Name: Buduan, Christian Aaron C.	Date Performed: 12/03/2024
Course/Section: CPE31S2	Date Submitted: 12/023/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st sem 2024-2025

Activity 13: OpenStack Prerequisite Installation

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

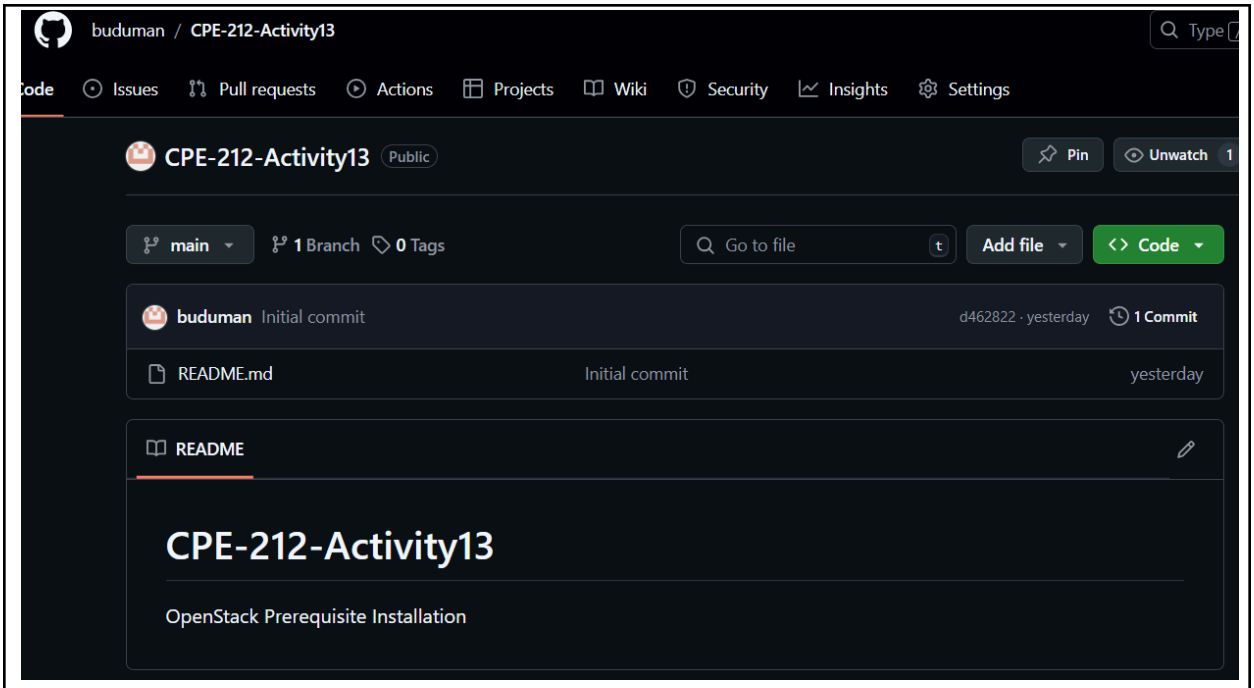
Oracle VirtualBox (Hypervisor)

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)



Create a new repository

```
qcacbuduan@Workstation:~$ git clone git@github.com:buduman/CPE-212-Activity13.git
Cloning into 'CPE-212-Activity13'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

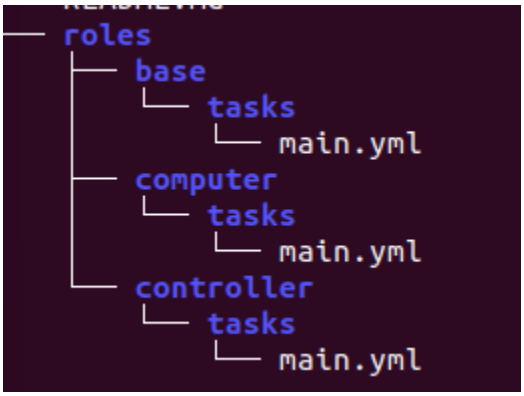
clone the created repository into your workstation

```
qcacbuduan@Workstation:~/CPE-212-Activity13$ cat inventory
[controller]
server1

[computer]
centosbuduan
qcacbuduan@Workstation:~/CPE-212-Activity13$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = qcacbuduan
host_key_checking = True
private_key_file = ~/.ssh/ansible
deprecation_warnings=False
```

create inventory file and ansible.cfg file with the following configurations

```
qcacbuduan@Workstation:~/CPE-212-Activity13$ cat main.yml
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    -  base

- hosts: controller
  become: true
  roles:
    - controller

- hosts: computer
  become: true
  roles:
    - computer
```

Create playbook that will install updates on your servers and implement roles such as controller and computer



create directory called roles and inside the roles directory, create the following directories and files.

