Name: Dalena, Justin Miguel S.	Date Performed: Dec. 6, 2024
Course/Section: CPE31S21	Date Submitted: Dec. 6, 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 2024-2025

<div align="center">

Activity 13: OpenStack Prerequisite Installation

</div>

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.
2. Create a playbook that converts the steps in the following items in https://docs.openstack.org/install-guide/
   a. NTP
   b. OpenStack packages
   c. SQL Database
   d. Message Queue
   e. Memcached

   f. Etcd

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

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

5. Output (screenshots and explanations)

Created a new repository for my activity

Activity_13 Public   Pin   Unwatch 1

main ▾   1 Branch   0 Tags   Go to file   t   Add file ▾   <> Code ▾

Justin-Dalena Initial commit   4bb5fdd · now   1 Commit

README.md   Initial commit   now

README   ✏

Activity_13

Checking the connections of my control node to my manage node (server 1)

```
justin@Workstation:~/Activity_13$ ansible all -m ping -i inventory
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host Server1 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
Server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

Created an ansible files named script.yml to install packages to remote server

```
  GNU nano 4.8                                              script.yml
---
- name: Installer Script
  hosts: all
  become: true
  tasks:
    - name: Installation of NTP Ubuntu
      apt:
       name: chrony
       state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Start NTP
      service:
       name: chronyd
       state: restarted
       enabled: true

    - name: Installation of OpenStack packages
      apt:
       name: python3-openstackclient
       state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Installation of Database
      apt:
       name: mariadb-server
       state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Installation of Message Queue
      apt:
       name: rabbitmq-server
       state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Start Message Queue
      service:
        name: rabbitmq-server
        state: restarted
        enabled: true
```

```yaml
    when: ansible_distribution == "Ubuntu"

  - name: Start Message Queue
    service:
      name: rabbitmq-server
      state: restarted
      enabled: true

  - name: Installation of Memcached
    apt:
     name: memcached
     state: latest
    when: ansible_distribution == "Ubuntu"

  - name: Start Memcached
    service:
     name: memcached
     state: restarted
     enabled: true

  - name: Installation of Etcd
    apt:
     name: etcd-server
     state: latest
    when: ansible_distribution == "Ubuntu"

  - name: Start Etcd
    service:
     name: etcd
     state: restarted
     enabled: true
```

Ran the ansible playbook to install all the packages in remote server

```
TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host Server1 should use /usr/bin/python3, but is using /usr/bin/python
for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered
platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for
more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
ok: [Server1]

TASK [Installation of NTP Ubuntu] *********************************************************
ok: [Server1]

TASK [Start NTP] *********************************************************
changed: [Server1]

TASK [Installation of OpenStack packages] *********************************************************
ok: [Server1]

TASK [Installation of Database] *********************************************************
ok: [Server1]

TASK [Installation of Message Queue] *********************************************************
ok: [Server1]

TASK [Start Message Queue] *********************************************************
changed: [Server1]

TASK [Installation of Memcached] *********************************************************
ok: [Server1]

TASK [Start Memcached] *********************************************************
changed: [Server1]

TASK [Installation of Etcd] *********************************************************
ok: [Server1]

TASK [Start Etcd] *********************************************************
changed: [Server1]

PLAY RECAP *********************************************************
Server1                    : ok=11   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

justin@Workstation:~/Activity_13$ S
```

Checking of all the packages if they are installed properly

```
justin@Server1:~$ dpkg -l | grep mysql
ii  libdbd-mysql-perl                        4.046-1
                amd64        Perl5 database interface to the MariaDB/MySQL data
base
ii  libmysqlclient20:amd64                   5.7.42-0ubuntu0.18.04.1
                amd64        MySQL database client library
ii  mysql-common                             5.8+1.0.4
                all          MySQL database common files, e.g. /etc/mysql/my.cn
f
justin@Server1:~$ dpkg -l | grep memcached
ii  memcached                                1.5.6-0ubuntu1.2
                amd64        high-performance memory object caching system
justin@Server1:~$ dpkg -l | grep etcd
ii  etcd-server                              3.2.17+dfsg-1ubuntu0.1
                amd64        highly-available key value store -- daemon
justin@Server1:~$ dpkg -l | grep rabbitmq-server
ii  rabbitmq-server                          3.6.10-1ubuntu0.5
                all          AMQP server written in Erlang
justin@Server1:~$ openstack --version
openstack 3.14.2
```

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?

   Using OpenStack gives you flexibility and command over your company's cloud.
   infrastructure. Because it facilitates scalability, businesses can grow.
   resources as needed at a reasonable price. Because OpenStack is open-source
   encourages creativity and removes vendor lock-in, allowing users the flexibility to
   personalize their surroundings. It also blends very nicely with a variety of
   tools and technologies, which makes it appropriate for a range of IT requirements.
   It's in use.
   Additionally, the community guarantees ongoing support and upgrades.

Conclusions:
I installed all required packages after finishing the task to set
I used an Ansible playbook to set up OpenStack on my Ubuntu server. The procedure
included setting up dependencies and ensuring that everything was set up correctly.
in line with OpenStack's functionality. Ansible's use simplified the installation process and
more efficient in terms of time and effort. This encounter improved me.
know how to properly use automation tools and install OpenStack.