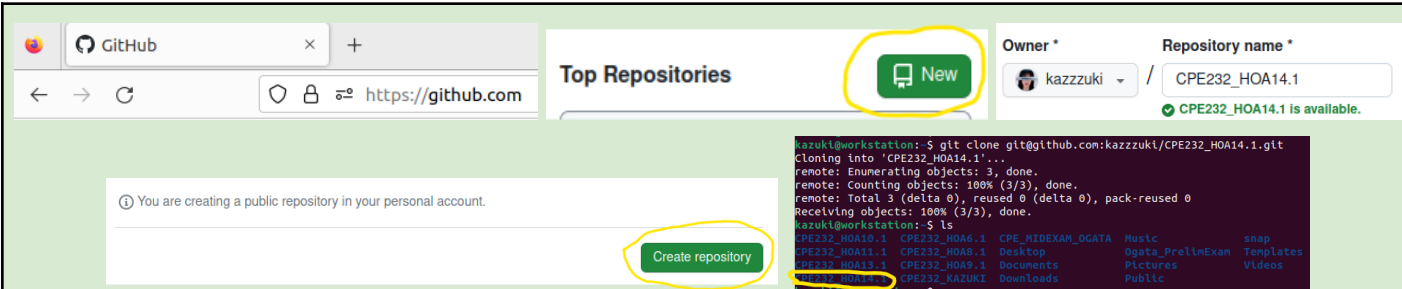


Name: Kazuki A. Ogata	Date Performed: December 8, 2023
Course/Section: CPE 232 - CPE31S5	Date Submitted: December 16, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st semester S.Y 2023-2024
Activity 14: OpenStack Installation (Keystone, Glance, Nova)	
1. Objectives	
Create a workflow to install OpenStack using Ansible as your Infrastructure as Code (IaC).	
2. Intended Learning Outcomes	
<ol style="list-style-type: none"> 1. Analyze the advantages and disadvantages of cloud services 2. Evaluate different Cloud deployment and service models 3. Create a workflow to install and configure OpenStack base services using Ansible as documentation and execution. 	
3. Resources	
Oracle VirtualBox (Hypervisor)	
1x Ubuntu VM or Centos VM	
4. Tasks	
<ol style="list-style-type: none"> 1. Create a new repository for this activity. 2. Create a playbook that converts the steps in the following items in https://docs.openstack.org/install-guide/ <ol style="list-style-type: none"> a. Keystone (Identity Service) b. Glance (Imaging Service) c. Nova (Compute Service) 3. Create different plays in installing per server type (controller, compute etc.) and identify it as a group in the Inventory file. 4. Add, commit and push it to your GitHub repo. 	
5. Output (screenshots and explanations)	
 <p>I created a new repository for this activity and named it "CPE232_HOA14.1".</p> <pre> kazuki@workstation:~/CPE232_HOA14.1\$ mkdir -p roles/Keystone/tasks kazuki@workstation:~/CPE232_HOA14.1\$ mkdir -p roles/Glance/tasks kazuki@workstation:~/CPE232_HOA14.1\$ mkdir -p roles/Nova_Compute_Node/tasks kazuki@workstation:~/CPE232_HOA14.1\$ mkdir -p roles/Nova_Controller_Node/tasks kazuki@workstation:~/CPE232_HOA14.1\$ ls ansible.cfg inventory README.md roles kazuki@workstation:~/CPE232_HOA14.1\$ cd roles kazuki@workstation:~/CPE232_HOA14.1/roles\$ ls Glance Keystone Nova_Compute_Node Nova_Controller_Node </pre> <p>Using the "mkdir" command, I created different directories for the roles per openstack steps.</p>	

```

kazuki@workstation: ~/CPE232_HOA14.1
kazuki@workstation:~/CPE232_HOA14.1$ cat /home/kazuki/CPE232_HOA14.1/roles/Keystone/tasks/main.yml

- name: Create Keystone database
  mysql_db:
    name: keystone
    login_unix_socket: /var/run/mysqld/mysqld.sock

- name: Grant privileges on Keystone database
  mysql_user:
    name: keystone
    password: keystone_ogata
    priv: "keystone.*:ALL"
    state: present
    host: "[{ item }]"
    login_unix_socket: /var/run/mysqld/mysqld.sock
  loop:
    - localhost
    - "%"

- name: Install Keystone packages
  apt:
    name: keystone
    state: present

- name: Edit the /etc/keystone/keystone.conf file (connection)
  lineinfile:
    path: /etc/keystone/keystone.conf
    regexp: '^connection ='
    line: 'connection = mysql+pymysql://keystone:keystone_ogata@controller/keystone'

- name: Edit the /etc/keystone/keystone.conf file (fernet)
  lineinfile:
    path: /etc/keystone/keystone.conf
    regexp: '^#?provider ='
    line: 'provider = fernet'

- name: Populate Identity service database
  command: 'su -s /bin/sh -c "keystone-manage db_sync" keystone '
  become_user: keystone
  async: 3600
  poll: 0

- name: Initialize Fernet key repositories (fernet setup)
  command: "keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone"
  become_user: keystone

- name: Initialize Fernet key repositories (credential setup)
  command: "keystone-manage credential_setup --keystone-user keystone --keystone-group keystone"
  become_user: keystone

- name: Bootstrap the identity service
  command: "keystone-manage bootstrap --bootstrap-password keystone_ogata --bootstrap-admin-url http://controller:5000/v3/ --bootstrap-internal-url http://controller:5000/v3/ --bootstrap-public-url http://controller:5000/v3/ --bootstrap-region-id RegionOne"
  become_user: keystone
  async: 3600
  poll: 0

- name: Configure Apache HTTP server
  lineinfile:
    path: /etc/apache2/apache2.conf
    regexp: '^ServerName'
    line: 'ServerName controller'

- name: Restart Apache Service
  service:
    name: apache2
    state: restarted

- name: Configure administrative account
  shell:
    export OS_USERNAME=admin
    export OS_PASSWORD=keystone_ogata
    export OS_PROJECT_NAME=admin
    export OS_USER_DOMAIN_NAME=Default
    export OS_PROJECT_DOMAIN_NAME=Default
    export OS_AUTH_URL=http://192.168.56.131:5000/v3
    export OS_IDENTITY_API_VERSION=3

