Group 3: Domondon, Mark Stefan P. John Edward Miles Espiritu Ruffer Laus	**Date Performed:** **April 23, 2024**
Course/Section: CPE232-CPE31S1	**Date Submitted: April 30, 2024**
Instructor: Dr. Jonathan V. Taylar	**Semester and SY: 2nd sem 2023-2024**

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)

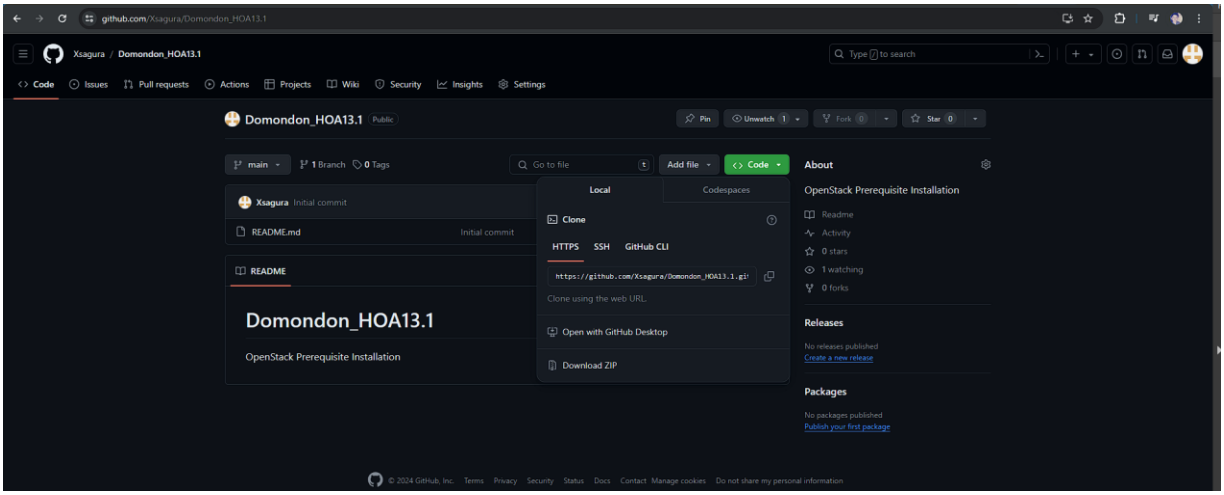
1x Ubuntu VM or Centos VM

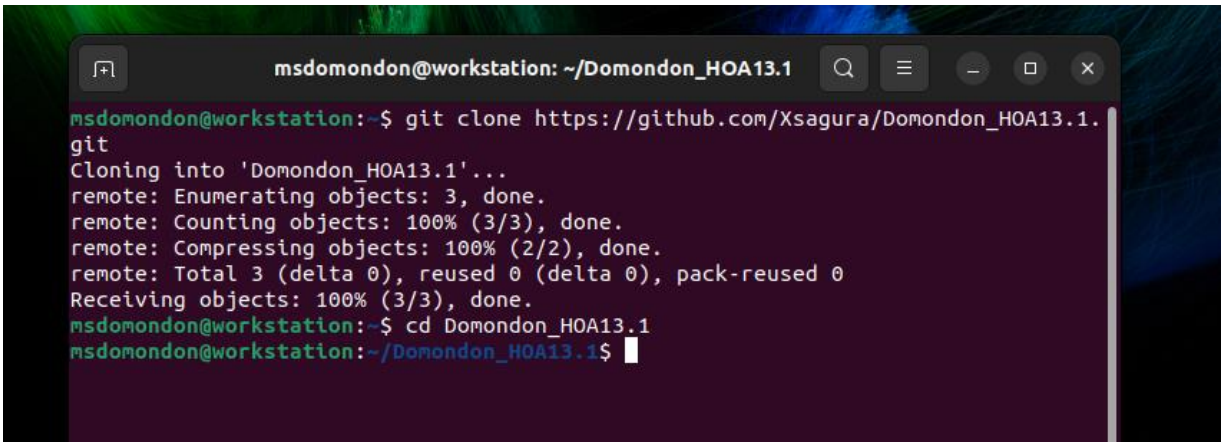
4. Tasks

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.