```
qcacbuduan@Workstation:~/CPE-212-Activity13$ cat roles/base/tasks/main.yml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```

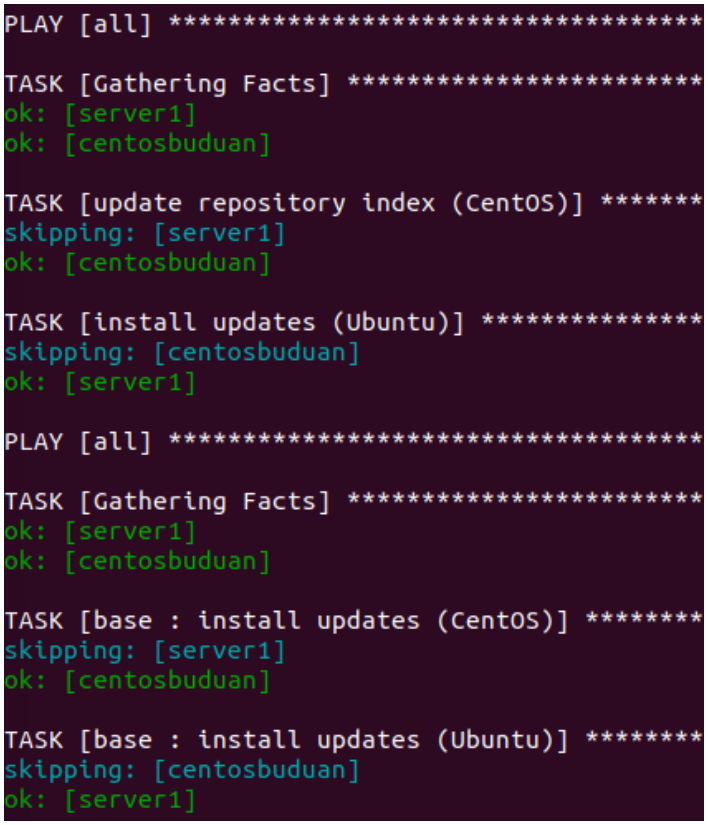
Under the base directory, create a playbook that will make sure the update repository is updated.

