| | |
|---|---|
| **Name: Renier L. Lope** | **Date Performed: 11/30/2024** |
| **Course/Section: CPE212 - CPE31S2** | **Date Submitted: 11/30/2024** |
| **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

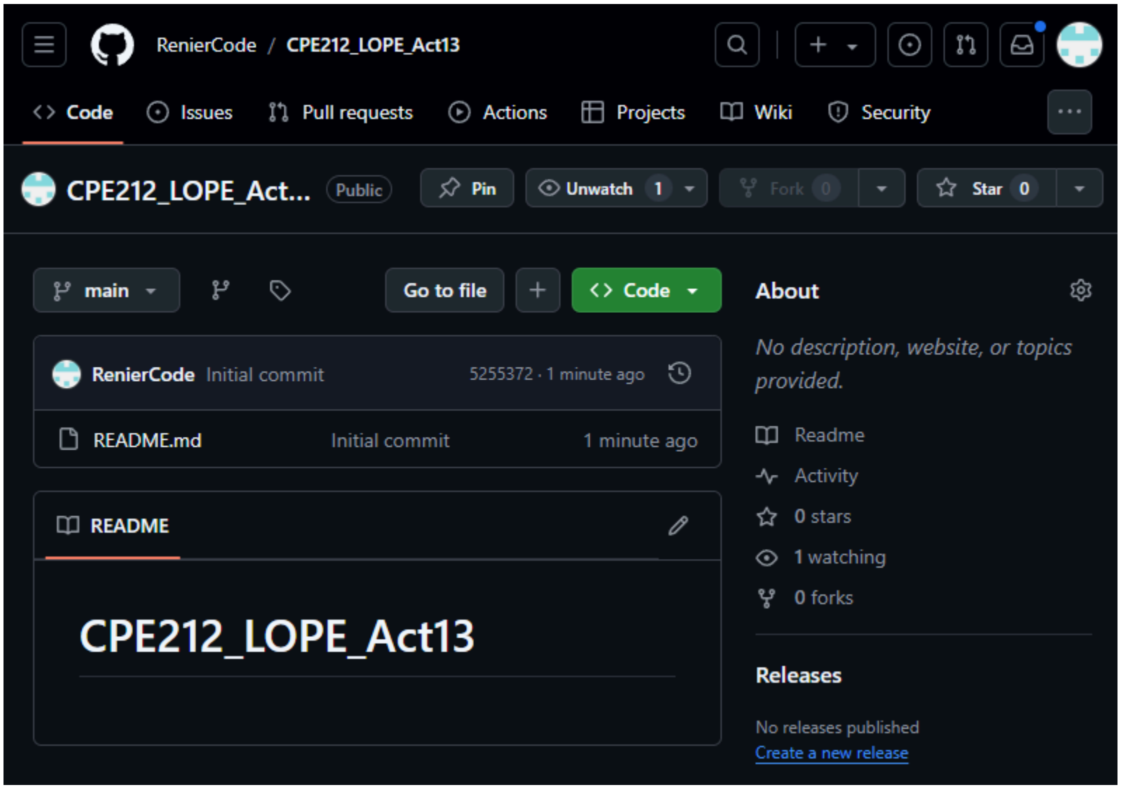
Oracle VirtualBox (Hypervisor)

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)



RenierCode Initial commit 5255372 · 1 minute ago

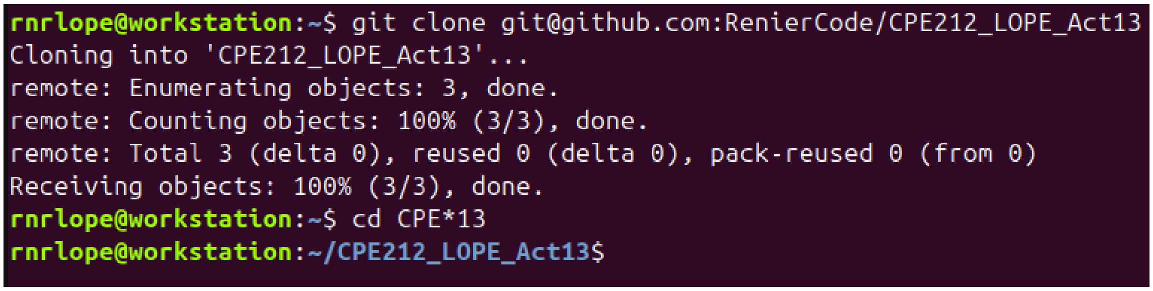
README.md Initial commit 1 minute ago

CPE212_LOPE_Act13

No description, website, or topics provided.

