Name: Maxine Audrey D. Pulao	Date Performed: December 10, 2022
Course/Section: CPE31S2	Date Submitted: December 11, 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022-2023

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

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

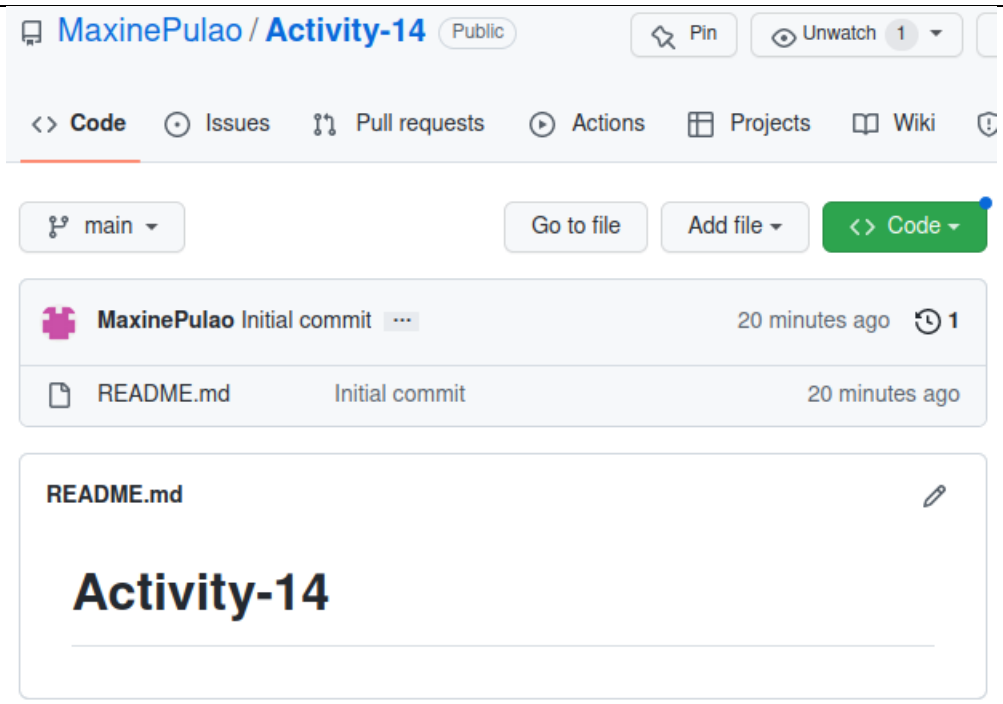
Oracle VirtualBox (Hypervisor)

1x Ubuntu VM or Centos VM

4. Tasks

1. Create a new repository for this activity.

```
pulao@pulao-VirtualBox:~/ansible$ git clone git@github.com:MaxinePulao/Activity
14.git
Cloning into 'Activity-14'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```



- Creating repositories allocated for Activity 14's files.

2. Create a playbook that converts the steps in the following items in https://docs.openstack.org/install-guide/
    a. Keystone (Identity Service)

```yaml
#Keystone

##Installing the mod_wsgi
- name: Install the prerequisites for mod_wsgi
  apt:
    name:
       - apache2
       - apache2-utils
       - libexpat1
       - ssl-cert
       - python3
       - libapache2-mod-wsgi-py3
  when: ansible_distribution == "Ubuntu"

- name:
  copy:
    dest: /var/www/html/test_script.py
    content: |
      def application(environ,start_response):
          status = '200 OK'
          html = '\n' \
                 '\n' \
                 ' mod_wsgi is working \n' \
                 '\n' \
                 '\n'
          response_header = [('Content-type','text/html')]
          start_response(status,response_header)
          return [html]
- name: Create a seperate apache config to serve our python script over HTTP
  copy:
    dest: /etc/apache2/conf-available/wsgi.conf
    content: |
      WSGIScriptAlias /test_wsgi /var/www/html/test_script.py
#- name: restart apache server
#  shell: sudo systemctl restart apache2