```

This is my playbook for installing Keystone, I followed every step on the given document. I started by doing the prerequisites such as creating a database for Keystone, granting privileges on it, then installing the Keystone package. Lastly, do some configurations in keystone.

```
kazuki@workstation:~/CPE232_HOA14.1$ cat /home/kazuki/CPE232_HOA14.1/roles/Glance/tasks/main.yml
```

```
- name: Create Glance database
  mysql_db:
    name: glance
    login_unix_socket: /var/run/mysqld/mysqld.sock

- name: Grant privileges on Glance database
  mysql_user:
    name: glance
    password: glance_ogata
    priv: "glance.*:ALL"
    state: present
    host: "{{ item }}"
    login_unix_socket: /var/run/mysqld/mysqld.sock
  loop:
    - localhost
    - "%"

- name: Source the admin credentials
  shell: ". /home/kazuki/admin-openrc"

- name: Create Glance user
  command: "openstack user create --domain default --password-prompt glance "
  environment:
    OS_USERNAME: admin
    OS_PASSWORD: glance_ogata
    OS_PROJECT_NAME: admin
    OS_AUTH_URL: http://192.168.56.131/v3
  async: 3600
  poll: 0

- name: Add admin role to glance user and service project
  command: "openstack role add --project service --user glance admin"
  environment:
    OS_USERNAME: admin
    OS_PASSWORD: glance_ogata
    OS_PROJECT_NAME: admin
    OS_AUTH_URL: http://192.168.56.131/v3
  async: 3600
  poll: 0

- name: Create glance service entity
  command: 'openstack service create --name glance --description "OpenStack Image" image '
  async: 3600
  poll: 0

- name: Create image service API endpoints (public)
  command: 'openstack endpoint create --region RegionOne image public http://controller:9292 '
  async: 3600
  poll: 0

- name: Create image service API endpoints (internal)
  command: 'openstack endpoint create --region RegionOne image internal http://controller:9292 '
  async: 3600
  poll: 0

- name: Create image service API endpoints (admin)
  command: 'openstack endpoint create --region RegionOne image admin http://controller:9292 '
  async: 3600
  poll: 0

- name: Install Glance packages
  apt:
    name: glance
    state: present

- name: Edit the /etc/glance/glance-api.conf file
  lineinfile:
    path: /etc/glance/glance-api.conf
    regexp: "{{ item.regexp }}"
    line: "{{ item.line }}"
  with_items:
    - { regexp: '^connection = ', line: 'connection = mysql+pymysql://glance:GLANCE_DBPASS@controller:3306/glance' }
    - { regexp: '^www_authenticate_uri = ', line: 'www_authenticate_uri = http://controller:5000' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000' }
    - { regexp: '^memcached_servers = ', line: 'memcached_servers = controller:11211' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
```

```

- { regexp: '^project_domain_name = ', line: 'project_domain_name = service' }
- { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
- { regexp: '^project_name = ', line: 'project_name = service' }
- { regexp: '^username = ', line: 'username = glance' }
- { regexp: '^password = ', line: 'password = GLANCE_PASS' }
- { regexp: '^flavor = ', line: 'flavor = keystone' }
- { regexp: '^stores = ', line: 'stores = file,http' }
- { regexp: '^default_store = ', line: 'default_store = file' }
- { regexp: '^filesystem_store_datadir = ', line: 'filesystem_store_datadir = /var/lib/glance/images/' }
- { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000' }
- { regexp: '^auth_type = ', line: 'auth_type = password' }
- { regexp: '^user_domain_id = ', line: 'user_domain_id = default' }
- { regexp: '^username = ', line: 'username = MY_SERVICE' }
- { regexp: '^system_scope = ', line: 'system_scope = all' }
- { regexp: '^password = ', line: 'password = MY_PASSWORD' }
- { regexp: '^endpoint_id = ', line: 'endpoint_id = ENDPOINT_ID' }
- { regexp: '^region_name = ', line: 'region_name = RegionOne' }
- { regexp: '^use_keystone_quotas = ', line: 'use_keystone_quotas = True' }

- name: Ensure MY_SERVICE reader access
  command: " openstack role add --user MY_SERVICE --user-domain Default --system all reader "
  environment:
    OS_CLOUD: "devstack-system-admin"
  async: 3600
  poll: 0

- name: Populate Image service database
  command: 'su -s /bin/sh -c "glance-manage db_sync" glance'
  become_user: glance
  async: 3600
  poll: 0

- name: Restart image services
  service:
    name: glance-api
    state: restarted