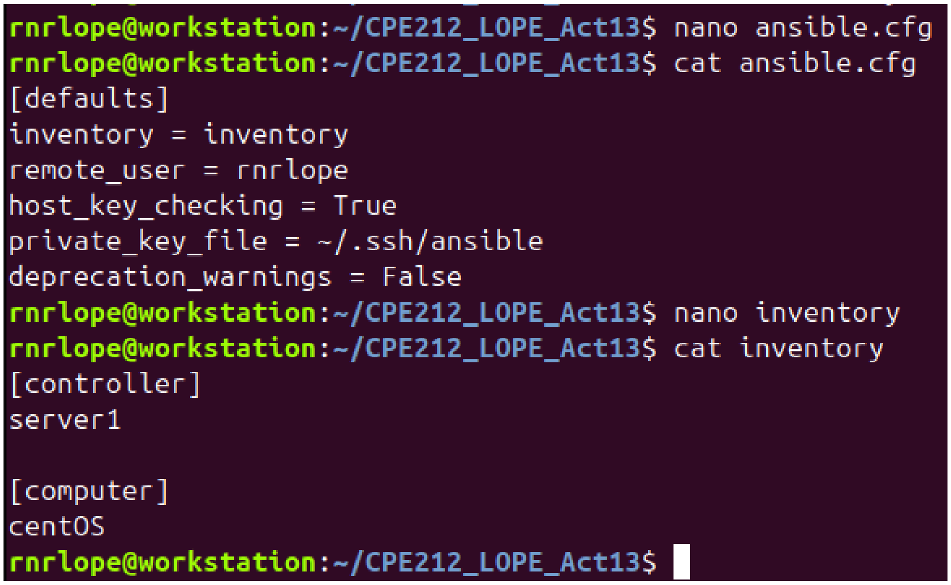
No releases published

**Figure 5.1: Create a new repository for the activity.**



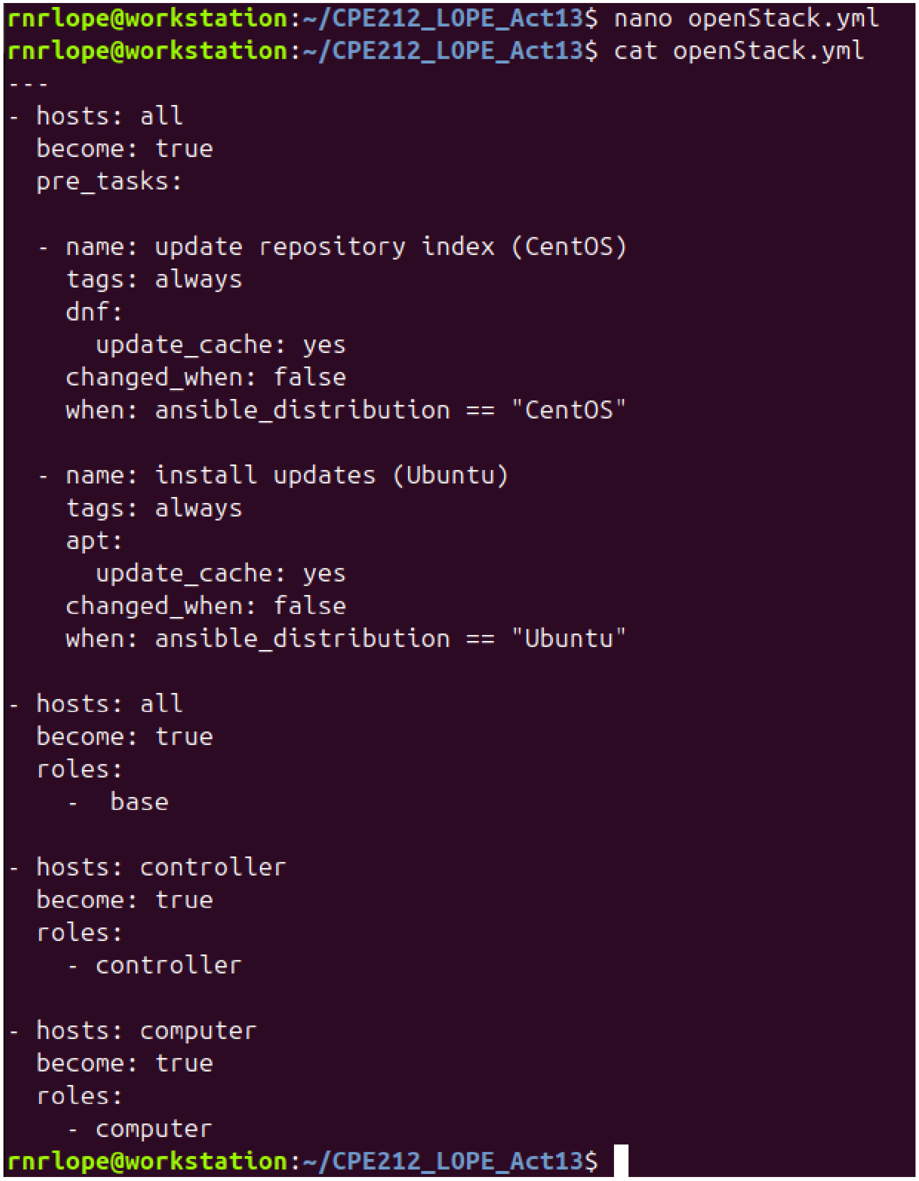
```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/CPE212_LOPE_Act13
Cloning into 'CPE212_LOPE_Act13'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
rnrlope@workstation:~$ cd CPE*13
rnrlope@workstation:~/CPE212_LOPE_Act13$
```

**Figure 5.2: Clone the new repository at the local machine.**



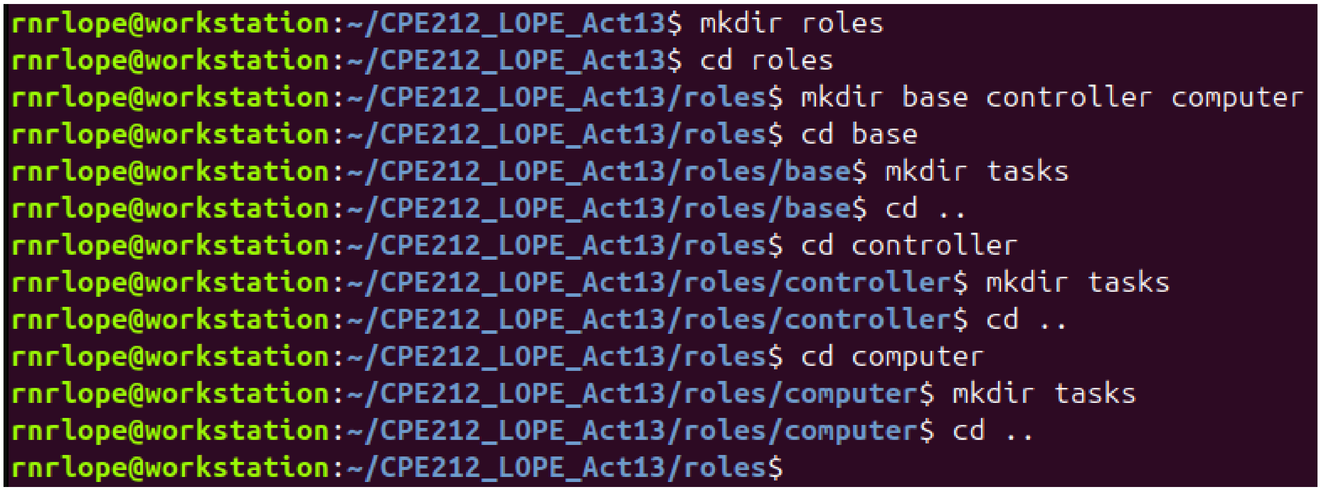
```
rnrlope@workstation:~/CPE212_LOPE_Act13$ nano ansible.cfg
rnrlope@workstation:~/CPE212_LOPE_Act13$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
private_key_file = ~/.ssh/ansible
deprecation_warnings = False
rnrlope@workstation:~/CPE212_LOPE_Act13$ nano inventory
rnrlope@workstation:~/CPE212_LOPE_Act13$ cat inventory
[controller]
server1

[computer]
centOS
rnrlope@workstation:~/CPE212_LOPE_Act13$
```

**Figure 5.3: Create the “ansible.cfg” and “inventory” files. (Contents of the files)**



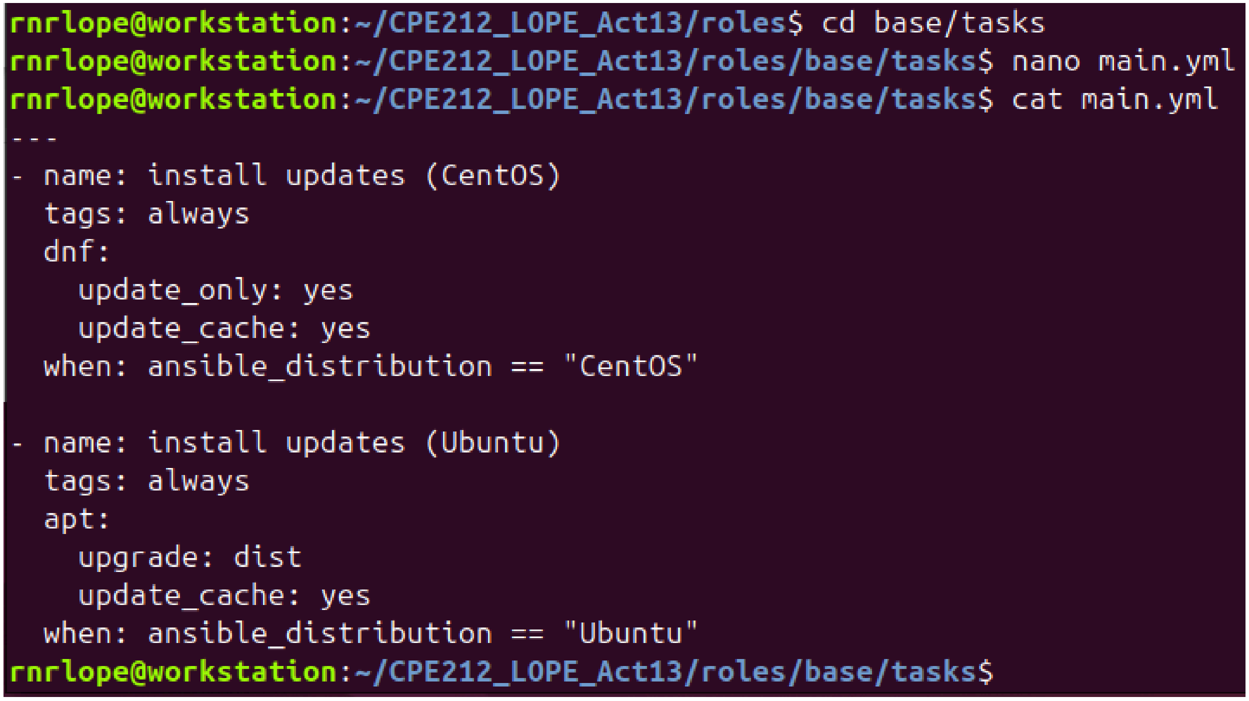
```
rnrlope@workstation:~/CPE212_LOPE_Act13$ nano openStack.yml
rnrlope@workstation:~/CPE212_LOPE_Act13$ cat openStack.yml
---
- hosts: all
  become: true
  pre_tasks:

  - 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
rnrlope@workstation:~/CPE212_LOPE_Act13$ 
```