- name: install the keystone package
  apt:
    name: keystone
  when: ansible_distribution == "Ubuntu"

#editing the [database] /etc/keystone/keystone.conf
#editing the [token] /etc/keystone/keystone.conf
#LINK: docs.openstack.org/keystone/yoga/install/keystone-install-ubuntu.html

#- name: populate the identity service repositories
#  shell: su -s /bin/sh -c "keystone-manage db_sync" keystone
#  when: ansible_distribution == "Ubuntu"
```

```
      - libexpat1
      - ssl-cert
      - python3
      - libapache2-mod-wsgi-py3
  when: ansible_distribution == "Ubuntu"

- name:
  copy:
    dest: /var/www/html/test_script.py
    content: |
      def application(environ,start_response):
          status = '200 OK'
          html = '\n' \
                 '\n' \
                 ' mod_wsgi is working \n' \
                 '\n' \
                 '\n'
          response_header = [('Content-type','text/html')]
          start_response(status,response_header)
          return [html]
- name: Create a seperate apache config to serve our python script over HTTP
  copy:
    dest: /etc/apache2/conf-available/wsgi.conf
    content: |
      WSGIScriptAlias /test_wsgi /var/www/html/test_script.py
#- name: restart apache server
#  shell: sudo systemctl restart apache2

- name: install the keystone package
  apt:
    name: keystone
  when: ansible_distribution == "Ubuntu"

#editing the [database] /etc/keystone/keystone.conf
#editing the [token] /etc/keystone/keystone.conf
#LINK: docs.openstack.org/keystone/yoga/install/keystone-install-ubuntu.html

#- name: populate the identity service repositories
#  shell: su -s /bin/sh -c "keystone-manage db_sync" keystone
#  when: ansible_distribution == "Ubuntu"

- name: initializing the fernet repositories (1)
  shell: keystone-manage fernet_setup --keystone-user keystone --keystone-group ke
  when: ansible_distribution == "Ubuntu"

- name: initializing the fernet repositories (2)
  shell: keystone-manage credential_setup --keystone-user keystone --keystone-grou

```

```
#- name: bootstrap the identity service (1)
#  shell: keystone-manage bootstrap --bootstrap-password 1234 --bootstrap-admin-ur
controller:5000/v3/ --bootstrap-internal-url http://controller:5000/v3/ --bootstra
http://controller:5000/v3/ --bootstrap-region-id RegionOne

- name: configure apache http server
  copy:
    dest: /etc/apache2/apache2.conf
    content: |
      ServerName controller
  when: ansible_distribution == "Ubuntu"

- name: configuring administrative account by setting the proper environmental var
  shell: export OS_USERNAME=admin

- name: configuring administrative account by setting the proper environmental var
  shell: export OS_PASSWORD=1234

- name: configuring administrative account by setting the proper environmental var
  shell: export OS_PROJECT_NAME=admin

- name: configuring administrative account by setting the proper environmental var
  shell: OS_USER_DOMAIN_NAME=Default

- name: configuring administrative account by setting the proper environmental var
  shell: OS_PROJECT_DOMAIN_NAME=Default

- name: configuring administrative account by setting the proper environmental var
  shell: OS_AUTH_URL=http://controller:5000/v3

- name: configuring administrative account by setting the proper environmental var
  shell: OS_IDENTITY_API_VERSION=3

- block:
  - name: Verifying if already active and running the keystone.
    shell: keystone-manage --help
    register: keystone_service

  - debug:
      msg="{{ keystone_service }}"
```

b. Glance (Imaging Service)

Open

main.yml
~/ansible/Activity-14/roles/glance/tasks

Save