```

This is my playbook for installing Glance, I followed every step on the given document. I started by creating a database for Glance, granting privileges on it, then before installing the glance package, I sourced the admin and created service credentials. Lastly, I installed and configured glance.

```

kazuki@workstation: ~/CPE232_HOA14.1
kazuki@workstation:~/CPE232_HOA14.1$ cat /home/kazuki/CPE232_HOA14.1/roles/Nova_Compute_Node/tasks/main.yml
- name: Install Nova packages
  apt:
    name: nova-compute
    state: present

- name: Edit /etc/nova/nova.conf file
  lineinfile:
    path: /etc/nova/nova.conf
    regexp: "{{ item.regexp }}"
    line: "{{ item.line }}"
  with_items:
    - { regexp: '^transport_url = ', line: 'transport_url = rabbit://openstack:hoa13ogatarabbit@controller' }
    - { regexp: '^auth_strategy = ', line: 'auth_strategy = keystone' }
    - { regexp: '^www_authenticate_uri = ', line: 'www_authenticate_uri = http://controller:5000/' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000/' }
    - { regexp: '^memcached_servers = ', line: 'memcached_servers = controller:11211' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^username = ', line: 'username = nova' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }
    - { regexp: '^send_service_user_token = ', line: 'send_service_user_token = true' }
    - { regexp: '^auth_url = ', line: 'auth_url = https://controller/identity' }
    - { regexp: '^auth_strategy = ', line: 'auth_strategy = keystone' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^username = ', line: 'username = nova' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }
    - { regexp: '^my_ip = ', line: 'my_ip = 10.0.0.31' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000/' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = default' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = default' }
    - { regexp: '^region_name = ', line: 'region_name = RegionOne' }
    - { regexp: '^region_name = ', line: 'region_name = RegionOne' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^username = ', line: 'username = neutron' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }
    - { regexp: '^enabled = ', line: 'enabled = true' }
    - { regexp: '^server_listen = ', line: 'server_listen = 0.0.0.0' }
    - { regexp: '^server_proxyclient_address = ', line: 'server_proxyclient_address = $my_ip' }
    - { regexp: '^novncproxy_base_url = ', line: 'novncproxy_base_url = http://controller:6080/vnc_auto.html' }
    - { regexp: '^api_servers = ', line: 'api_servers = http://controller:9292' }
    - { regexp: '^lock_path = ', line: 'lock_path = /var/lib/nova/tmp' }
    - { regexp: '^region_name = ', line: 'region_name = RegionOne' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000/v3' }
    - { regexp: '^username = ', line: 'username = placement' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }

name: Edit libvirt
lineinfile:
  path: /etc/nova/nova-compute.conf
  regexp: '^virt_type'
  line: 'virt_type = qemu'

name: Restart Compute service
service:
  name: nova-compute
  state: restarted

```

In the given document, there are two installation guides for nova, there is a compute and controller, so I created two playbooks for each of them, and this is my playbook for compute nodes. I started by installing the nova package then configuring the files then restarting the service.

```
kazuki@workstation:~/CPE232_H0A14.1$ cat /home/kazuki/CPE232_H0A14.1/roles/Nova_Controller_Node/tasks/main.yml
```

```
- name: Create Nova Database
  mysql_db:
    name: "{{ item }}"
    state: present
    login_unix_socket: /var/run/mysqld/mysqld.sock
  loop:
    - nova_api
    - nova
    - nova_cell0

- name: Grant All Privileges on Keystone to localhost
  mysql_user:
    name: nova
    host: "{{ item.host }}"
    priv: "{{ item.priv }}"
    password: nova_ogata
    login_unix_socket: /var/run/mysqld/mysqld.sock
  loop:
    - { host: 'localhost' , priv: 'nova_api.*:ALL' }
    - { host: '%' , priv: 'nova_api.*:ALL' }
    - { host: 'localhost' , priv: 'nova.*:ALL' }
    - { host: '%' , priv: 'nova.*:ALL' }
    - { host: 'localhost' , priv: 'nova_cell0.*:ALL' }
    - { host: '%' , priv: 'nova_cell0.*:ALL' }

- name: Source the admin
  shell: ". /home/kazuki/admin-openrc"

- name: Create Nova user
  shell: "openstack user create --domain default --password-prompt nova"
  async: 3600
  poll: 0

- name: Add admin role
  shell: "openstack role add --project service --user nova admin"
  async: 3600
  poll: 0

- name: Create nova service entity
  shell: ' openstack service create --name nova --description "OpenStack Compute" compute '
  async: 3600
  poll: 0

- name: Create the compute API service endpoints (public)
  shell: " openstack endpoint create --region RegionOne compute public http://controller:8774/v2.1 "
  async: 3600
  poll: 0

- name: Create the compute API service endpoints (internal)
  shell: " openstack endpoint create --region RegionOne compute internal http://controller:8774/v2.1 "
  async: 3600
  poll: 0

- name: Create the compute API service endpoints (admin)
  shell: " openstack endpoint create --region RegionOne compute admin http://controller:8774/v2.1 "
  async: 3600
  poll: 0

- name: Install Placement
  apt:
    name: placement-api
    state: present

- name: Create placement service
  shell: "openstack user create --domain default --password-prompt placement"
  async: 3600
  poll: 0

- name: Add placement user to the service project
  shell: "openstack role add --project service --user placement admin"
  async: 3600
  poll: 0

- name: Create the Placement API entry
  shell: ' openstack service create --name placement --description "Placement API" placement '
  async: 3600
  poll: 0
```



```

- name: Create the Placement API service endpoints (public)
  shell: " openstack endpoint create --region RegionOne placement public http://controller:8778 "
  async: 3600
  poll: 0

- name: Create the Placement API service endpoints (internal)
  shell: " openstack endpoint create --region RegionOne placement internal http://controller:8778 "
  async: 3600
  poll: 0

- name: Create the Placement API service endpoints (admin)
  shell: " openstack endpoint create --region RegionOne placement admin http://controller:8778 "
  async: 3600
  poll: 0

- name: Install Nova packages
  apt:
    name:
      - nova-api
      - nova-conductor
      - nova-novncproxy
      - nova-scheduler
    state: present

- name: Edit /etc/nova/nova.conf file
  lineinfile:
    path: /etc/nova/nova.conf
    regexp: "{{ item.regexp }}"
    line: "{{ item.line }}"
  with_items:
    - { regexp: '^connection = ', line: 'connection = mysql+pymysql://nova:nova_ogata@controller/nova_api' }
    - { regexp: '^connection = ', line: 'connection = mysql+pymysql://nova:nova_ogata@controller/nova' }
    - { regexp: '^transport_url = ', line: 'transport_url = rabbit://openstack:hoa13ogatarabbit@controller:5672/' }
    - { regexp: '^auth_strategy = ', line: 'auth_strategy = keystone' }
    - { regexp: '^www_authenticate_url = ', line: 'www_authenticate_url = http://controller:5000/' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000/' }
    - { regexp: '^memcached_servers = ', line: 'memcached_servers = controller:11211' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^username = ', line: 'username = nova' }
    - { regexp: '^password = ', line: 'password = nova_ogata' }
    - { regexp: '^send_service_user_token = ', line: 'send_service_user_token = true' }
    - { regexp: '^auth_url = ', line: 'auth_url = https://controller/identity' }
    - { regexp: '^auth_strategy = ', line: 'auth_strategy = keystone' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^username = ', line: 'username = nova' }
    - { regexp: '^password = ', line: 'password = nova_ogata' }
    - { regexp: '^my_ip = ', line: 'my_ip = 10.0.0.11' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = default' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = default' }
    - { regexp: '^region_name = ', line: 'region_name = RegionOne' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^username = ', line: 'username = neutron' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }
    - { regexp: '^service_metadata_proxy = ', line: 'service_metadata_proxy = true' }
    - { regexp: '^metadata_proxy_shared_secret = ', line: 'metadata_proxy_shared_secret = METADATA_SECRET' }
    - { regexp: '^enabled = ', line: 'enabled = true' }
    - { regexp: '^server_listen = ', line: 'server_listen = $my_ip' }
    - { regexp: '^server_proxyclient_address = ', line: 'server_proxyclient_address = $my_ip' }
    - { regexp: '^api_servers = ', line: 'api_servers = http://controller:9292' }
    - { regexp: '^lock_path = ', line: 'lock_path = /var/lib/nova/tmp' }
    - { regexp: '^region_name = ', line: 'region_name = RegionOne' }
    - { regexp: '^project_domain_name = ', line: 'project_domain_name = Default' }
    - { regexp: '^project_name = ', line: 'project_name = service' }
    - { regexp: '^auth_type = ', line: 'auth_type = password' }
    - { regexp: '^user_domain_name = ', line: 'user_domain_name = Default' }
    - { regexp: '^auth_url = ', line: 'auth_url = http://controller:5000/v3' }
    - { regexp: '^username = ', line: 'username = placement' }
    - { regexp: '^password = ', line: 'password = keystone_ogata' }