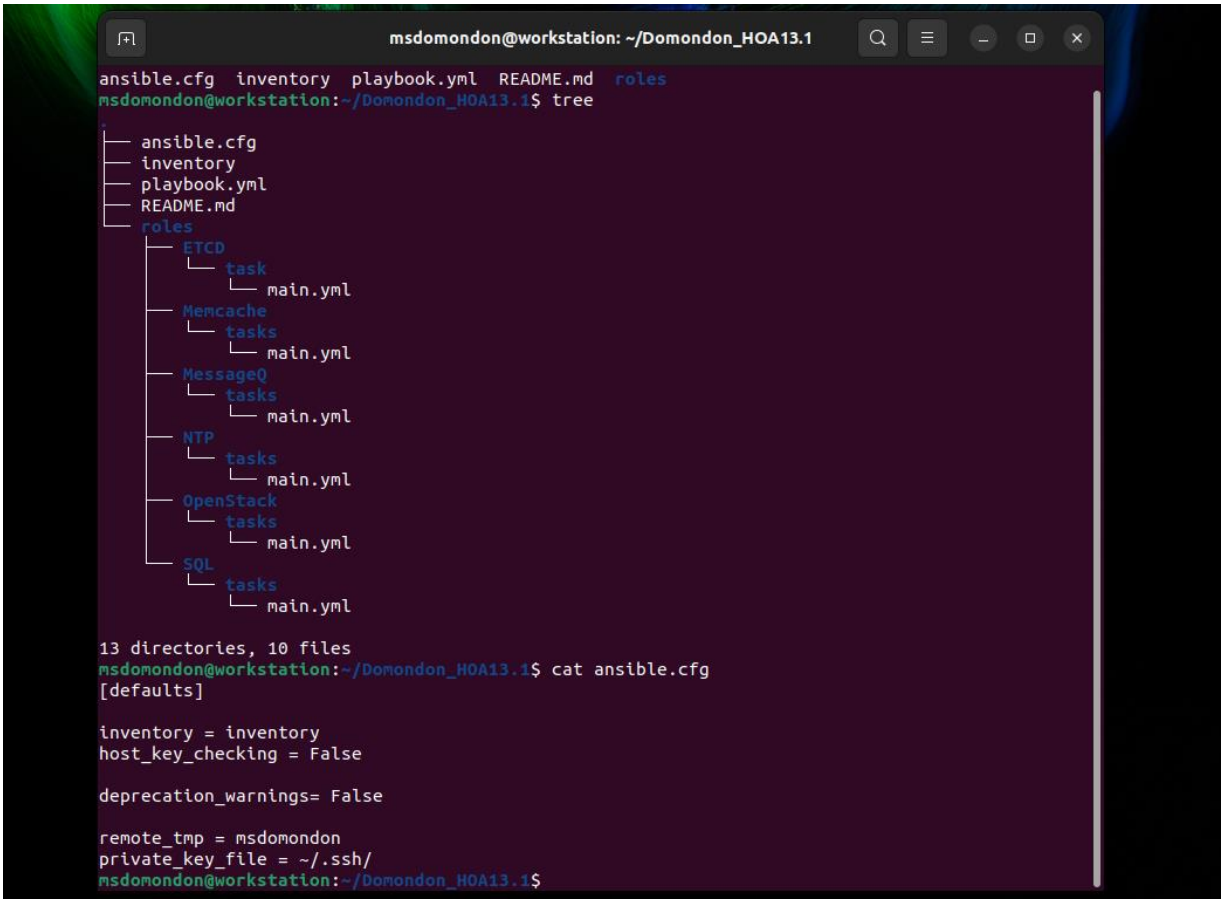
5. Output (screenshots and explanations)

- **Creating the repository for the activity**





- I included the necessary functions for the tasks, applied for the playbook.

