Cinder contribute to the extensive functionality and user-friendly interface of the OpenStack cloud computing platform.

**Conclusions:**

In conclusion, implementing a workflow for OpenStack installation using Ansible as the Infrastructure as Code (IaC) solution presents a strategic method for cloud deployment. This approach leverages the robust cloud computing capabilities of OpenStack alongside Ansible's automation proficiency. With IaC, the deployment process becomes more efficient, uniform, and easily replicable. Ansible's role in orchestrating OpenStack installation not only boosts efficiency through task automation but also ensures a standardized and dependable infrastructure. This strategy facilitates smooth management, adaptability, and scalability for future needs. In essence, the adoption of Ansible for OpenStack deployment reflects a dedication to agility, efficiency, and the sustained manageability of cloud infrastructure.