```

```

- name: Populate nova-api database
  shell: ' su -s /bin/sh -c "nova-manage api_db sync" nova '
  async: 3600
  poll: 0

- name: Register cell0 database
  shell: ' su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova '
  async: 3600
  poll: 0

- name: Create cell1
  shell: ' su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 --verbose" nova '
  async: 3600
  poll: 0

- name: Populate nova database
  shell: ' su -s /bin/sh -c "nova-manage db sync" nova '
  async: 3600
  poll: 0

- name: Verify nova cell0 and cell1 registration
  shell: ' su -s /bin/sh -c "nova-manage cell_v2 list_cells" nova '
  async: 3600
  poll: 0

- name: Restart compute services
  service:
    name: "{{ item }}"
    state: restarted
  loop:
    - nova-api
    - nova-conductor
    - nova-novncproxy
    - nova-scheduler

```

This is my playbook for nova controller nodes, the same with the other services, I followed every step on the given document. I started by creating databases for nova_api, nova, and nova_cell0, granted privileges to them, sourced the admin, then installed a placement API. Lastly, I installed and configured nova services such as nova-api, nova-conductor, nova-novncproxy, and nov-scheduler.

```

kazuki@workstation:~/CPE232_H0A14.1$ cat inventory

[ubuntu]
192.168.56.131  ansible_connection=ssh

[compute]
192.168.56.131  ansible_connection=ssh

[controller]
192.168.56.131  ansible_connection=ssh

```

```

GNU nano 6.2 openstack_install.yml
---
- hosts: all
  become: true
  tasks:
    - name: Update
      apt:
        update_cache: yes
        state: present

- hosts: ubuntu
  become: true
  roles:
    - Keystone
    - Glance

- hosts: compute
  become: true
  roles:
    - Nova_Compute_Node

- hosts: controller
  become: true
  roles:
    - Nova_Controller_Node