```
1 #Glance
2
3 #- name:  source the admin credential to gain access to admin-only CLI commands
4 #  shell: . admin-openrc
5 #  when: ansible_distribution == "Ubuntu"
6
7 #- name: create service credentials
8 #  shell: openstack user create --domain default --password-prompt glance
9 #  when: ansible_distribution == "Ubuntu"
10
11 #- name: adding the admin role to the glance user and service project
12 #  shell: openstack role add --project service --user glance admin
13 #  when: ansible_distribution == "Ubuntu"
14
15 #- name: create the glance service entity
16 #  shell: openstack service create --name glance --description "OpenStack Image
17 #  when: ansible_distribution == "Ubuntu"
18
19 #- name: create the image service API endpoints (1)
20 #  shell: openstack endpoint create --region RegionOne image public http://cont
21 #  when: ansible_distribution == "Ubuntu"
22
23 #- name: create the image service API endpoints (2)
24 #  shell: openstack endpoint create --region RegionOne image internal http://co
25 #  when: ansible_distribution == "Ubuntu"
26
27 #- name: create the image service API endpoints (3)
28 #  shell: openstack endpoint create --region RegionOne image admin http://contr
29 #  when: ansible_distribution == "Ubuntu"
30
31 ##Register quota limits (Optional)
32
33 ##install and config
34 - name: install and configure components of glance
35   apt:
36     name: glance
37   when: ansible_distribution == "Ubuntu"
```

```
38
39 #editing the [database] /etc/glance/glance-api.conf
40
41 - name: configuring database access
42   copy:
43     dest: /etc/glance/glance-api.conf
44     content: |
45       [database]
46       connection = mysql+pymysql://glance:1234@controller/glance
47       [keystone_authtoken]
48       www_authenicate_uri =  http://controller:5000
49       auth_url = http://controller:5000
50       memcached_servers = controller:11211
51       auth_type = password
52       project_domain_name = Default
53       user_domain_name = Default
54       project_name = service
55       username = glance
56       password = 1234
57       [paste_deploy]
58       flavor = keystone
59
60 - name: configuring the local file system store and location of image files
61   copy:
62     dest: /etc/glance/glance-api.conf
63     content: |
64       [glance_store]
65       stores = file, http
66       default_store = file
67       filesystem_store_datadir = /var/lib/glance/images/
68
69 - name: configuring the access to keystone
70   copy:
71     dest: /etc/glance/glance-api.conf
72     content: |
73       [oslo_limit]
74       auth_url = http://controller:5000
75       auth_type = password
76       user_domain_id = default
77       username = MY_SERVICE
78       system_scope = all
79       password = 1234
80       endpoint_id = ENDPOINT_ID
81       region_name = RegionOne
82
83 - name: enable per-tenant quotas
84   copy:
85     dest: /etc/glance/glance-api.conf
```

```
    content: |
      [DEFAULT]
      use_keystone_quotas = True

#- name: MY_SERVICE read access to systemscope resources
#  shell: openstack role add --user MY_SERVICE --user-domain Default --system all
#  when: ansible_distribution == "Ubuntu"

#- name: populate the image service database
#  shell: su -s  /bin/sh -c "glance-manage db_sync" glance
#  when: ansible_distribution == "Ubuntu"

- name: restart the image services
  shell: service glance-api restart
  when: ansible_distribution == "Ubuntu"

- block:
  - name: Verifying if already installed Glance.
    shell: glance --version
    register: glance_version

  - debug:
      msg="{{ glance_version }}"

- block:
  - name: Verifying if already active and running the Glance.
    shell: systemctl status glance-api
    register: glance_service

  - debug:
      msg="{{ glance_service }}"
Footer
```

c. Nova (Compute Service)

```yaml
#Nova

- name: install the packages
  apt:
    name: nova-compute
  when: ansible_distribution == "Ubuntu"

- name: configuring RabbitMQ message queue access
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [DEFAULT]
      tranport_url = rabbit://openstack:1234@controller
      my_ip = 192.168.56.119

- name: configuring identity service access (1)
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [api]
      auth_strategy = keystone

- name: configuring identity service access (2)
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [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 = nova
      password = 1234