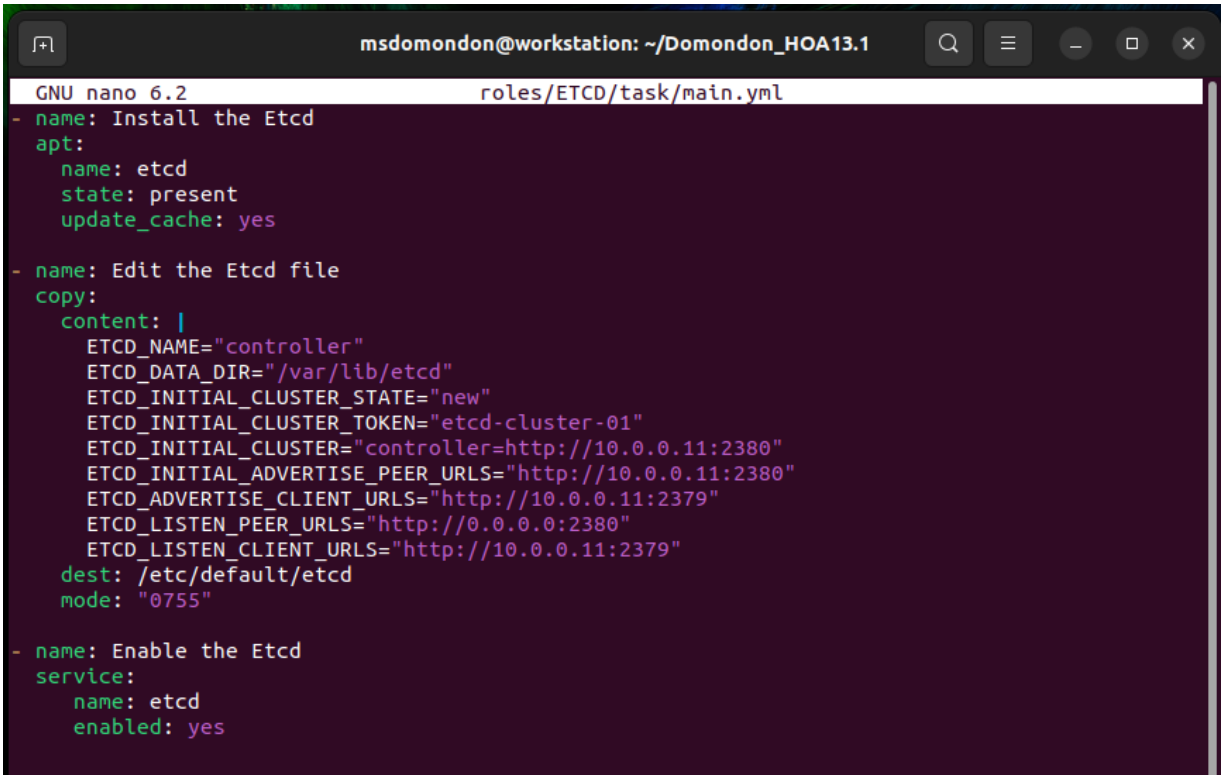


- Calling parts

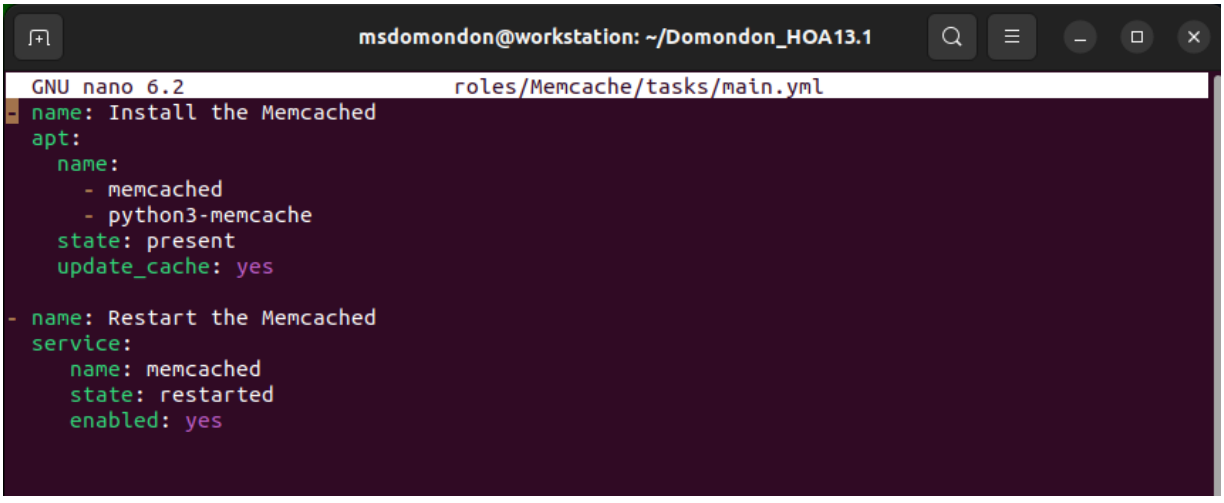
```
msdomondon@workstation:~/Domondon_HOA13.1$ cat inventory
[Ubuntu]
192.168.56.108

msdomondon@workstation:~/Domondon_HOA13.1$ cat playbook.yml
---
- hosts: all
  become: true
  pre_tasks:

  - name: Install updates (Ubuntu)
    apt:
      upgrade: dist
      update_cache: yes
    changed_when: false

- hosts: Ubuntu
  become: true
  roles:
    - NTP
    - OpenStack
    - SQL
    - MessageQ
    - Memcache
    - ETCD

msdomondon@workstation:~/Domondon_HOA13.1$
```

```
msdomondon@workstation:~/Domondon_HOA13.1$ tree roles
roles
├── ETCD
│   └── task
│       └── main.yml
├── Memcache
│   └── tasks
│       └── main.yml
├── MessageQ
│   └── tasks
│       └── main.yml
├── NTP
│   └── tasks
│       └── main.yml
├── OpenStack
│   └── tasks
│       └── main.yml
└── SQL
    └── tasks
        └── main.yml

12 directories, 6 files
msdomondon@workstation:~/Domondon_HOA13.1$
```

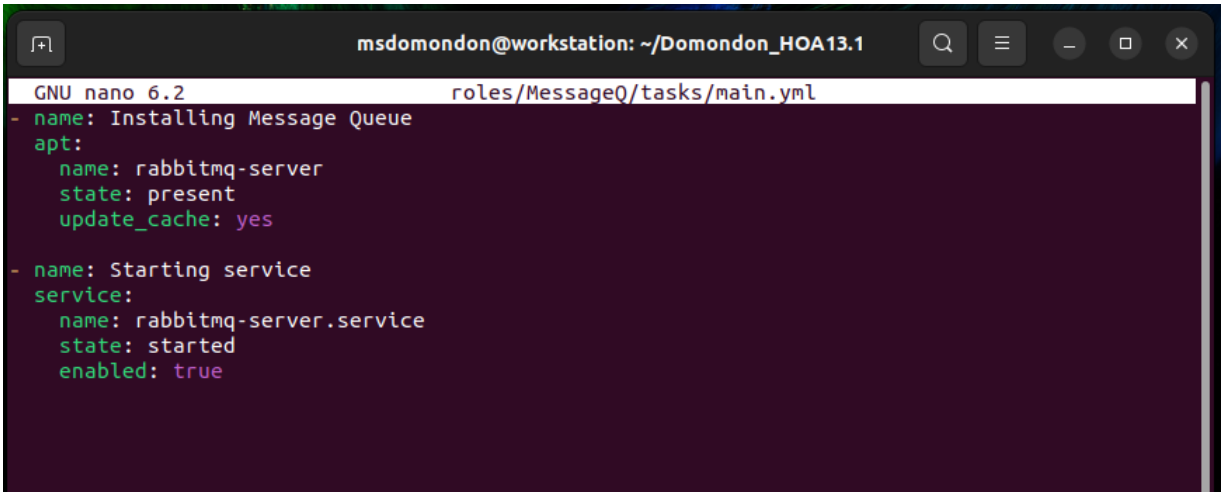
- Inside the roles main.yml of “ETCD”