```

The figure on the left is my inventory for this activity. I created 3 groups, 1 for Ubuntu meaning it's for keystone and glance. Then I separated the compute and controller node for nova. The figure on the right is my main playbook. I started by putting a task that updates the target host first, then used the roles syntax to call the playbook I created from separate files and execute/run them using 1 yml file.


```
kazuki@workstation:~/CPE232_H0A14.1$ ansible-playbook --ask-become-pass openstack_install.yml
BECOME password:
```

```
PLAY [all] *****
```

```
TASK [Gathering Facts] *****
ok: [192.168.56.131]
```

```
TASK [Update] *****
changed: [192.168.56.131]
```

```
PLAY [ubuntu] *****
```

```
TASK [Gathering Facts] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Create Keystone database] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Grant privileges on Keystone database] *****
ok: [192.168.56.131] => (item=localhost)
ok: [192.168.56.131] => (item=%)
```

```
TASK [Keystone : Install Keystone packages] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Edit the /etc/keystone/keystone.conf file (connection)] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Edit the /etc/keystone/keystone.conf file (fernet)] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Populate Identity service database] *****
changed: [192.168.56.131]
```

```
TASK [Keystone : Initialize Fernet key repositories (fernet setup)] *****
changed: [192.168.56.131]
```

```
TASK [Keystone : Initialize Fernet key repositories (credential setup)] *****
changed: [192.168.56.131]
```

```
TASK [Keystone : Bootstrap the identity service] *****
changed: [192.168.56.131]
```

```
TASK [Keystone : Configure Apache HTTP server] *****
ok: [192.168.56.131]
```

```
TASK [Keystone : Restart Apache Service] *****
changed: [192.168.56.131]
```

```
TASK [Keystone : Configure administrative account] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create Glance database] *****
ok: [192.168.56.131]
```

```
TASK [Glance : Grant privileges on Glance database] *****
ok: [192.168.56.131] => (item=localhost)
ok: [192.168.56.131] => (item=%)
```

```
TASK [Glance : Source the admin credentials] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create Glance user] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Add admin role to glance user and service project] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create glance service entity] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create image service API endpoints (public)] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create image service API endpoints (internal)] *****
changed: [192.168.56.131]
```

```
TASK [Glance : Create image service API endpoints (admin)] *****
changed: [192.168.56.131]
```

```

TASK [Glance : Install Glance packages] *****
ok: [192.168.56.131]

TASK [Glance : Edit the /etc/glance/glance-api.conf file] *****
ok: [192.168.56.131] => (item={'regex': '^connection = ', 'line': 'connection = mysql+pymysql://glance:GLANCE_DBPASS@controller/glance'})
ok: [192.168.56.131] => (item={'regex': '^www_authenticate_uri = ', 'line': 'www_authenticate_uri = http://controller:5000'})
ok: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000'})
ok: [192.168.56.131] => (item={'regex': '^memcached_servers = ', 'line': 'memcached_servers = controller:11211'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = glance'})
changed: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = GLANCE_PASS'})
ok: [192.168.56.131] => (item={'regex': '^flavor = ', 'line': 'flavor = keystone'})
ok: [192.168.56.131] => (item={'regex': '^stores = ', 'line': 'stores = file,http'})
ok: [192.168.56.131] => (item={'regex': '^default_store = ', 'line': 'default_store = file'})
ok: [192.168.56.131] => (item={'regex': '^filesystem_store_datadir = ', 'line': 'filesystem_store_datadir = /var/lib/glance/images/'})
ok: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_id = ', 'line': 'user_domain_id = default'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = MY_SERVICE'})
ok: [192.168.56.131] => (item={'regex': '^system_scope = ', 'line': 'system_scope = all'})
changed: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = MY_PASSWORD'})
ok: [192.168.56.131] => (item={'regex': '^endpoint_id = ', 'line': 'endpoint_id = ENDPOINT_ID'})
ok: [192.168.56.131] => (item={'regex': '^region_name = ', 'line': 'region_name = RegionOne'})
ok: [192.168.56.131] => (item={'regex': '^use_keystone_quotas = ', 'line': 'use_keystone_quotas = True'})

TASK [Glance : Ensure MY_SERVICE reader access] *****
changed: [192.168.56.131]

TASK [Glance : Populate Image service database] *****
changed: [192.168.56.131]

TASK [Glance : Restart image services] *****
changed: [192.168.56.131]

PLAY [compute] *****

TASK [Gathering Facts] *****
ok: [192.168.56.131]

TASK [Nova_Compute_Node : Install Nova packages] *****
ok: [192.168.56.131]

TASK [Nova_Compute_Node : Edit /etc/nova/nova.conf file] *****
changed: [192.168.56.131] => (item={'regex': '^transport_url = ', 'line': 'transport_url = rabbit://openstack:hoai3ogatarabbit@controller'})
ok: [192.168.56.131] => (item={'regex': '^auth_strategy = ', 'line': 'auth_strategy = keystone'})
ok: [192.168.56.131] => (item={'regex': '^www_authenticate_uri = ', 'line': 'www_authenticate_uri = http://controller:5000/'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000/'})
ok: [192.168.56.131] => (item={'regex': '^memcached_servers = ', 'line': 'memcached_servers = controller:11211'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = nova'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})
ok: [192.168.56.131] => (item={'regex': '^send_service_user_token = ', 'line': 'send_service_user_token = true'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = https://controller/identity'})
ok: [192.168.56.131] => (item={'regex': '^auth_strategy = ', 'line': 'auth_strategy = keystone'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = nova'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})
changed: [192.168.56.131] => (item={'regex': '^my_ip = ', 'line': 'my_ip = 10.0.0.31'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
changed: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = default'})
changed: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = default'})
ok: [192.168.56.131] => (item={'regex': '^region_name = ', 'line': 'region_name = RegionOne'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = neutron'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})
ok: [192.168.56.131] => (item={'regex': '^enabled = ', 'line': 'enabled = true'})
changed: [192.168.56.131] => (item={'regex': '^server_listen = ', 'line': 'server_listen = 0.0.0.0'})
ok: [192.168.56.131] => (item={'regex': '^server_proxyclient_address = ', 'line': 'server_proxyclient_address = $my_ip'})
ok: [192.168.56.131] => (item={'regex': '^novncproxy_base_url = ', 'line': 'novncproxy_base_url = http://controller:6080/vnc_auto.html'})
changed: [192.168.56.131] => (item={'regex': '^api_servers = ', 'line': 'api_servers = http://controller:9292'})
ok: [192.168.56.131] => (item={'regex': '^lock_path = ', 'line': 'lock_path = /var/lib/nova/tmp'})
ok: [192.168.56.131] => (item={'regex': '^region_name = ', 'line': 'region_name = RegionOne'})
changed: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000/v3'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = placement'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})