- name: enable and configure remote console access
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [vnc]
      enabled = true
      server_listen = 0.0.0.0
      server_proxyclient_address = $my_ip
      novncproxy_base_url = http://controller:6080/vnc_auto.html

- name: configure the location of the image service API
```

```
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [DEFAULT]
      tranport_url = rabbit://openstack:1234@controller
      my_ip = 192.168.56.119

- name: configuring identity service access (1)
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [api]
      auth_strategy = keystone

- name: configuring identity service access (2)
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [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 = nova
      password = 1234

- name: enable and configure remote console access
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [vnc]
      enabled = true
      server_listen = 0.0.0.0
      server_proxyclient_address = $my_ip
      novncproxy_base_url = http://controller:6080/vnc_auto.html

- name: configure the location of the image service API
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [glance]
      api_servers = http://controller:9292

- name: configure the lock path
  copy:
```

```
    dest: /etc/nova/nova.conf
    content: |
      [solo_currency]
      lock_path = /var/lib/nova/tmp

- name: configure the placement API
  copy:
    dest: /etc/nova/nova.conf
    content: |
      [placement]
      region_name = RegionOne
      project_domain_name = Default
      project_name = service
      auth_type = password
      user_domain_name = Default
      auth_url = http://controller:5000/v3
      username = placement
      password = 1234

- name: configuring to make the computer node to support hardware acceleration
  copy:
    dest: /etc/nova/nova-compute.conf
    content: |
      [libvirt]
      virt_type = qemu

- name: restarting the computer service
  shell: service nova-compute restart

- block:
  - name: Verifying if already running and active the nova-compute.
    shell: systemctl status nova-compute
    register: novacompute_service

  - debug:
      msg="{{ novacompute_service }}"
```

d. Create different plays in installing per server type (controller, compute etc.) and identify it as a group in the Inventory file.

```
 GNU nano 6.2                          ansible.cfg *

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings= False

remote_user = pulao
private_key_file = ~/.ssh/
```

pulao@pulao-VirtualBox: ~/ansible/Activity-14

```
 GNU nano 6.2                                inventory *

[controller_node]
192.168.56.138
```

pulao@pulao-VirtualBox: ~/ansible/Activity-14

```
 GNU nano 6.2                                main.yml *
---

- hosts: all
  become: true
  pre_tasks:

  - name: Dpkg fixing in Ubuntu Servers
    shell: |
      dpkg --configure -a
    when: ansible_distribution == "Ubuntu"

  - name: Update and Upgrade remote in Ubuntu servers
    apt:
      update_cache: yes
      upgrade: yes
    when: ansible_distribution == "Ubuntu"

- hosts: controller_node
  become: true
  roles:
    - keystone
    - glance
    - nova
```

e. Add, commit and push it to your GitHub repo.