**Figure 5.4: Create a playbook file named "openStack.yml" to repository index and to execute the task of the specified roles.**



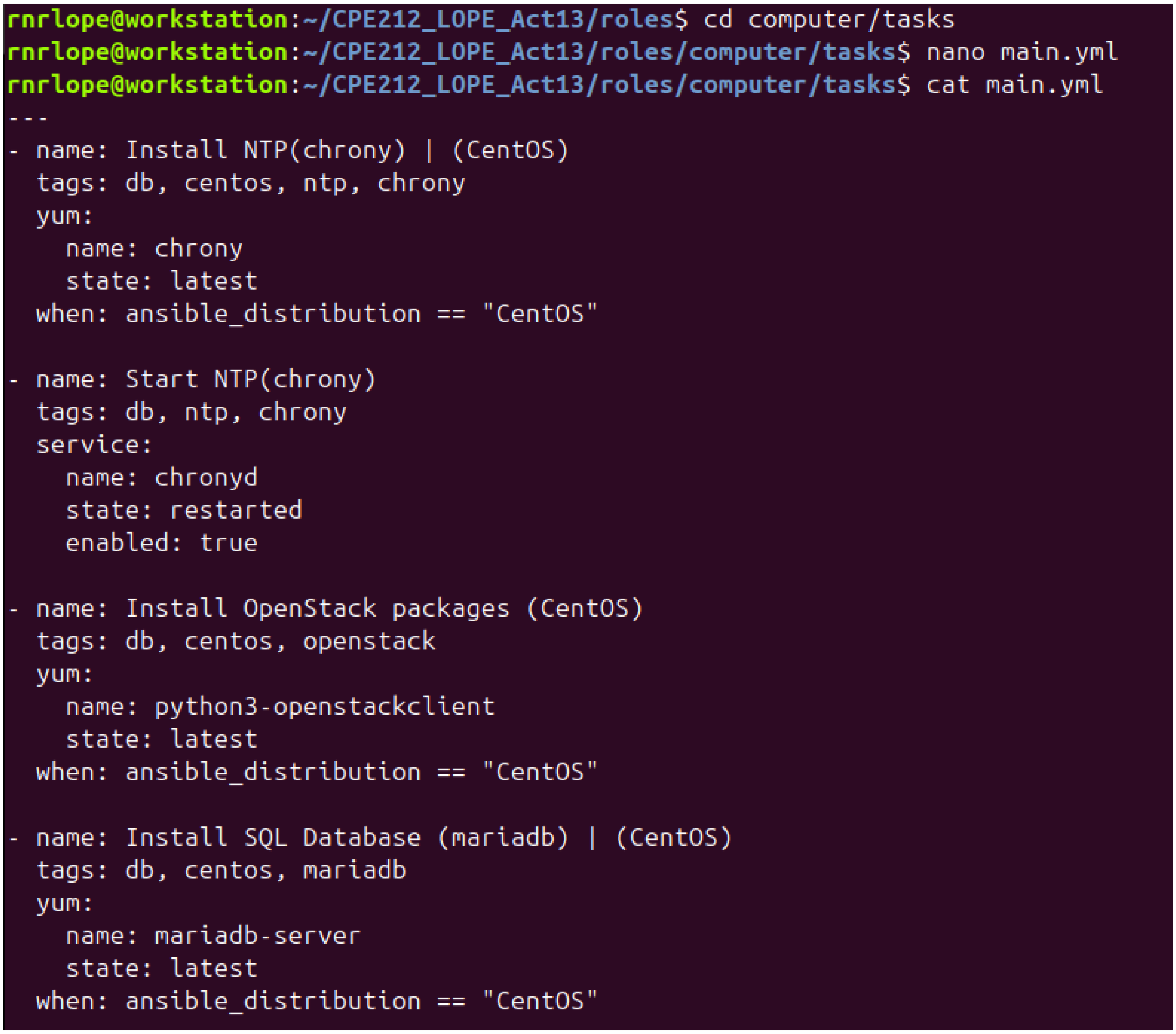
```
rnrlope@workstation:~/CPE212_LOPE_Act13$ mkdir roles
rnrlope@workstation:~/CPE212_LOPE_Act13$ cd roles
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ mkdir base controller computer
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd base
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/base$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/base$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd controller
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/controller$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/controller$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd computer
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/computer$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/computer$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$
```

**Figure 5.5: Create a new directory named "roles" to store the roles and inside create directories for the specified roles such as "base", "controller", and "computer" to separate the tasks based on the groups in inventory.**

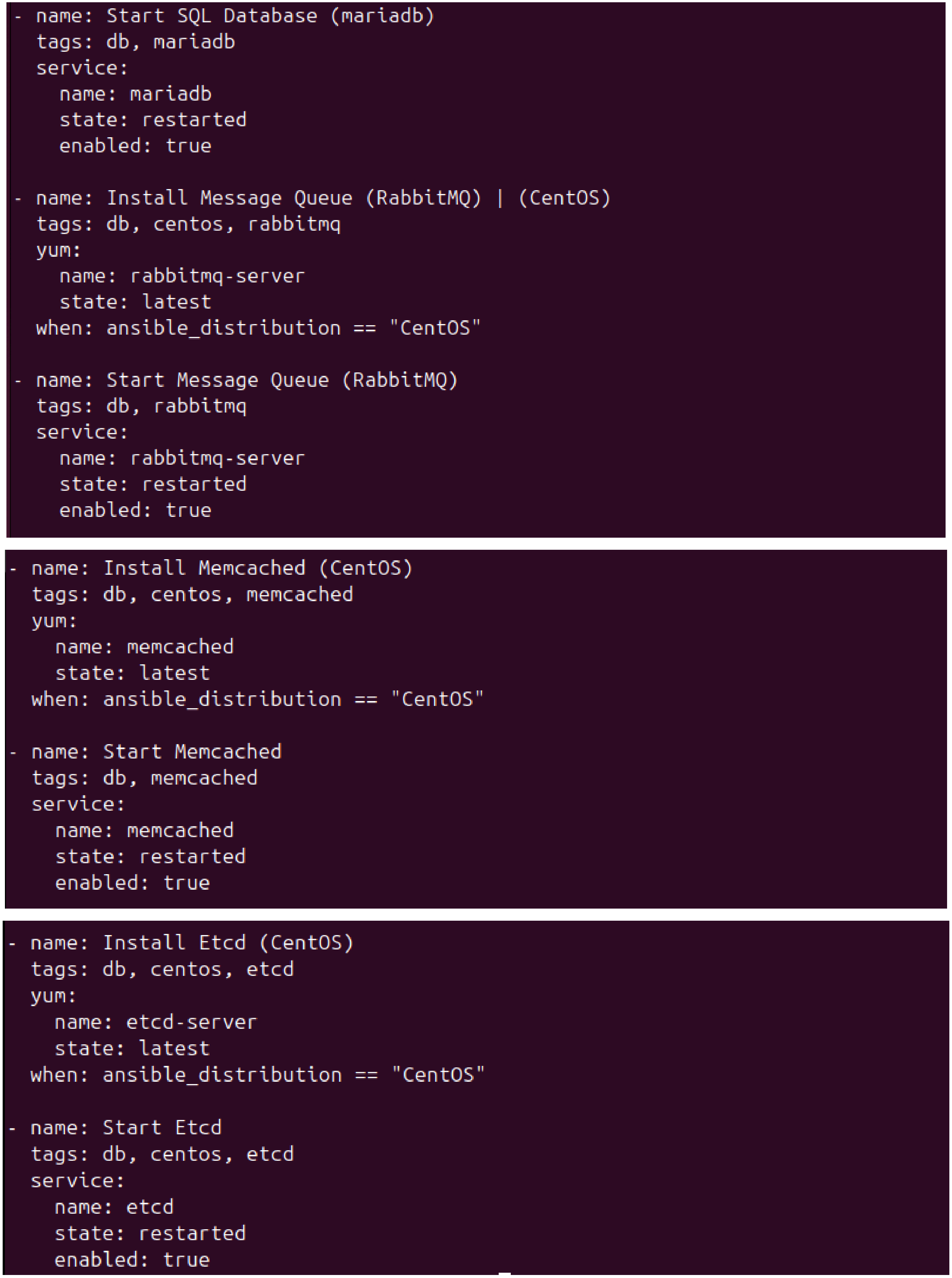


```
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd base/tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/base/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/base/tasks$ cat main.yml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/base/tasks$
```