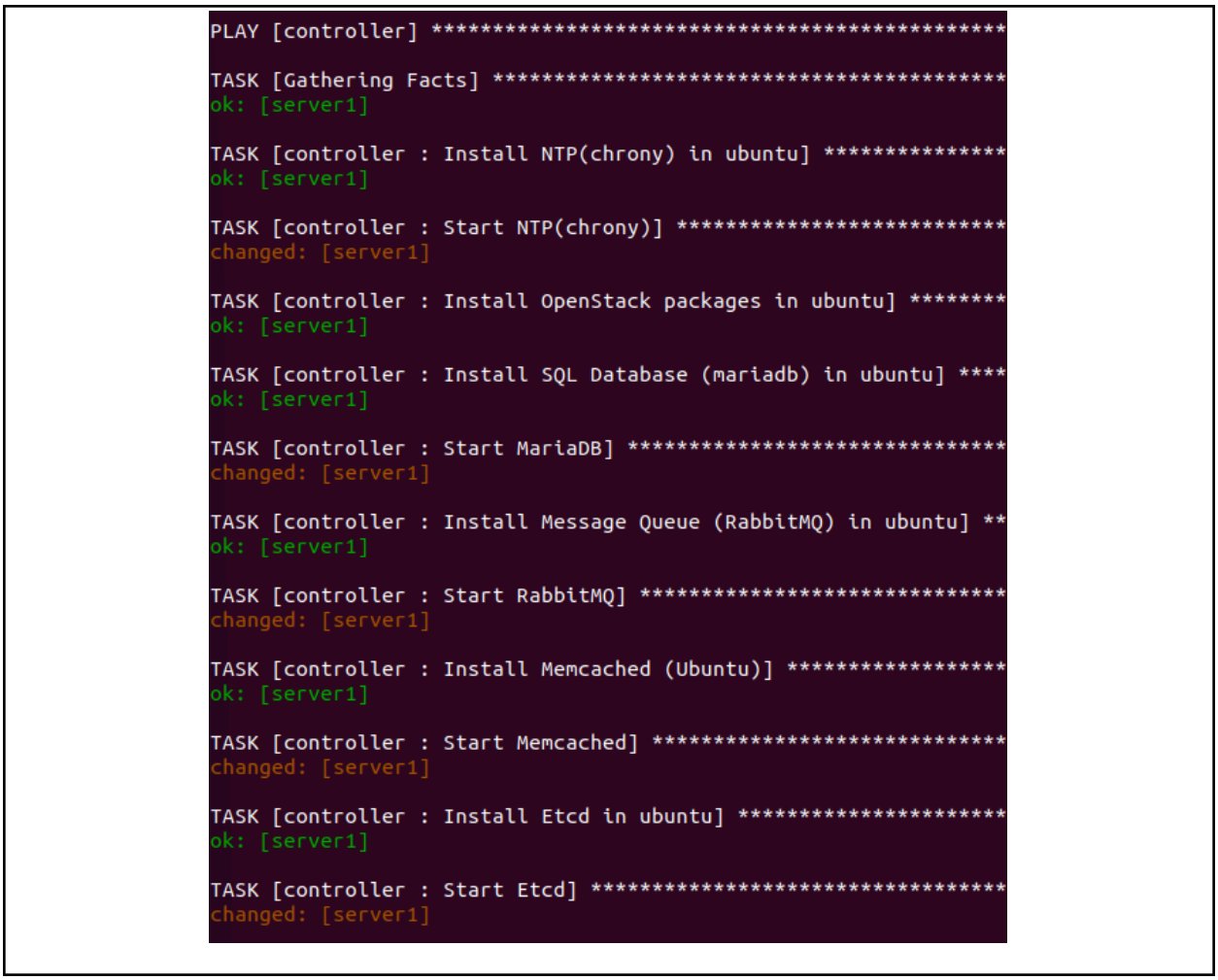
```
qcacbuduan@Workstation:~/CPE-212-Activity13$ cat roles/controller/tasks/main.yml
---
- name: Install NTP(chrony) in ubuntu
  tags: web, ubuntu, ntp, chrony
  apt:
    name: chrony
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start NTP(chrony)
  tags: web, ntp, chrony
  service:
    name: chronyd
    state: restarted
    enabled: true

- name: Install OpenStack packages in ubuntu
  tags: web, ubuntu, openstack
  apt:
    name: python3-openstackclient
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Install SQL Database (mariadb) in ubuntu
  tags: web, ubuntu, mariadb
  apt:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start MariaDB
  tags: web, mariadb
  service:
    name: mariadb
    state: restarted
    enabled: true
```