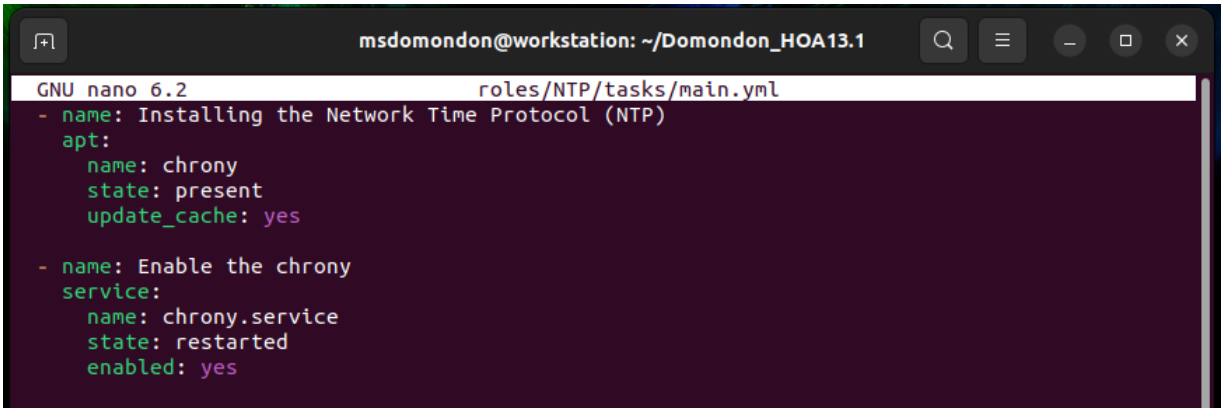
- Inside the roles main.yml of “Memcache”



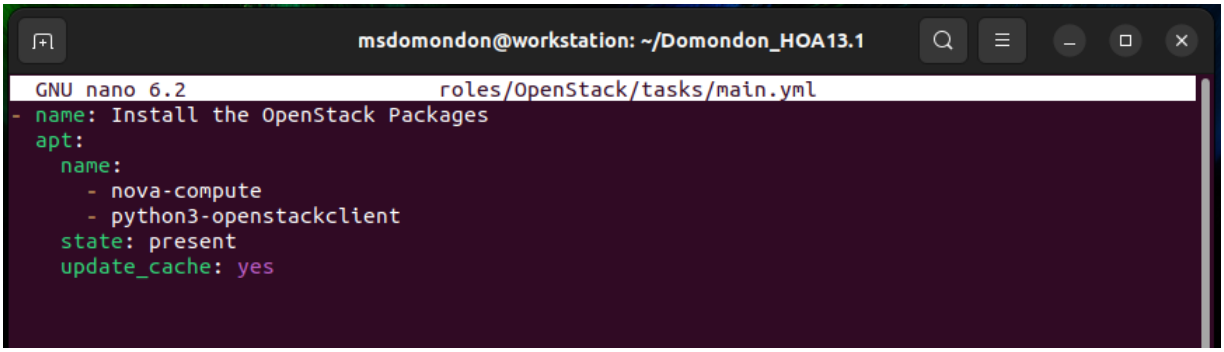
- Inside the roles main.yml of “MessageQ”



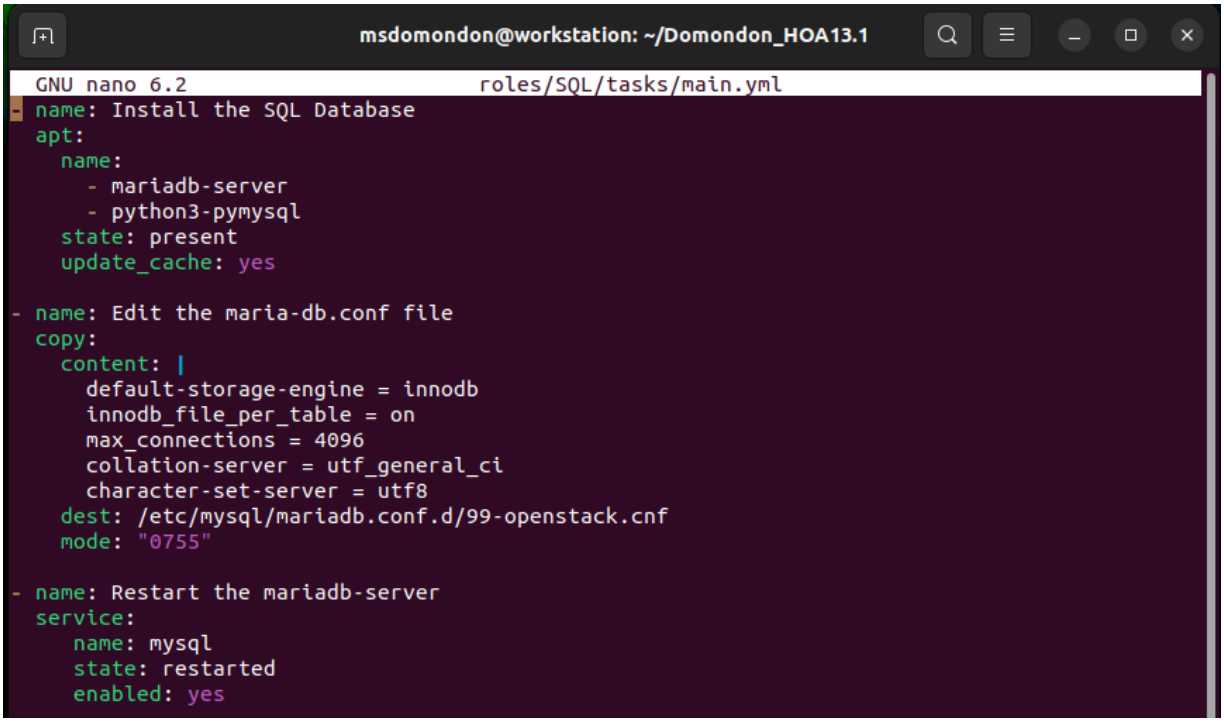
- Inside the roles main.yml of “NTP”