**Figure 5.6: Create a playbook file inside roles/base/tasks and put the desired tasks inside the playbook.**



```
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd computer/tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/computer/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/computer/tasks$ cat main.yml
---
- name: Install NTP(chrony) | (CentOS)
  tags: db, centos, ntp, chrony
  yum:
    name: chrony
    state: latest
  when: ansible_distribution == "CentOS"

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

- name: Install OpenStack packages (CentOS)
  tags: db, centos, openstack
  yum:
    name: python3-openstackclient
    state: latest
  when: ansible_distribution == "CentOS"

- name: Install SQL Database (mariadb) | (CentOS)
  tags: db, centos, mariadb
  yum:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"
```

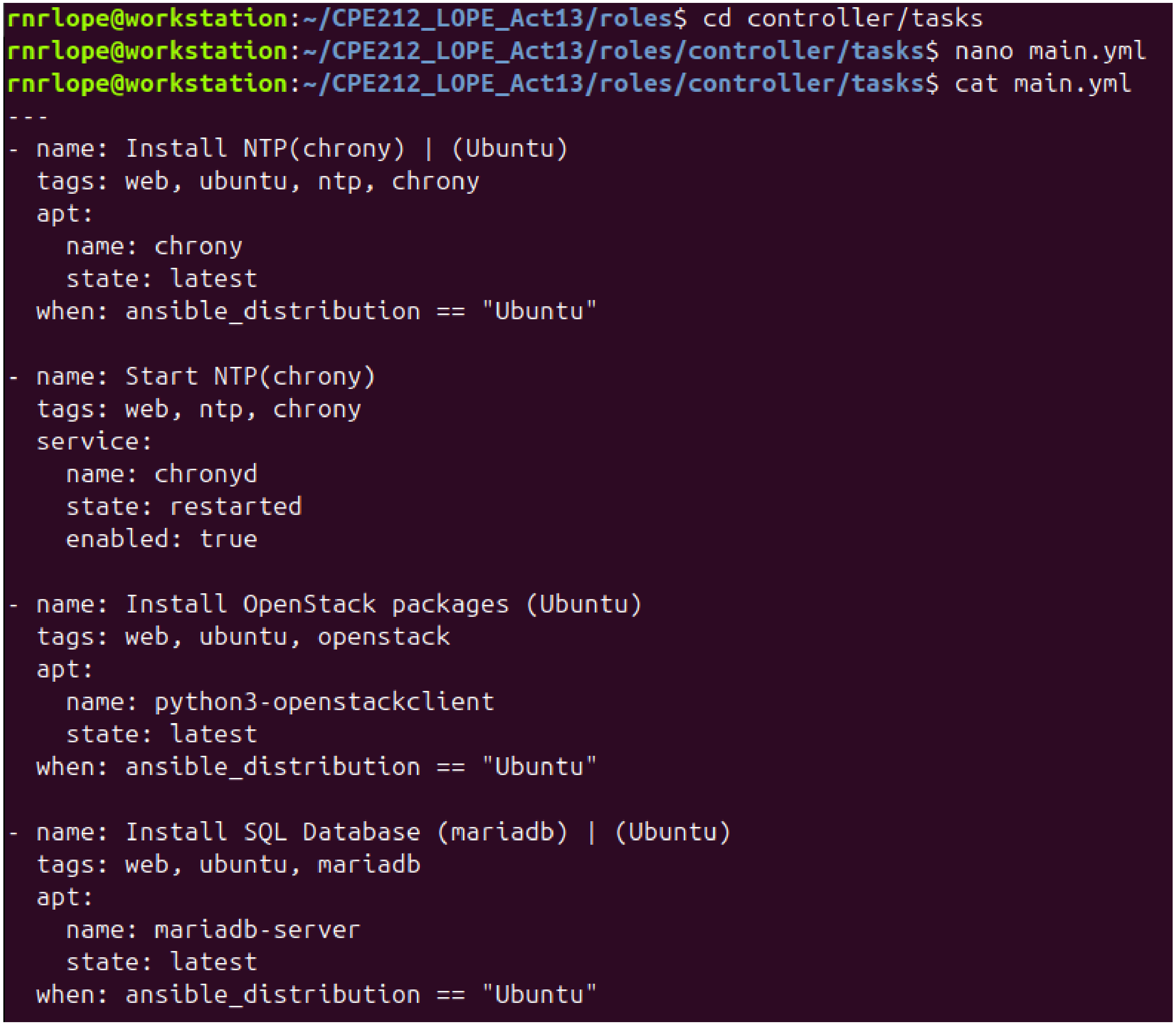


```
- name: Start SQL Database (mariadb)
  tags: db, mariadb
  service:
    name: mariadb
    state: restarted
    enabled: true

- name: Install Message Queue (RabbitMQ) | (CentOS)
  tags: db, centos, rabbitmq
  yum:
    name: rabbitmq-server
    state: latest
  when: ansible_distribution == "CentOS"

- name: Start Message Queue (RabbitMQ)
  tags: db, rabbitmq
  service:
    name: rabbitmq-server
    state: restarted
    enabled: true
```

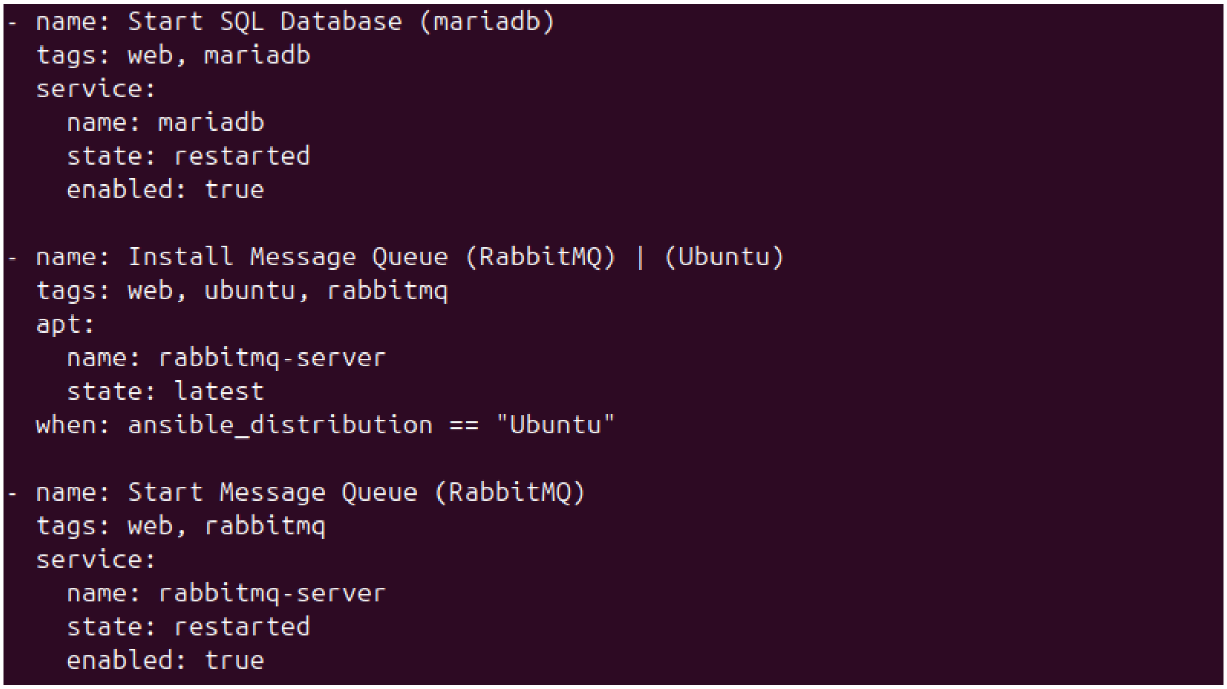
```
- name: Install Memcached (CentOS)
  tags: db, centos, memcached
  yum:
    name: memcached
    state: latest
  when: ansible_distribution == "CentOS"

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