```
pulao@pulao-VirtualBox:~/ansible/Activity-14$ git commit -m "Activity-14"
[main c41ffb3] Activity-14
 11 files changed, 338 insertions(+)
 create mode 100644 .inventory.swp
 create mode 100644 .main.yml.swp
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 main.yml
 create mode 100644 roles/glance/tasks/.main.yml.swp
 create mode 100644 roles/glance/tasks/main.yml
 create mode 100644 roles/keystone/tasks/.main.yml.swp
 create mode 100644 roles/keystone/tasks/main.yml
 create mode 100644 roles/nova/tasks/.main.yml.swp
 create mode 100644 roles/nova/tasks/main.yml
pulao@pulao-VirtualBox:~/ansible/Activity-14$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (18/18), 4.23 KiB | 228.00 KiB/s, done.
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:MaxinePulao/Activity-14.git
   6dca7b9..c41ffb3  main -> main
```

5. **Output** (screenshots and explanations)

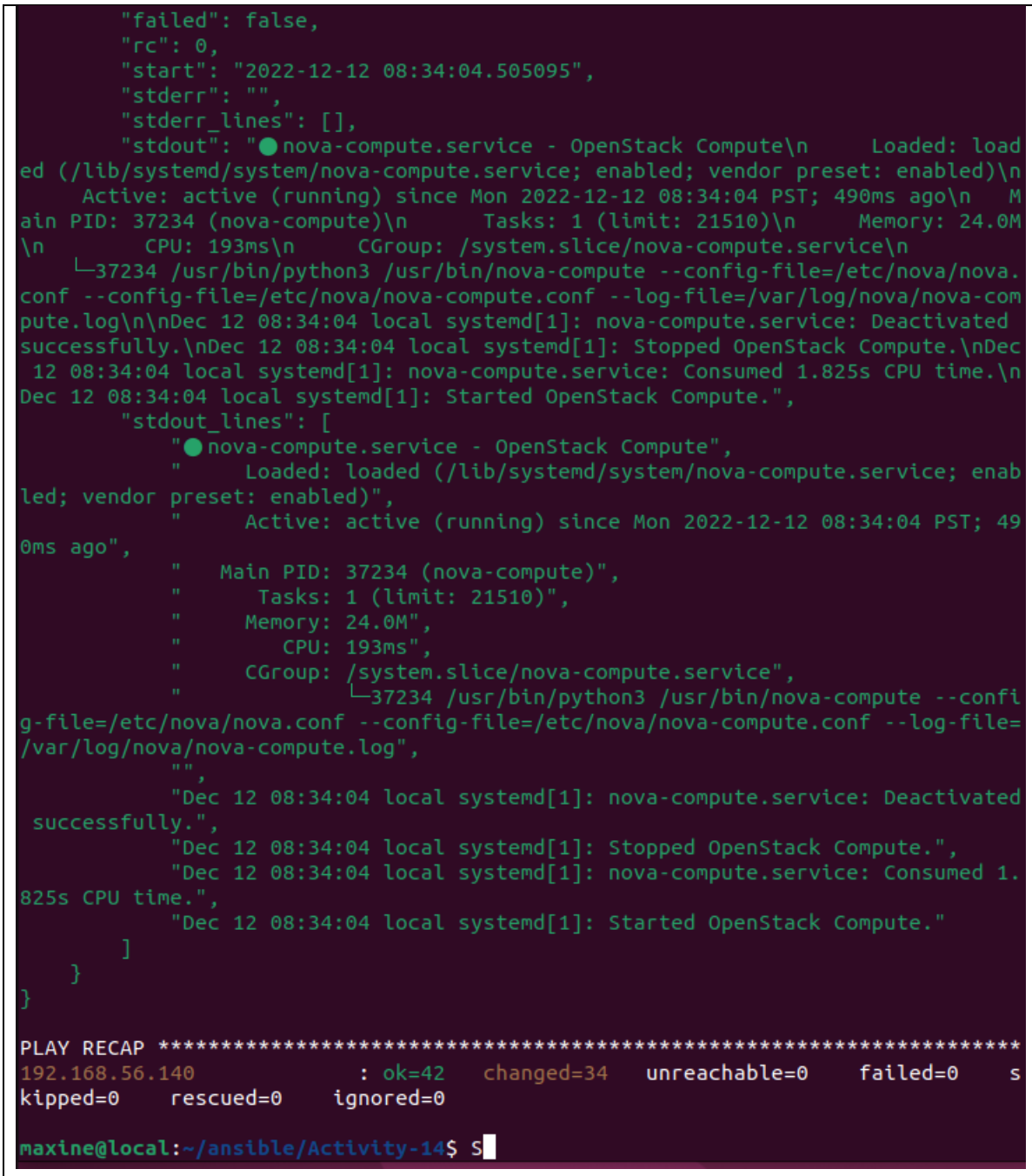
```
maxine@local:~/ansible/Activity-14$ ansible-playbook --ask-become-pass main.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.140]

TASK [Dpkg fixing in Ubuntu Servers] *******************************************
changed: [192.168.56.140]