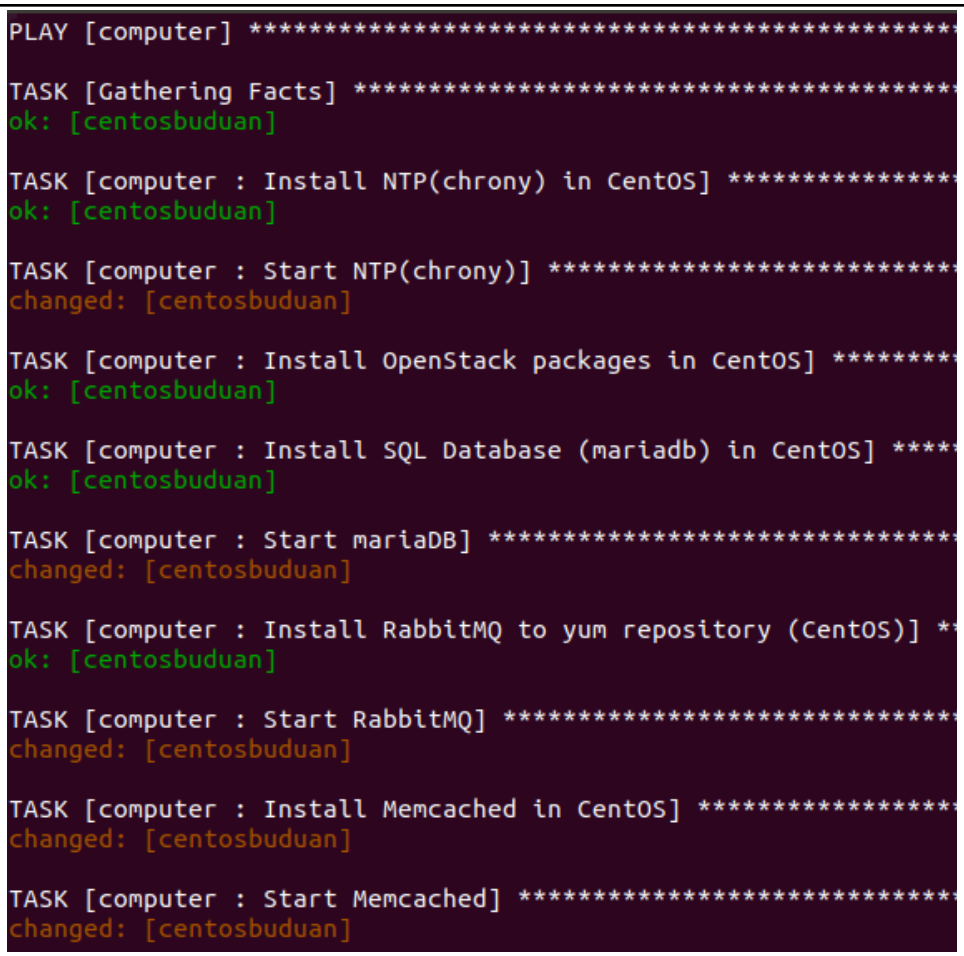
```
- name: Install Message Queue (RabbitMQ) in ubuntu
  tags: web, ubuntu, rabbitmq
  apt:
    name: rabbitmq-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start RabbitMQ
  tags: web, rabbitmq
  service:
    name: rabbitmq-server
    state: restarted
    enabled: true

- name: Install Memcached (Ubuntu)
  tags: web, ubuntu, memcached
  apt:
    name: memcached
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start Memcached
  tags: web, memcached
  service:
    name: memcached
    state: restarted
    enabled: true

- name: Install Etcd in ubuntu
  tags: web, ubuntu, etcd
  apt:
    name: etcd-server
    state: latest
  when: ansible_distribution == "Ubuntu"

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

under the controller directory, create a playbook that will install the required items in Ubuntu server.

under the computer directory, create a playbook that will install the required items in CentOS




PLAY [controller] **
TASK [Gathering Facts] **
ok: [server1]
TASK [controller : Install NTP(chrony) in ubuntu] ***************
ok: [server1]
TASK [controller : Start NTP(chrony)] ***************************
changed: [server1]
TASK [controller : Install OpenStack packages in ubuntu] ********
ok: [server1]
TASK [controller : Install SQL Database (mariadb) in ubuntu] ****
ok: [server1]
TASK [controller : Start MariaDB] *******************************
changed: [server1]
TASK [controller : Install Message Queue (RabbitMQ) in ubuntu] **
ok: [server1]
TASK [controller : Start RabbitMQ] ******************************
changed: [server1]
TASK [controller : Install Memcached (Ubuntu)] ******************
ok: [server1]
TASK [controller : Start Memcached] *****************************
changed: [server1]
TASK [controller : Install Etcd in ubuntu] **********************
ok: [server1]
TASK [controller : Start Etcd] **********************************
changed: [server1]



run the main.yml in the main directory

```
qcacbuduan@Workstation:~/CPE-212-Activity13$ git add --all
qcacbuduan@Workstation:~/CPE-212-Activity13$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   ansible.cfg
        new file:   inventory
        new file:   main.yml
        new file:   roles/base/tasks/main.yml
        new file:   roles/computer/tasks/main.yml
        new file:   roles/controller/tasks/main.yml

qcacbuduan@Workstation:~/CPE-212-Activity13$ git commit -m "hoa13"
[main d5fb57d] hoa13
 6 files changed, 215 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 main.yml
 create mode 100644 roles/base/tasks/main.yml
 create mode 100644 roles/computer/tasks/main.yml
 create mode 100644 roles/controller/tasks/main.yml
qcacbuduan@Workstation:~/CPE-212-Activity13$ git push origin main
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (15/15), 2.03 KiB | 2.03 MiB/s, done.
Total 15 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:buduman/CPE-212-Activity13.git
   d462822..d5fb57d  main -> main
```

push all files into the github repository

https://github.com/buduman/CPE-212-Activity13.git

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?
   - The benefits of implementing Openstack into your servers is that it is an open-source technology, especially allowing businesses and other enterprises to avoid the high license fees associated with proprietary cloud solutions. It provides flexibility and scalability, allowing organizations to flexibly alter their computing capacity according to demand, which is critical for dealing with changing workloads. OpenStack supports a wide range of technologies and connections, allowing for smooth collaboration and data sharing across several platforms.

Conclusions:

- In this activity, I was able to learn how to structure Ansible playbooks to automate the installation of essential OpenStack components on different server types. By organizing the configuration in an inventory file and separating the tasks by play, this exercise helped me review the importance of systematic approaches in DevOps and cloud infrastructure management.