```
- name: Install Etcd (CentOS)
  tags: db, centos, etcd
  yum:
    name: etcd-server
    state: latest
  when: ansible_distribution == "CentOS"

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

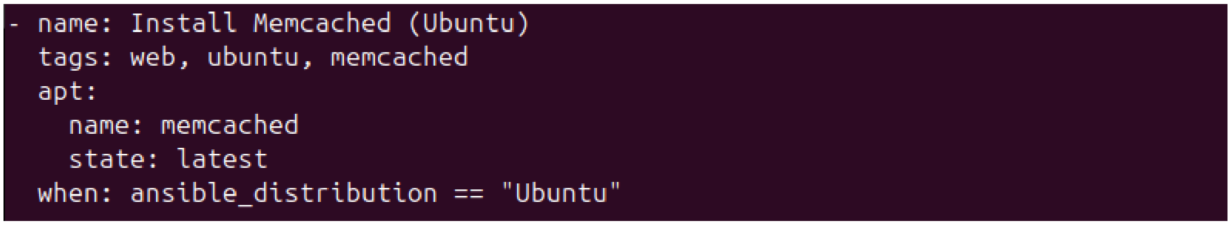
**Figure 5.7 - 5.10: Create a playbook file inside roles/computer/tasks and put the desired tasks for the group “computer” inside the playbook.**



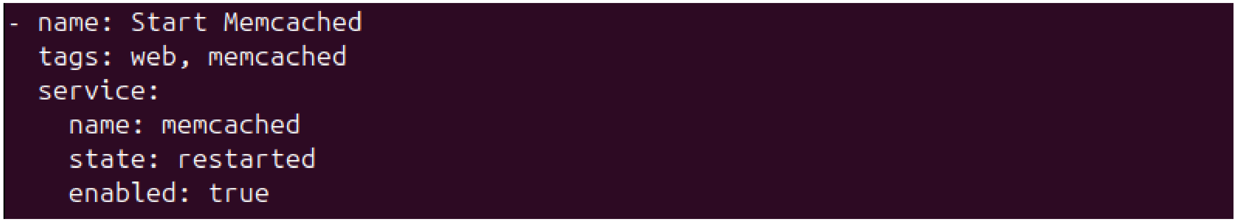
```
rnrlope@workstation:~/CPE212_LOPE_Act13/roles$ cd controller/tasks
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/controller/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act13/roles/controller/tasks$ cat main.yml
---
- name: Install NTP(chrony) | (Ubuntu)
  tags: web, ubuntu, ntp, chrony
  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 (Ubuntu)
  tags: web, ubuntu, openstack
  apt:
    name: python3-openstackclient
    state: latest
  when: ansible_distribution == "Ubuntu"

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



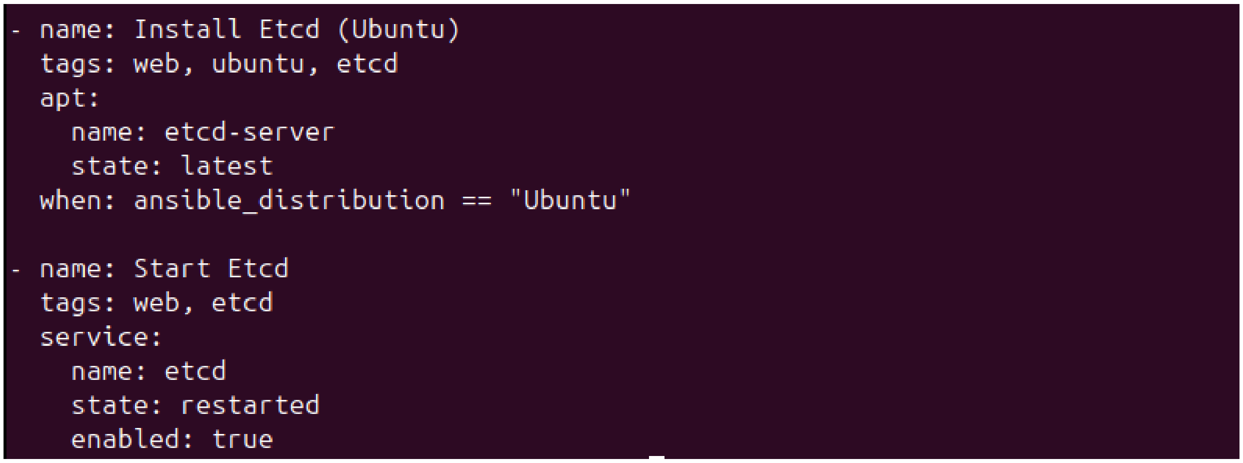
```
- name: Start SQL Database (mariadb)
  tags: web, mariadb
  service:
    name: mariadb
    state: restarted
    enabled: true