TASK [Update and Upgrade remote in Ubuntu servers] *****************************
[WARNING]: The value "True" (type bool) was converted to "'True'" (type
string). If this does not look like what you expect, quote the entire value to
ensure it does not change.
ok: [192.168.56.140]

PLAY [controller_node] *********************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.140]

TASK [keystone : Install the prerequisites for mod_wsgi] ***********************
changed: [192.168.56.140]

TASK [keystone : copy] *********************************************************
changed: [192.168.56.140]

TASK [keystone : Create a seperate apache config to serve our python script over
 HTTP] ***
changed: [192.168.56.140]

TASK [keystone : install the keystone package] *********************************
changed: [192.168.56.140]

TASK [keystone : initializing the fernet repositories (1)] *********************
changed: [192.168.56.140]

TASK [keystone : initializing the fernet repositories (2)] *********************
changed: [192.168.56.140]

TASK [keystone : configure apache http server] *********************************
changed: [192.168.56.140]
```

```
TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables (1)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables (2)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables (3)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables(4)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables(5)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables(6)] ***
changed: [192.168.56.140]

TASK [keystone : configuring administrative account by setting the proper enviro
nmental variables (7)] ***
changed: [192.168.56.140]

TASK [keystone : Verifying if already active and running the keystone.] ********
changed: [192.168.56.140]

TASK [keystone : debug] ********************************************************
ok: [192.168.56.140] => {
    "msg": {
        "changed": true,
        "cmd": "keystone-manage --help",
        "delta": "0:00:01.942779",
        "end": "2022-12-12 08:33:44.053639",
        "failed": false,
        "rc": 0,
        "start": "2022-12-12 08:33:42.110860",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "usage: keystone-manage [bootstrap|credential_migrate|credenti
al_rotate|credential_setup|db_sync|db_version|doctor|domain_config_upload|fernet
```

```
TASK [glance : install and configure components of glance] *********************
changed: [192.168.56.140]

TASK [glance : configuring database access] ************************************
changed: [192.168.56.140]

TASK [glance : configuring the local file system store and location of image fil
es] ***
changed: [192.168.56.140]

TASK [glance : configuring the access to keystone] *****************************
changed: [192.168.56.140]

TASK [glance : enable per-tenant quotas] ***************************************
changed: [192.168.56.140]

TASK [glance : restart the image services] *************************************
[WARNING]: Consider using the service module rather than running 'service'.  If
you need to use command because service is insufficient you can add 'warn:
false' to this command task or set 'command_warnings=False' in ansible.cfg to
get rid of this message.
changed: [192.168.56.140]

TASK [glance : Verifying if already installed Glance.] *************************
changed: [192.168.56.140]

TASK [glance : debug] **********************************************************
ok: [192.168.56.140] => {
    "msg": {
        "changed": true,
        "cmd": "glance --version",
        "delta": "0:00:01.175175",
        "end": "2022-12-12 08:33:57.905256",
        "failed": false,
        "rc": 0,
        "start": "2022-12-12 08:33:56.730081",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "3.6.0",
        "stdout_lines": [
            "3.6.0"
        ]
    }
}

TASK [glance : Verifying if already active and running the Glance.] ************
```

```
TASK [glance : Verifying if already active and running the Glance.] ************
changed: [192.168.56.140]