- Inside the roles main.yml of “OpenStack”

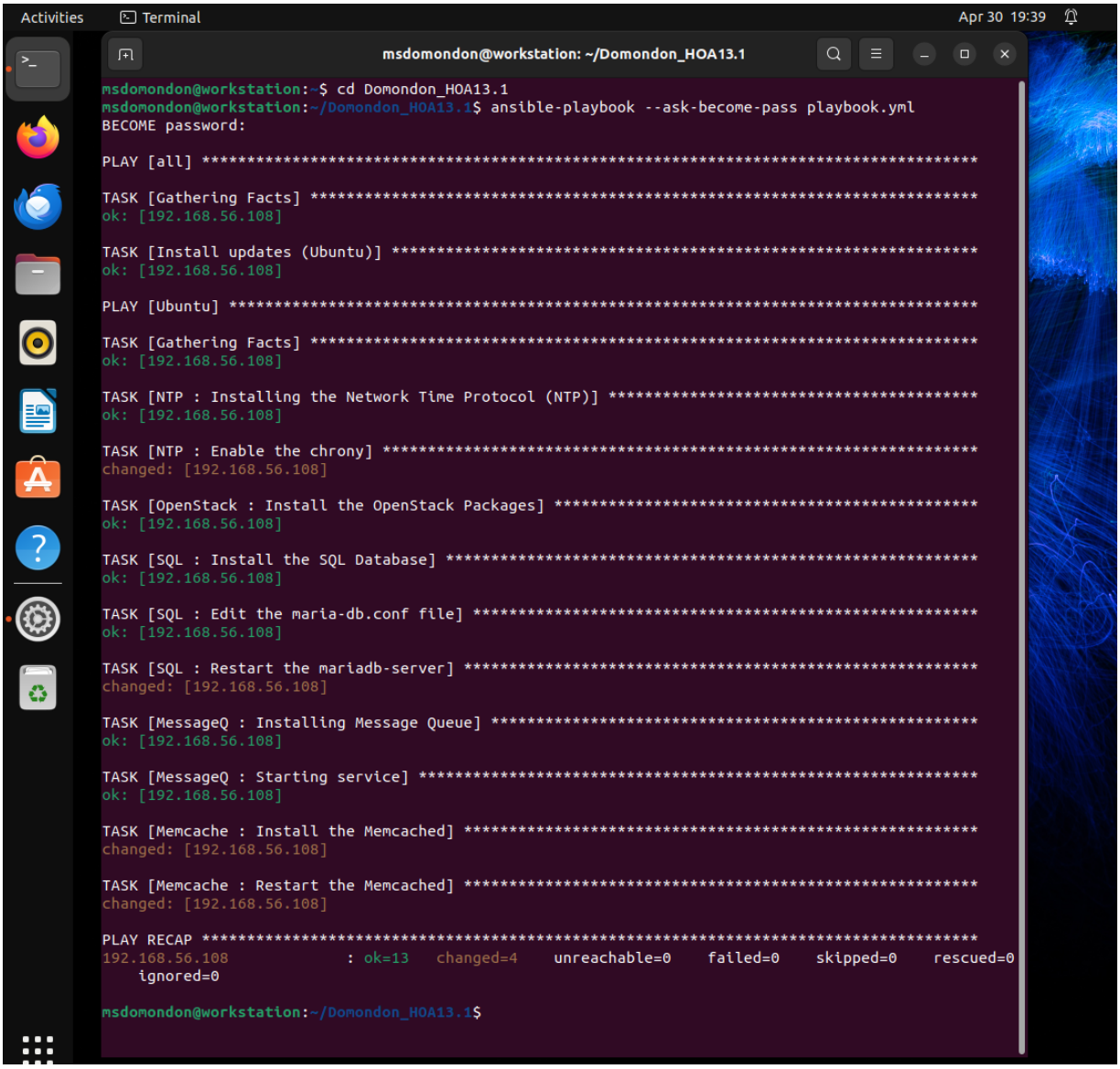


- Inside the roles main.yml of “SQL”



Running the playbook

Note: the functions of each roles are designed to be installed properly allocated size is (5GB)



```
TASK [ETCD : Install the Etcd] ************************************************************
changed: [192.168.56.108]

TASK [ETCD : Edit the Etcd file] **********************************************************
changed: [192.168.56.108]

TASK [ETCD : Enable the Etcd] *************************************************************
ok: [192.168.56.108]

PLAY RECAP ********************************************************************************
192.168.56.108             : ok=16   changed=5    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0

msdomondon@workstation:~/Domondon_HOA13.1$
```