```

```
TASK [Nova_Compute_Node : Edit libvirt] *****
ok: [192.168.56.131]
```

```
TASK [Nova_Compute_Node : Restart Compute service] *****
changed: [192.168.56.131]
```

```
PLAY [controller] *****
```

```
TASK [Gathering Facts] *****
ok: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create Nova Database] *****
ok: [192.168.56.131] => (item=nova_api)
ok: [192.168.56.131] => (item=nova)
ok: [192.168.56.131] => (item=nova_cell0)
```

```
TASK [Nova_Controller_Node : Grant All Privileges on Keystone to localhost] *****
changed: [192.168.56.131] => (item={'host': 'localhost', 'priv': 'nova_api.*:ALL'})
changed: [192.168.56.131] => (item={'host': '%', 'priv': 'nova_api.*:ALL'})
changed: [192.168.56.131] => (item={'host': 'localhost', 'priv': 'nova.*:ALL'})
changed: [192.168.56.131] => (item={'host': '%', 'priv': 'nova.*:ALL'})
changed: [192.168.56.131] => (item={'host': 'localhost', 'priv': 'nova_cell0.*:ALL'})
changed: [192.168.56.131] => (item={'host': '%', 'priv': 'nova_cell0.*:ALL'})
```

```
TASK [Nova_Controller_Node : Source the admin] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create Nova user] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Add admin role] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create nova service entity] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the compute API service endpoints (public)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the compute API service endpoints (internal)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the compute API service endpoints (admin)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Install Placement] *****
ok: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create placement service] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Add placement user to the service project] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the Placement API entry] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the Placement API service endpoints (public)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the Placement API service endpoints (internal)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Create the Placement API service endpoints (admin)] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Install Nova packages] *****
changed: [192.168.56.131]
```

```
TASK [Nova_Controller_Node : Edit /etc/nova/nova.conf file] *****
changed: [192.168.56.131] => (item={'regex': '^connection = ', 'line': 'connection = mysql+pymysql://nova:nova_ogata@controller/nova_api'})
changed: [192.168.56.131] => (item={'regex': '^connection = ', 'line': 'connection = mysql+pymysql://nova:nova_ogata@controller/nova'})
changed: [192.168.56.131] => (item={'regex': '^transport_url = ', 'line': 'transport_url = rabbit://openstack:hoa13ogatarabbit@controller:5672/'})
ok: [192.168.56.131] => (item={'regex': '^auth_strategy = ', 'line': 'auth_strategy = keystone'})
ok: [192.168.56.131] => (item={'regex': '^www_authenticate_uri = ', 'line': 'www_authenticate_uri = http://controller:5000/'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000/'})
ok: [192.168.56.131] => (item={'regex': '^memcached_servers = ', 'line': 'memcached_servers = controller:11211'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
changed: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
changed: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
```

```

ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = nova'})
changed: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = nova_ogata'})
ok: [192.168.56.131] => (item={'regex': '^send_service_user_token = ', 'line': 'send_service_user_token = true'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = https://controller/identity'})
ok: [192.168.56.131] => (item={'regex': '^auth_strategy = ', 'line': 'auth_strategy = keystone'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
ok: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
ok: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = nova'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = nova_ogata'})
changed: [192.168.56.131] => (item={'regex': '^my_ip = ', 'line': 'my_ip = 10.0.0.11'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
changed: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = default'})
changed: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = default'})
ok: [192.168.56.131] => (item={'regex': '^region_name = ', 'line': 'region_name = RegionOne'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = neutron'})
changed: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})
ok: [192.168.56.131] => (item={'regex': '^service_metadata_proxy = ', 'line': 'service_metadata_proxy = true'})
ok: [192.168.56.131] => (item={'regex': '^metadata_proxy_shared_secret = ', 'line': 'metadata_proxy_shared_secret = METADATA_SECRET'})
ok: [192.168.56.131] => (item={'regex': '^enabled = ', 'line': 'enabled = true'})
changed: [192.168.56.131] => (item={'regex': '^server_listen = ', 'line': 'server_listen = $my_ip'})
ok: [192.168.56.131] => (item={'regex': '^server_proxyclient_address = ', 'line': 'server_proxyclient_address = $my_ip'})
changed: [192.168.56.131] => (item={'regex': '^api_servers = ', 'line': 'api_servers = http://controller:9292'})
ok: [192.168.56.131] => (item={'regex': '^lock_path = ', 'line': 'lock_path = /var/lib/nova/tmp'})
ok: [192.168.56.131] => (item={'regex': '^region_name = ', 'line': 'region_name = RegionOne'})
changed: [192.168.56.131] => (item={'regex': '^project_domain_name = ', 'line': 'project_domain_name = Default'})
ok: [192.168.56.131] => (item={'regex': '^project_name = ', 'line': 'project_name = service'})
ok: [192.168.56.131] => (item={'regex': '^auth_type = ', 'line': 'auth_type = password'})
changed: [192.168.56.131] => (item={'regex': '^user_domain_name = ', 'line': 'user_domain_name = Default'})
changed: [192.168.56.131] => (item={'regex': '^auth_url = ', 'line': 'auth_url = http://controller:5000/v3'})
changed: [192.168.56.131] => (item={'regex': '^username = ', 'line': 'username = placement'})
ok: [192.168.56.131] => (item={'regex': '^password = ', 'line': 'password = keystone_ogata'})

```

```

TASK [Nova_Controller_Node : Populate nova-api database] *****
changed: [192.168.56.131]

```

```

TASK [Nova_Controller_Node : Register cell0 database] *****
changed: [192.168.56.131]

```

```

TASK [Nova_Controller_Node : Create cell1] *****
changed: [192.168.56.131]

```

```

TASK [Nova_Controller_Node : Populate nova database] *****
changed: [192.168.56.131]

```

```

TASK [Nova_Controller_Node : Verify nova cell0 and cell1 registration] *****
changed: [192.168.56.131]

```

```

TASK [Nova_Controller_Node : Restart compute services] *****
changed: [192.168.56.131] => (item=nova-api)
changed: [192.168.56.131] => (item=nova-conductor)
changed: [192.168.56.131] => (item=nova-novncproxy)
changed: [192.168.56.131] => (item=nova-scheduler)

```

```

PLAY RECAP *****
192.168.56.131      : ok=59   changed=42   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

```

This is the output of my main playbook, there are now errors meaning it was successfully executed. There are a lot of “ok” in the output, this is because I already executed them because whenever I run the playbook then I encounter an error, I re-run the playbook, the proofs below will show that the executing of this playbook was successful.

DATABASES PROOF FOR KEYSTONE, GLANCE, AND NOVA:

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| glance   |
| information_schema |
| keystone |
| mysql    |
| nova     |
| nova_api |
| nova_cell0 |
| performance_schema |
| sys      |
+-----+
9 rows in set (0.001 sec)
```

KEYSTONE PROOF:

```
MariaDB [(none)]> SHOW GRANTS FOR 'keystone'@'localhost';
+-----+
| Grants for keystone@localhost |
+-----+
| GRANT USAGE ON *.* TO 'keystone'@'localhost' IDENTIFIED BY PASSWORD '*9C4C42C0184110DA32E4BB60379DCAFFF19F4401' |
| GRANT ALL PRIVILEGES ON `keystone`.* TO 'keystone'@'localhost' |
+-----+
2 rows in set (0.000 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'keystone'@'%';
+-----+
| Grants for keystone@% |
+-----+
| GRANT USAGE ON *.* TO 'keystone'@'%' IDENTIFIED BY PASSWORD '*9C4C42C0184110DA32E4BB60379DCAFFF19F4401' |
| GRANT ALL PRIVILEGES ON `keystone`.* TO 'keystone'@'%' |
+-----+
2 rows in set (0.000 sec)

MariaDB [(none)]> keystone-manage --version;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'keystone-manage --version' at line 1
MariaDB [(none)]> Ctrl-C -- exit!
Aborted
kazuki@server2:~$ keystone-manage --version
21.0.1
```