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

- name: Start Message Queue (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"
```

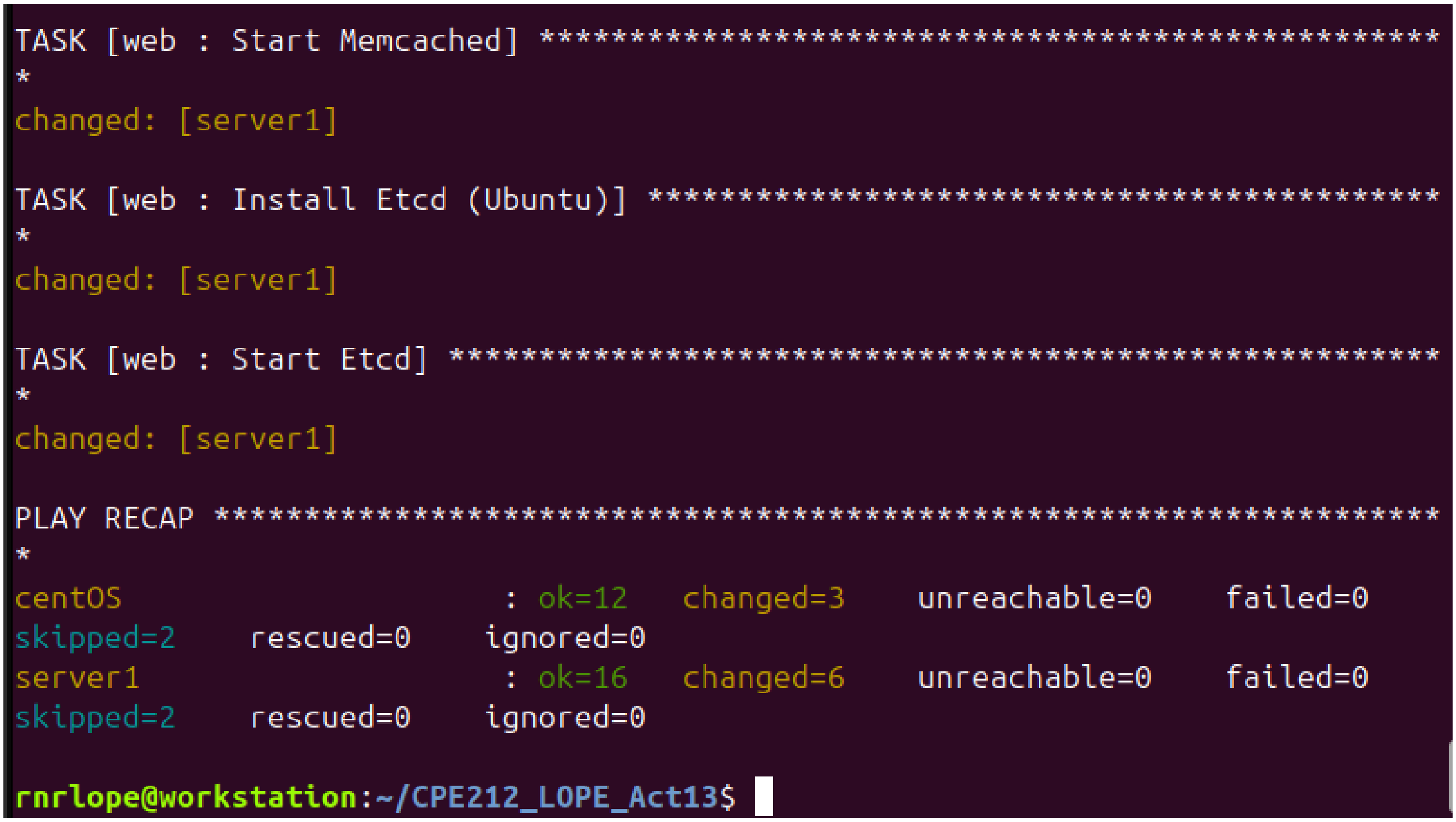


```
- name: Start Memcached
  tags: web, memcached
  service:
    name: memcached
    state: restarted
    enabled: true
```



```
- name: Install Etcd (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
```

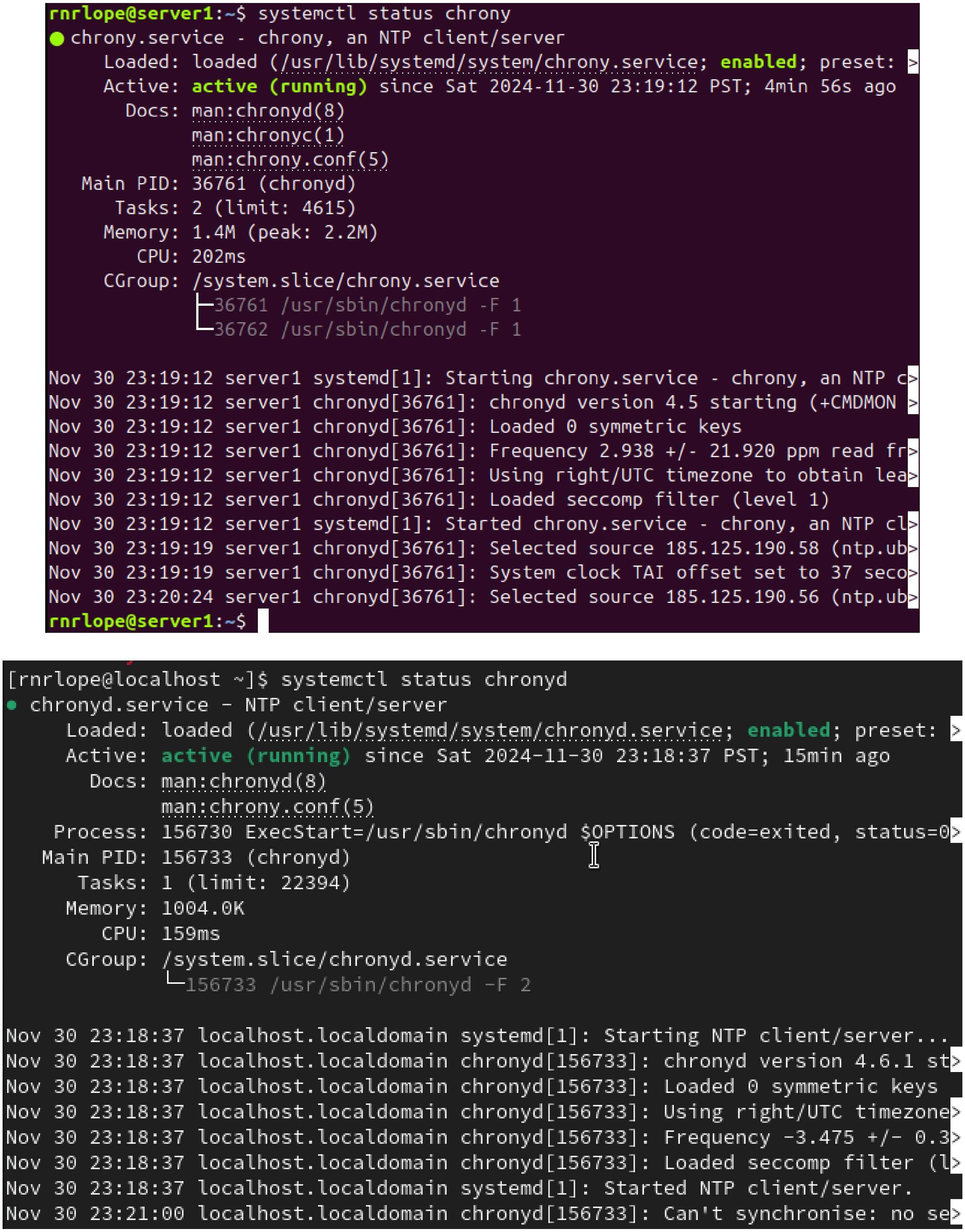
**Figure 5.11- 5.14: Create a playbook file inside roles/controller/tasks and put the desired tasks for the group "controller" inside the playbook.**



```
TASK [web : Start Memcached] ***************************************************
*
changed: [server1]

TASK [web : Install Etcd (Ubuntu)] *********************************************
*
changed: [server1]

TASK [web : Start Etcd] ********************************************************
*
changed: [server1]