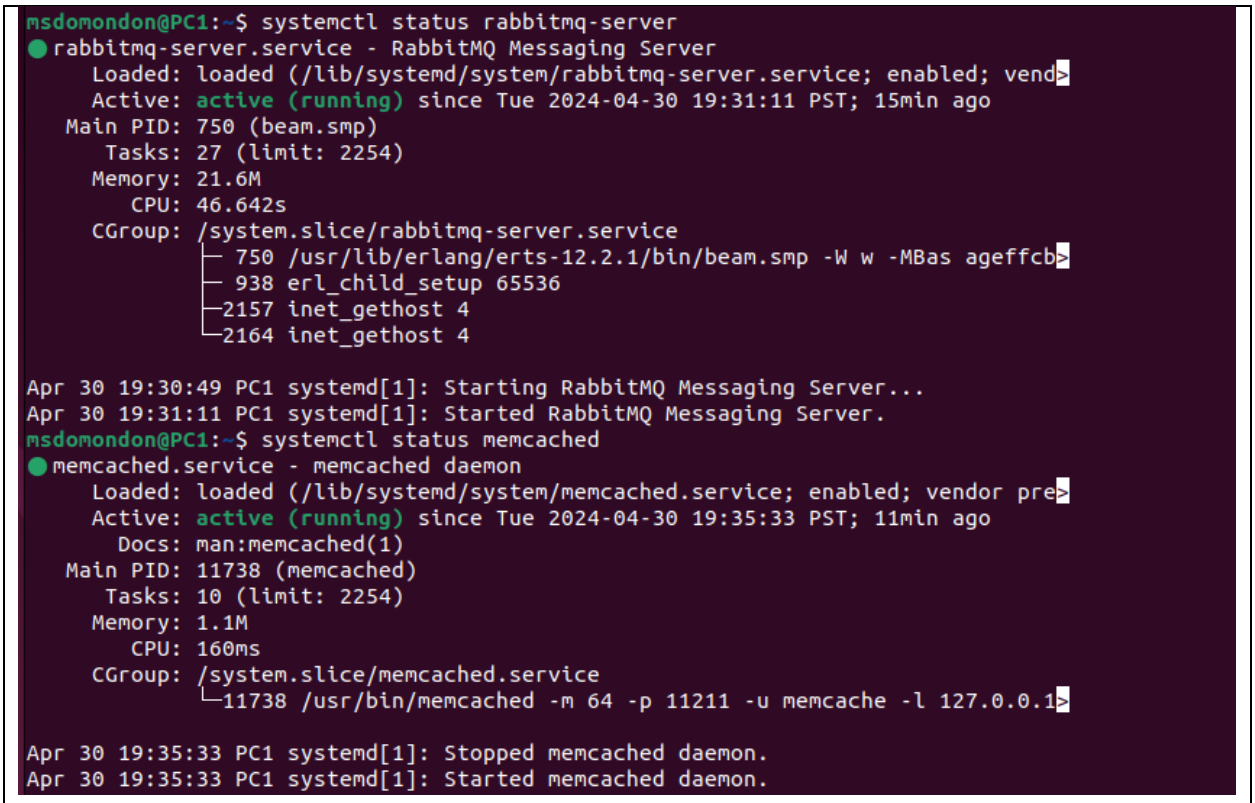
- **Verification of installations**

```
msdomondon@PC1:~$ systemctl status etcd
× etcd.service - etcd - highly-available key value store
     Loaded: loaded (/lib/systemd/system/etcd.service; enabled; vendor preset: >
     Active: failed (Result: exit-code) since Tue 2024-04-30 19:50:25 PST; 40s >
       Docs: https://etcd.io/docs
             man:etcd
    Process: 35468 ExecStart=/usr/bin/etcd $DAEMON_ARGS (code=exited, status=2)
   Main PID: 35468 (code=exited, status=2)
        CPU: 14ms

Apr 30 19:50:25 PC1 etcd[35468]:         github.com/coreos/etcd/etcdmain/etcd.g>
Apr 30 19:50:25 PC1 etcd[35468]: github.com/coreos/etcd/etcdmain.startEtcdOrPro>
Apr 30 19:50:25 PC1 etcd[35468]:         github.com/coreos/etcd/etcdmain/etcd.g>
Apr 30 19:50:25 PC1 etcd[35468]: github.com/coreos/etcd/etcdmain.Main()
Apr 30 19:50:25 PC1 etcd[35468]:         github.com/coreos/etcd/etcdmain/main.g>
Apr 30 19:50:25 PC1 etcd[35468]: main.main()
Apr 30 19:50:25 PC1 etcd[35468]:         github.com/coreos/etcd/main.go:28 +0x17
Apr 30 19:50:25 PC1 systemd[1]: etcd.service: Main process exited, code=exited,>
Apr 30 19:50:25 PC1 systemd[1]: etcd.service: Failed with result 'exit-code'.
Apr 30 19:50:25 PC1 systemd[1]: Failed to start etcd - highly-available key val>
lines 1-19/19 (END)
```

```
msdomondon@PC1:~$ systemctl status NTP
Unit NTP.service could not be found.
msdomondon@PC1:~$ systemctl status nova-compute
● nova-compute.service - OpenStack Compute
     Loaded: loaded (/lib/systemd/system/nova-compute.service; enabled; vendor >
     Active: active (running) since Tue 2024-04-30 19:31:11 PST; 14min ago
   Main PID: 2378 (nova-compute)
      Tasks: 2 (limit: 2254)
     Memory: 14.0M
        CPU: 5.286s
     CGroup: /system.slice/nova-compute.service
             └─2378 /usr/bin/python3 /usr/bin/nova-compute --config-file=/etc/n>

Apr 30 19:31:11 PC1 systemd[1]: Started OpenStack Compute.
Apr 30 19:31:13 PC1 nova-compute[2378]: Modules with known eventlet monkey patc>
msdomondon@PC1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.16 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese>
     Active: active (running) since Tue 2024-04-30 19:35:09 PST; 10min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 9761 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 2254)
     Memory: 12.7M
        CPU: 584ms
     CGroup: /system.slice/mariadb.service
             └─9761 /usr/sbin/mariadbd

Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] InnoDB: 10.6.1>
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] Plugin 'FEEDBA>
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] InnoDB: Loadin>
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Warning] You need to>
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] InnoDB: Buffer>
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] Server socket >
Apr 30 19:35:08 PC1 mariadbd[9761]: 2024-04-30 19:35:08 0 [Note] /usr/sbin/mari>
Apr 30 19:35:08 PC1 mariadbd[9761]: Version: '10.6.16-MariaDB-0ubuntu0.22.04.1'>
Apr 30 19:35:09 PC1 systemd[1]: Started MariaDB 10.6.16 database server.
Apr 30 19:35:09 PC1 /etc/mysql/debian-start[9813]: Triggering myisam-recover fo>
```