```
kazuki@server2:~$ 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 2023-12-15 23:39:20 +08; 12min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 991 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 1145 (apache2)
    Tasks: 46 (limit: 4594)
   Memory: 80.7M
      CPU: 1.264s
   CGroup: /system.slice/apache2.service
           └─1145 /usr/sbin/apache2 -k start
             └─1173 "(wsgi:keystone-pu" -k start
               └─1174 "(wsgi:keystone-pu" -k start
                 └─1175 "(wsgi:keystone-pu" -k start
                   └─1176 "(wsgi:keystone-pu" -k start
                     └─1193 "(wsgi:keystone-pu" -k start
                       └─1204 "(wsgi:placement-a" -k start
                         └─1215 "(wsgi:placement-a" -k start
                           └─1222 "(wsgi:placement-a" -k start
                             └─1227 "(wsgi:placement-a" -k start
                               └─1229 "(wsgi:placement-a" -k start
                                 └─1296 /usr/sbin/apache2 -k start
                                   └─1297 /usr/sbin/apache2 -k start
                                     └─1364 /usr/sbin/apache2 -k start
                                       └─1365 /usr/sbin/apache2 -k start
                                         └─1366 /usr/sbin/apache2 -k start

Dec 15 23:39:19 server2 systemd[1]: Starting The Apache HTTP Server...
Dec 15 23:39:20 server2 systemd[1]: Started The Apache HTTP Server.
```


GLANCE PROOF:

```
MariaDB [(none)]> SHOW GRANTS FOR 'glance'@'localhost';
+-----+
| Grants for glance@localhost |
+-----+
| GRANT USAGE ON *.* TO 'glance'@'localhost' IDENTIFIED BY PASSWORD '*F2FBA2FD64B7AE7C5E2BBE65152D0E5B913F2905' |
| GRANT ALL PRIVILEGES ON `glance`.* TO 'glance'@'localhost' |
+-----+
2 rows in set (0.001 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'glance'@'%';
+-----+
| Grants for glance@% |
+-----+
| GRANT USAGE ON *.* TO 'glance'@'%' IDENTIFIED BY PASSWORD '*F2FBA2FD64B7AE7C5E2BBE65152D0E5B913F2905' |
| GRANT ALL PRIVILEGES ON `glance`.* TO 'glance'@'%' |
+-----+
2 rows in set (0.000 sec)
```

```
kazuki@server2:~$ sudo systemctl status glance-api
● glance-api.service - OpenStack Image Service API
   Loaded: loaded (/lib/systemd/system/glance-api.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-12-15 23:40:00 +08; 15min ago
     Docs: man:glance-api(1)
  Main PID: 2668 (glance-api)
    Tasks: 3 (limit: 4594)
   Memory: 119.2M
      CPU: 6.192s
   CGroup: /system.slice/glance-api.service
           └─2668 /usr/bin/python3 /usr/bin/glance-api --config-file=/etc/glance/glance-api.conf
             └─2834 /usr/bin/python3 /usr/bin/glance-api --config-file=/etc/glance/glance-api.conf
               └─2835 /usr/bin/python3 /usr/bin/glance-api --config-file=/etc/glance/glance-api.conf

Dec 15 23:40:00 server2 systemd[1]: Started OpenStack Image Service API.
lines 1-14/14 (END)
```

NOVA_COMPUTE_NODE PROOF:

```
kazuki@server2:~$ sudo systemctl status nova-compute
● nova-compute.service - OpenStack Compute
   Loaded: loaded (/lib/systemd/system/nova-compute.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-12-15 23:58:19 +08; 5s ago
  Main PID: 9148 (nova-compute)
    Tasks: 1 (limit: 4594)
   Memory: 92.1M
      CPU: 1.770s
   CGroup: /system.slice/nova-compute.service
           └─9148 /usr/bin/python3 /usr/bin/nova-compute --config-file=/etc/nova/nova.conf --co

Dec 15 23:58:24 server2 nova-compute[9148]: start_index = consume_optional(start_index)
Dec 15 23:58:24 server2 nova-compute[9148]: File "/usr/lib/python3.10/argparse.py", line 2031,
Dec 15 23:58:24 server2 nova-compute[9148]: take_action(action, args, option_string)
Dec 15 23:58:24 server2 nova-compute[9148]: File "/usr/lib/python3.10/argparse.py", line 1955,
Dec 15 23:58:24 server2 nova-compute[9148]: action(self, namespace, argument_values, option_s
Dec 15 23:58:24 server2 nova-compute[9148]: File "/usr/lib/python3/dist-packages/oslo_config/cf
Dec 15 23:58:24 server2 nova-compute[9148]: ConfigParser._parse_file(values, namespace)
Dec 15 23:58:24 server2 nova-compute[9148]: File "/usr/lib/python3/dist-packages/oslo_config/cf
Dec 15 23:58:24 server2 nova-compute[9148]: raise ConfigFileParseError(pe.filename, str(pe))
Dec 15 23:58:24 server2 nova-compute[9148]: oslo_config.cfg.ConfigFileParseError: Failed to parse
lines 1-20/20 (END)
```


NOVA_CONTROLLER_NODE PROOF:

```
kazuki@server2:~$ sudo systemctl status nova-api
● nova-api.service - OpenStack Compute API
   Loaded: loaded (/lib/systemd/system/nova-api.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-12-16 00:01:28 +08; 1s ago
     Docs: man:nova-api(1)
    Main PID: 10265 (nova-api)
      Tasks: 1 (limit: 4594)
     Memory: 50.3M
        CPU: 802ms
    CGroup: /system.slice/nova-api.service
            └─10265 /usr/bin/python3 /usr/bin/nova-api --config-file=/etc/nova/nova.conf --log-file=/var/log/nova/nova-api.log