TASK [glance : debug] **********************************************************
ok: [192.168.56.140] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl status glance-api",
        "delta": "0:00:00.008760",
        "end": "2022-12-12 08:33:58.470320",
        "failed": false,
        "rc": 0,
        "start": "2022-12-12 08:33:58.461560",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "● glance-api.service - OpenStack Image Service API\n     Load
ed: loaded (/lib/systemd/system/glance-api.service; enabled; vendor preset: enab
led)\n     Active: active (running) since Mon 2022-12-12 08:33:56 PST; 2s ago\n
      Docs: man:glance-api(1)\n   Main PID: 36641 (glance-api)\n      Tasks: 1 (
limit: 21510)\n     Memory: 74.3M\n        CPU: 964ms\n     CGroup: /system.slic
e/glance-api.service\n             └─36641 /usr/bin/python3 /usr/bin/glance-api
--config-file=/etc/glance/glance-api.conf --config-dir=/etc/glance/ --log-file=/
var/log/glance/glance-api.log\n\nDec 12 08:33:56 local systemd[1]: Started OpenS
tack Image Service API.",
        "stdout_lines": [
            "● glance-api.service - OpenStack Image Service API",
            "     Loaded: loaded (/lib/systemd/system/glance-api.service; enable
d; vendor preset: enabled)",
            "     Active: active (running) since Mon 2022-12-12 08:33:56 PST; 2s
 ago",
            "       Docs: man:glance-api(1)",
            "   Main PID: 36641 (glance-api)",
            "      Tasks: 1 (limit: 21510)",
            "     Memory: 74.3M",
            "        CPU: 964ms",
            "     CGroup: /system.slice/glance-api.service",
            "             └─36641 /usr/bin/python3 /usr/bin/glance-api --config-
file=/etc/glance/glance-api.conf --config-dir=/etc/glance/ --log-file=/var/log/g
lance/glance-api.log",
            "",
            "Dec 12 08:33:56 local systemd[1]: Started OpenStack Image Service A
PI."
        ]
    }
}
```

```
TASK [nova : configuring RabbitMQ message queue access] ************************
changed: [192.168.56.140]

TASK [nova : configuring identity service access (1)] **************************
changed: [192.168.56.140]

TASK [nova : configuring identity service access (2)] **************************
changed: [192.168.56.140]

TASK [nova : enable and configure remote console access] ***********************
changed: [192.168.56.140]

TASK [nova : configure the location of the image service API] ******************
changed: [192.168.56.140]

TASK [nova : configure the lock path] ******************************************
changed: [192.168.56.140]

TASK [nova : configure the placement API] **************************************
changed: [192.168.56.140]

TASK [nova : configuring to make the computer node to support hardware accelerat
ion] ***
changed: [192.168.56.140]

TASK [nova : restarting the computer service] **********************************
changed: [192.168.56.140]

TASK [nova : Verifying if already running and active the nova-compute.] ********
changed: [192.168.56.140]

TASK [nova : debug] ************************************************************
ok: [192.168.56.140] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl status nova-compute",
        "delta": "0:00:00.010904",
        "end": "2022-12-12 08:34:04.515999",
        "failed": false,
        "rc": 0,
        "start": "2022-12-12 08:34:04.505095",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "● nova-compute.service - OpenStack Compute\n     Loaded: load
```



Reflections:

Answer the following:

1. Describe Keystone, Glance and Nova services

   Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers), which is used to host and manage cloud computing systems. This is a service that provides resizable compute capacity in OpenStack. Glance is

an image service that allows users to discover, retrieve, and register VM (virtual machine) images and container images, which can use Swift or Ceph as its actual storage backend. And lastly, Keystone is the dashboard in OpenStack, which is an interface to the client. Network services are provided by neurons

Conclusions:

In this activity, I have mastered on how to create a playbook to install and update my ubuntu server and openstack packages. This activity also showed me what could services are and what is their purpose. However, it may have some issues regarding on security since some of it is public and accessible. To keep their cloud environment secure, the majority of cloud service providers implement pertinent security standards and industry certifications. However, storing data and business-critical files in virtual data centres can potentially open you up to risks. Common risks are data loss and theft. This activity also taught me the components and what makes up a cloud service and its purpose to my servers. There are four cloud deployment models. Public, Private, Community, and Hybrid. Each deployment model is defined by where the environment's infrastructure resides. There are three main cloud service models: Software as a Service, Platform as a Service, Infrastructure as a Service. After performing this activity, I hope I could be more comfortable and trylu master managing servers like Ubuntu and CentOS.