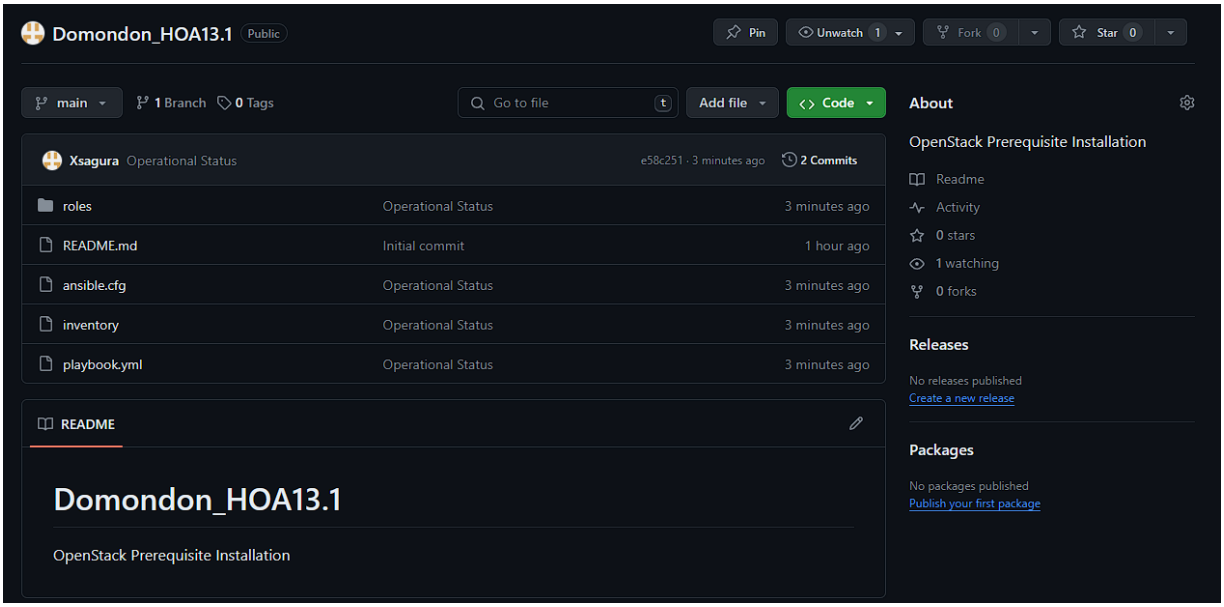
```
msdomondon@PC1:~$ sudo systemctl status chrony
● chrony.service - chrony, an NTP client/server
     Loaded: loaded (/lib/systemd/system/chrony.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-04-30 19:49:53 PST; 7min ago
       Docs: man:chronyd(8)
             man:chronyc(1)
             man:chrony.conf(5)
   Main PID: 31427 (chronyd)
      Tasks: 2 (limit: 2254)
     Memory: 1.2M
        CPU: 92ms
     CGroup: /system.slice/chrony.service
             ├─31427 /usr/sbin/chronyd -F 1
             └─31428 /usr/sbin/chronyd -F 1

Apr 30 19:49:53 PC1 systemd[1]: Starting chrony, an NTP client/server...
Apr 30 19:49:53 PC1 chronyd[31427]: chronyd version 4.2 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDR>
Apr 30 19:49:53 PC1 chronyd[31427]: Frequency -6.507 +/- 15.747 ppm read from /var/lib/chrony/chrony.>
Apr 30 19:49:53 PC1 chronyd[31427]: Using right/UTC timezone to obtain leap second data
Apr 30 19:49:53 PC1 chronyd[31427]: Loaded seccomp filter (level 1)
Apr 30 19:49:53 PC1 systemd[1]: Started chrony, an NTP client/server.
Apr 30 19:50:01 PC1 chronyd[31427]: Selected source 91.189.91.157 (ntp.ubuntu.com)
Apr 30 19:50:01 PC1 chronyd[31427]: System clock TAI offset set to 37 seconds
Apr 30 19:50:03 PC1 chronyd[31427]: Selected source 38.54.86.251 (2.ubuntu.pool.ntp.org)
lines 1-23/23 (END)
```

```
msdomondon@PC1:~$ chronyc
chrony version 4.2
Copyright (C) 1997-2003, 2007, 2009-2021 Richard P. Curnow and others
chrony comes with ABSOLUTELY NO WARRANTY.  This is free software, and
you are welcome to redistribute it under certain conditions.  See the
GNU General Public License version 2 for details.

chronyc>
chronyc>
msdomondon@PC1:~$ sudo mysql -u root -p
[sudo] password for msdomondon:
error: Found option without preceding group in config file: /etc/mysql/mariadb.conf.d/99-openstack.cnf
 at line: 1
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> Ctrl-C -- exit!
Aborted
```