Dec 16 00:01:28 server2 systemd[1]: Started OpenStack Compute API.
Dec 16 00:01:29 server2 nova-api[10265]: Modules with known eventlet monkey patching issues were imported prior to eventlet monkey
lines 1-13/13 (END)

kazuki@server2:~$ sudo systemctl status nova-conductor
● nova-conductor.service - OpenStack Compute Conductor
   Loaded: loaded (/lib/systemd/system/nova-conductor.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-12-16 00:01:51 +08; 3s ago
     Docs: man:nova-conductor(1)
    Main PID: 10375 (nova-conductor)
      Tasks: 1 (limit: 4594)
     Memory: 89.6M
        CPU: 1.404s
    CGroup: /system.slice/nova-conductor.service
            └─10375 /usr/bin/python3 /usr/bin/nova-conductor --config-file=/etc/nova/nova.conf --log-file=/var/log/nova/nova-condu

Dec 16 00:01:54 server2 nova-conductor[10375]:     return super(_CachedArgumentParser, self).parse_args(args, namespace)
Dec 16 00:01:54 server2 nova-conductor[10375]: File "/usr/lib/python3.10/argparse.py", line 1845, in parse_args
Dec 16 00:01:54 server2 nova-conductor[10375]:     args, argv = self.parse_known_args(args, namespace)
Dec 16 00:01:54 server2 nova-conductor[10375]: File "/usr/lib/python3.10/argparse.py", line 1878, in parse_known_args
Dec 16 00:01:54 server2 nova-conductor[10375]:     namespace, args = self._parse_known_args(args, namespace)
Dec 16 00:01:54 server2 nova-conductor[10375]: File "/usr/lib/python3.10/argparse.py", line 2091, in _parse_known_args
Dec 16 00:01:54 server2 nova-conductor[10375]:     start_index = consume_optional(start_index)
Dec 16 00:01:54 server2 nova-conductor[10375]: File "/usr/lib/python3.10/argparse.py", line 2031, in consume_optional
Dec 16 00:01:54 server2 nova-conductor[10375]:     take_action(action, args, option_string)
Dec 16 00:01:54 server2 nova-conductor[10375]: File "/usr/lib/python3.10/argparse.py", line 1955, in take_action
lines 1-21/21 (END)

kazuki@server2:~$ sudo systemctl status nova-novncproxy
● nova-novncproxy.service - OpenStack Compute novncproxy
   Loaded: loaded (/lib/systemd/system/nova-novncproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-12-16 00:02:19 +08; 2s ago
     Docs: man:nova-novncproxy(1)
    Main PID: 10526 (nova-novncproxy)
      Tasks: 1 (limit: 4594)
     Memory: 78.8M
        CPU: 1.190s
    CGroup: /system.slice/nova-novncproxy.service
            └─10526 /usr/bin/python3 /usr/bin/nova-novncproxy --config-file=/etc/nova/nova.conf --log-file=/var/log/nova/nova-novnc

Dec 16 00:02:19 server2 systemd[1]: Started OpenStack Compute novncproxy.
Dec 16 00:02:21 server2 nova-novncproxy[10526]: Modules with known eventlet monkey patching issues were imported prior to eventlet monkey
lines 1-13/13 (END)

kazuki@server2:~$ sudo systemctl status nova-scheduler
● nova-scheduler.service - OpenStack Compute Scheduler
   Loaded: loaded (/lib/systemd/system/nova-scheduler.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-12-16 00:02:44 +08; 166ms ago
     Docs: man:nova-scheduler(1)
    Main PID: 10685 (nova-scheduler)
      Tasks: 1 (limit: 4594)
     Memory: 3.0M
        CPU: 27ms
    CGroup: /system.slice/nova-scheduler.service
            └─10685 /usr/bin/python3 /usr/bin/nova-scheduler --config-file=/etc/nova/nova.conf --log-file=/var/log/nova/nova-schedu

Dec 16 00:02:44 server2 systemd[1]: Started OpenStack Compute Scheduler.
lines 1-12/12 (END)
```

```

kazuki@workstation:~/CPE232_HOA14.1$ git add .
kazuki@workstation:~/CPE232_HOA14.1$ git commit -m "HOA14"
[main 483a6f4] HOA14
7 files changed, 492 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 inventory
create mode 100644 openstack_install.yml
create mode 100644 roles/Glance/tasks/main.yml
create mode 100644 roles/Keystone/tasks/main.yml
create mode 100644 roles/Nova_Compute_Node/tasks/main.yml
create mode 100644 roles/Nova_Controller_Node/tasks/main.yml
kazuki@workstation:~/CPE232_HOA14.1$ git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (18/18), 4.65 KiB | 1.55 MiB/s, done.
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:kazzzuki/CPE232_HOA14.1.git
ab5ea5e..483a6f4 main -> main

```

I used the command “git add .” to add all the created files to my github repository then commit it using the “git commit” command and then lastly push it using the “git push” command.

GITHUB REPOSITORY LINK: https://github.com/kazzzuki/CPE232_HOA14.1

Reflections:

Answer the following:

1. Describe Keystone, Glance and Nova services

- Based on the given document of OpenStack, Keystone is the identity service in OpenStack, Glance is the image service in OpenStack, and Nova is the compute service in Openstack. The Keystone service provides different services like authentication and authorization. The Glance service provides a repository for storing images, and they support different image formats. The Nova service provides the foundation for LaaS in OpenSTack. Overall, these services are the core components of the OpenSTack cloud infrastructure. Keystone make sure that identity management is secured, Glance handles and manages the image, and Nova is the one in charge of compute resources.

Conclusions:

In this activity, I successfully created a workflow in installing OpenStack using Ansible as IaC. This workflow involves a step by step approach such as the installation and configuration of different services of OpenStack like Keystone, Glance, and Nova. I learned about these different cloud services and their advantages and disadvantages. For Keystone, it provides an identity service that simplifies authentication to access different OpenStack services, while its con is it has a complex configuration and is so hard to troubleshoot based on my experience. For Glance, it serves as a repository for images making it easier to manage and share across the cloud but storing too many images can consume large amounts of storage space. Lastly, for Nova, it allows easy scaling by adding compute nodes in the cloud and it is also flexible but it also has a complex configuration like setting up a network can be too hard to configure. These learnings help me understand when tp use these OpenStack services.