PLAY RECAP *********************************************************************
*
centOS                     : ok=12   changed=3    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=16   changed=6    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act13$
```

**FIgure 5.15: PLAY RECAP of executing the playbook file openStack.yml**

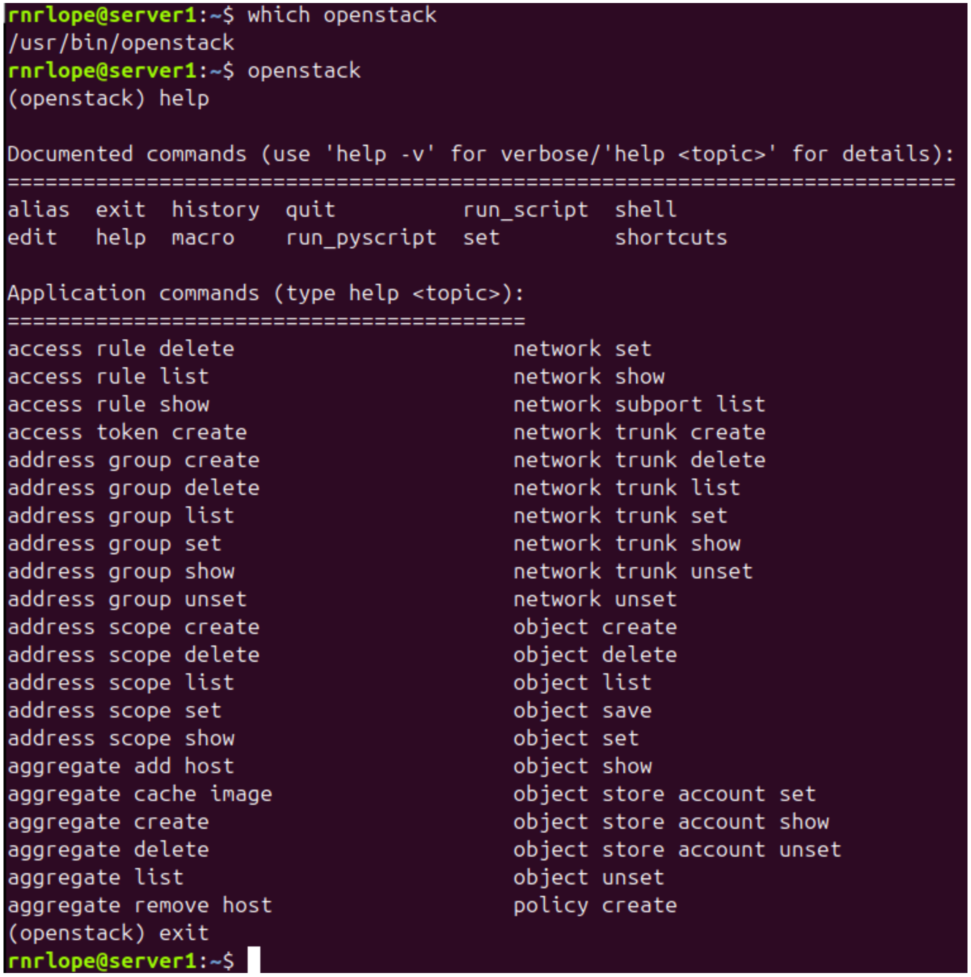
## VERIFYING INSTALLS:



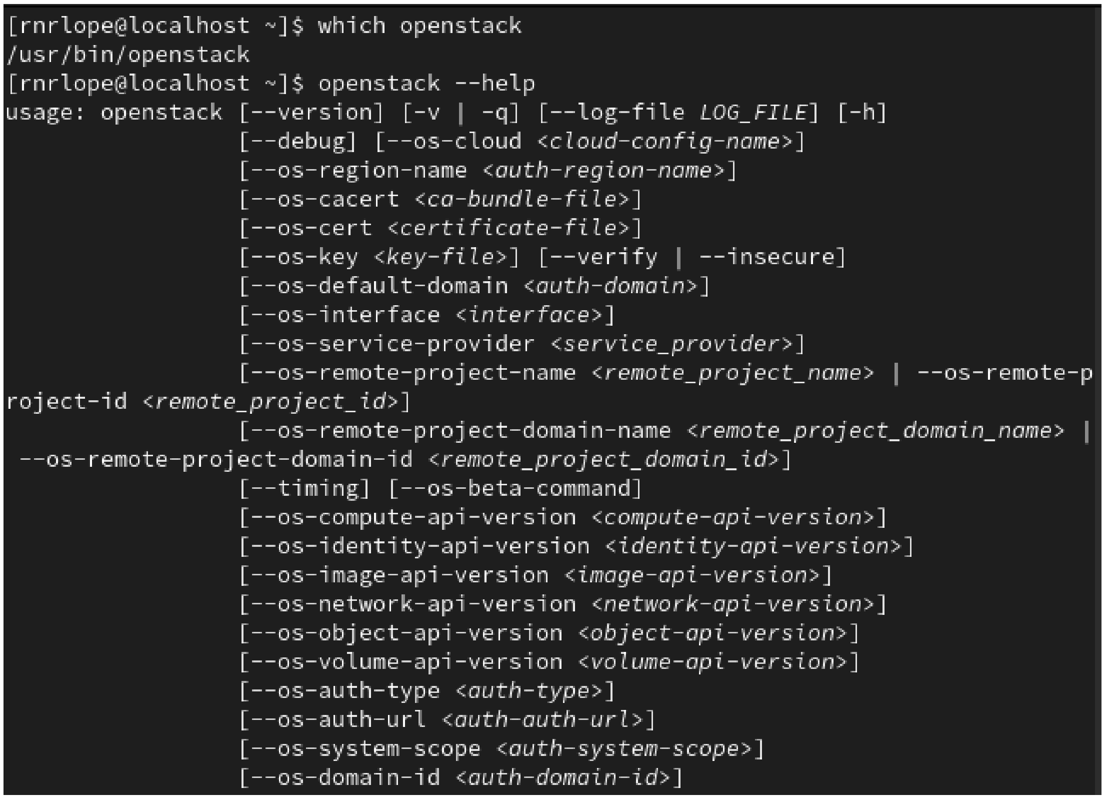
```
rnrlope@server1:~$ systemctl status chrony
● chrony.service - chrony, an NTP client/server
     Loaded: loaded (/usr/lib/systemd/system/chrony.service; enabled; preset: >
     Active: active (running) since Sat 2024-11-30 23:19:12 PST; 4min 56s ago
       Docs: man:chronyd(8)
             man:chronyc(1)
             man:chrony.conf(5)
   Main PID: 36761 (chronyd)
      Tasks: 2 (limit: 4615)
     Memory: 1.4M (peak: 2.2M)
        CPU: 202ms
     CGroup: /system.slice/chrony.service
             ├─36761 /usr/sbin/chronyd -F 1
             └─36762 /usr/sbin/chronyd -F 1

Nov 30 23:19:12 server1 systemd[1]: Starting chrony.service - chrony, an NTP c>
Nov 30 23:19:12 server1 chronyd[36761]: chronyd version 4.5 starting (+CMDMON >
Nov 30 23:19:12 server1 chronyd[36761]: Loaded 0 symmetric keys
Nov 30 23:19:12 server1 chronyd[36761]: Frequency 2.938 +/- 21.920 ppm read fr>
Nov 30 23:19:12 server1 chronyd[36761]: Using right/UTC timezone to obtain lea>
Nov 30 23:19:12 server1 chronyd[36761]: Loaded seccomp filter (level 1)
Nov 30 23:19:12 server1 systemd[1]: Started chrony.service - chrony, an NTP cl>
Nov 30 23:19:19 server1 chronyd[36761]: Selected source 185.125.190.58 (ntp.ub>
Nov 30 23:19:19 server1 chronyd[36761]: System clock TAI offset set to 37 seco>
Nov 30 23:20:24 server1 chronyd[36761]: Selected source 185.125.190.56 (ntp.ub>
rnrlope@server1:~$
```

```
[rnrlope@localhost ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
     Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; preset: >
     Active: active (running) since Sat 2024-11-30 23:18:37 PST; 15min ago
       Docs: man:chronyd(8)
             man:chrony.conf(5)
    Process: 156730 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0>
   Main PID: 156733 (chronyd)
      Tasks: 1 (limit: 22394)
     Memory: 1004.0K
        CPU: 159ms
     CGroup: /system.slice/chronyd.service
             └─156733 /usr/sbin/chronyd -F 2