msdomondon@PC1:~$ systemctl status rabbitmq-server
rabbitmq-server.service - RabbitMQ Messaging Server
Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vend>
Active: active (running) since Tue 2024-04-30 19:31:11 PST; 15min ago
Main PID: 750 (beam.smp)
Tasks: 27 (limit: 2254)
Memory: 21.6M
CPU: 46.642s
CGroup: /system.slice/rabbitmq-server.service
├─ 750 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas ageffcb>
├─ 938 erl_child_setup 65536
├─2157 inet_gethost 4
└─2164 inet_gethost 4
Apr 30 19:30:49 PC1 systemd[1]: Starting RabbitMQ Messaging Server...
Apr 30 19:31:11 PC1 systemd[1]: Started RabbitMQ Messaging Server.
msdomondon@PC1:~$ systemctl status memcached
memcached.service - memcached daemon
Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor pre>
Active: active (running) since Tue 2024-04-30 19:35:33 PST; 11min ago
Docs: man:memcached(1)
Main PID: 11738 (memcached)
Tasks: 10 (limit: 2254)
Memory: 1.1M
CPU: 160ms
CGroup: /system.slice/memcached.service
└─11738 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1>
Apr 30 19:35:33 PC1 systemd[1]: Stopped memcached daemon.
Apr 30 19:35:33 PC1 systemd[1]: Started memcached daemon.

- GitHub add, commit, and push.

```
msdomondon@workstation:~/Domondon_HOA13.1$ git add *
msdomondon@workstation:~/Domondon_HOA13.1$ git commit -m "Operational Status"
[main e58c251] Operational Status
 9 files changed, 124 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 playbook.yml
 create mode 100644 roles/ETCD/tasks/main.yml
 create mode 100644 roles/Memcache/tasks/main.yml
 create mode 100644 roles/MessageQ/tasks/main.yml
 create mode 100644 roles/NTP/tasks/main.yml
 create mode 100644 roles/OpenStack/tasks/main.yml
 create mode 100644 roles/SQL/tasks/main.yml
msdomondon@workstation:~/Domondon_HOA13.1$ git push
Username for 'https://github.com': xsagura
Password for 'https://xsagura@github.com':
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (24/24), 2.53 KiB | 1.27 MiB/s, done.
Total 24 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Xsagura/Domondon_HOA13.1.git
   b64de11..e58c251  main -> main
msdomondon@workstation:~/Domondon_HOA13.1$
```



GitHub link: https://github.com/Xsagura/Domondon_HOA13.1.git

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?

OpenStack is a open source cloud computing platform much like AWS, only it is free.

Basically when you get OpenStack, you get a software suite that enables you to set up your own **Cloud** environment.

There are a few ways to deploy your cloud infrastructure using OpenStack.

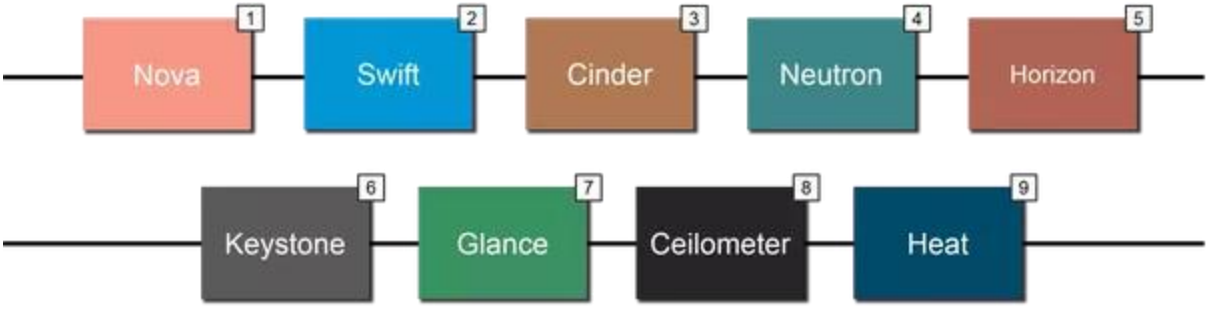
I found and checked the web:

1. **OpenStack-based Public Cloud:** A vendor provides a public cloud computing system based on the OpenStack project
2. **On-Premises Distribution:** In this model, a customer downloads and installs an OpenStack distribution within their internal network.
3. **Hosted OpenStack Private Cloud:** A vendor hosts a OpenStack based private cloud including the underlying hardware and the OpenStack software.
4. **OpenStack-as-a-service:** A vendor hosts OpenStack Management software as a service and customers use it.

Now comes the services that you will be able to enable. The services can be broadly divided into 3 categories:

- **Compute**
- **Storage**
- **Networking**

Now, these services also have attractive names and they are broken down according to different functionalities.



Conclusions:

In this activity I learned OpenStack from scratch can be both challenging and rewarding. It requires good networking, virtualization, and Linux skills. The installation process can be lengthy due to its aggregation of plug-and-play services. Despite difficulties, the experience teaches valuable lessons about system administration, Proxmox, and Ansible.