Nov 30 23:18:37 localhost.localdomain systemd[1]: Starting NTP client/server...
Nov 30 23:18:37 localhost.localdomain chronyd[156733]: chronyd version 4.6.1 st>
Nov 30 23:18:37 localhost.localdomain chronyd[156733]: Loaded 0 symmetric keys
Nov 30 23:18:37 localhost.localdomain chronyd[156733]: Using right/UTC timezone>
Nov 30 23:18:37 localhost.localdomain chronyd[156733]: Frequency -3.475 +/- 0.3>
Nov 30 23:18:37 localhost.localdomain chronyd[156733]: Loaded seccomp filter (l>
Nov 30 23:18:37 localhost.localdomain systemd[1]: Started NTP client/server.
Nov 30 23:21:00 localhost.localdomain chronyd[156733]: Can't synchronise: no se>
```

**FIgure 5.16 - 5.17: Verifying if NTP (chrony) is installed.**

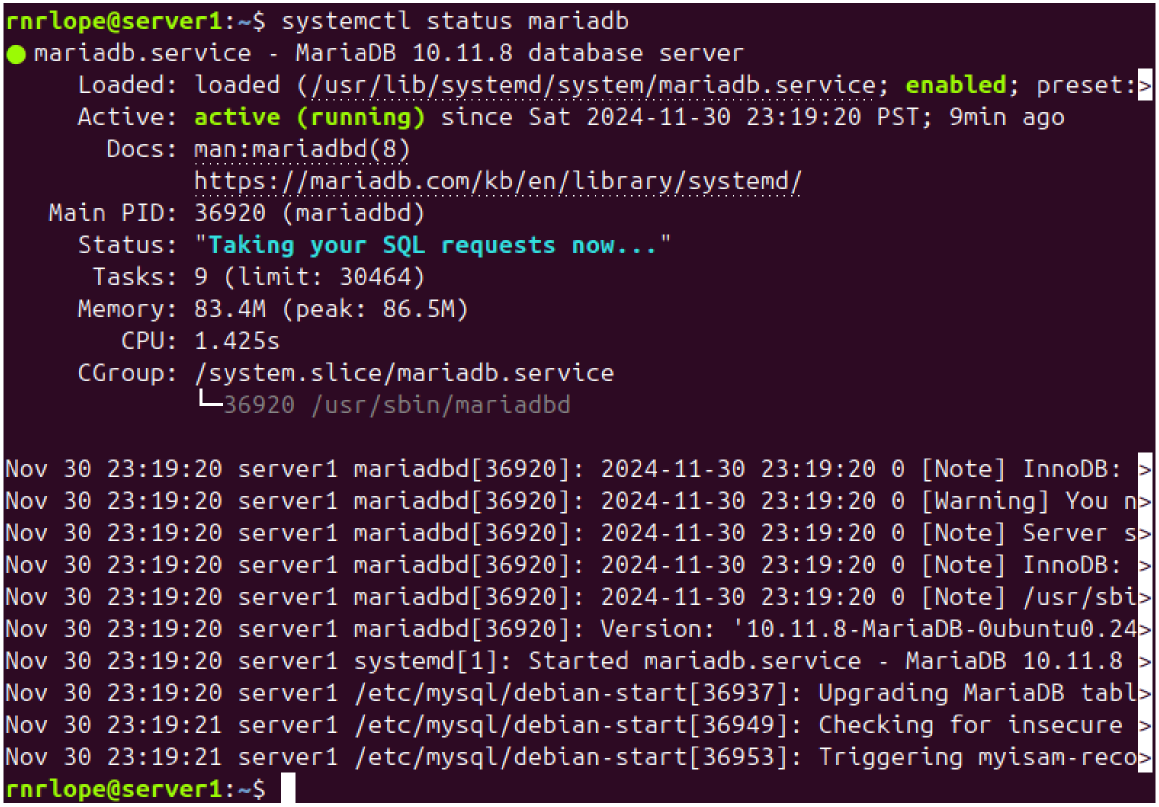


```
rnrlope@server1:~$ which openstack
/usr/bin/openstack
rnrlope@server1:~$ openstack
(openstack) help

Documented commands (use 'help -v' for verbose/'help <topic>' for details):
===========================================================================
alias  exit  history  quit          run_script  shell
edit   help  macro    run_pyscript  set         shortcuts

Application commands (type help <topic>):
=========================================
access rule delete                      network set
access rule list                        network show
access rule show                        network subport list
access token create                     network trunk create
address group create                    network trunk delete
address group delete                    network trunk list
address group list                      network trunk set
address group set                       network trunk show
address group show                      network trunk unset
address group unset                     network unset
address scope create                    object create
address scope delete                    object delete
address scope list                      object list
address scope set                       object save
address scope show                      object set
aggregate add host                      object show
aggregate cache image                   object store account set
aggregate create                        object store account show
aggregate delete                        object store account unset
aggregate list                          object unset
aggregate remove host                   policy create
(openstack) exit
rnrlope@server1:~$
```

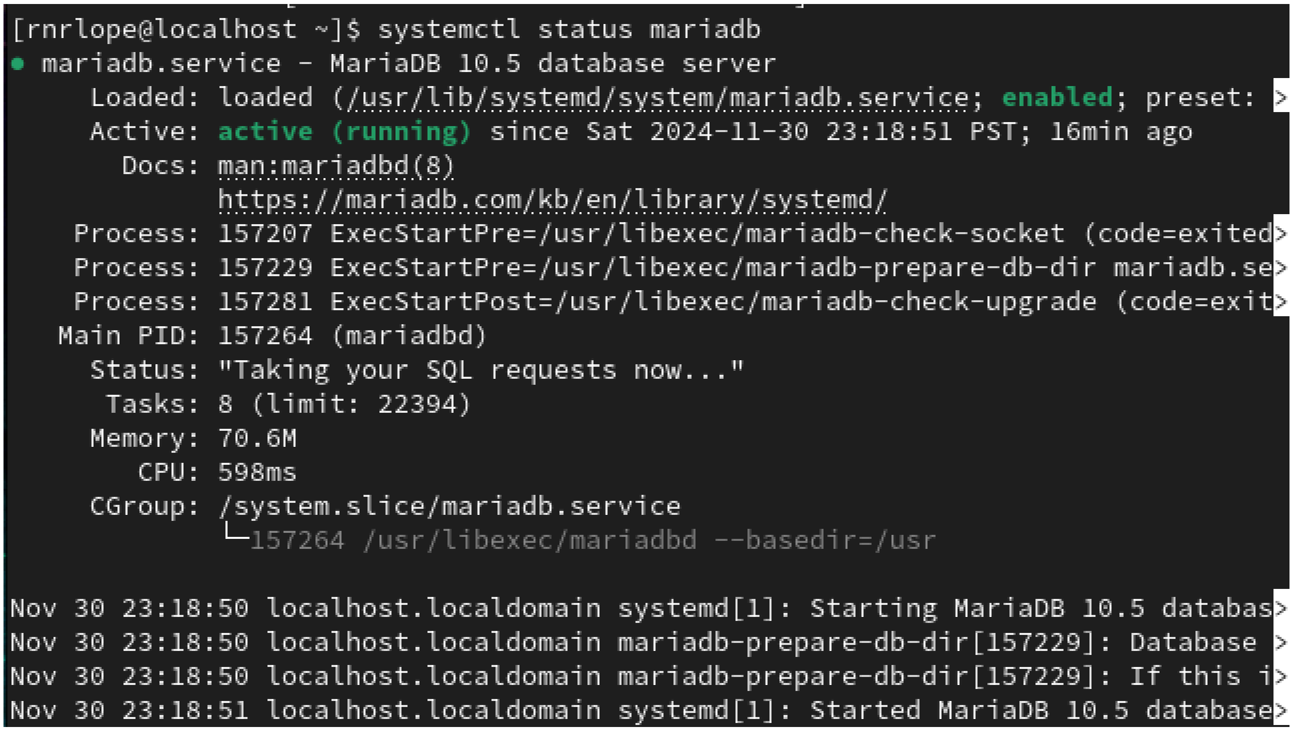


```
[rnrlope@localhost ~]$ which openstack
/usr/bin/openstack
[rnrlope@localhost ~]$ openstack --help
usage: openstack [--version] [-v | -q] [--log-file LOG_FILE] [-h]
                 [--debug] [--os-cloud <cloud-config-name>]
                 [--os-region-name <auth-region-name>]
                 [--os-cacert <ca-bundle-file>]
                 [--os-cert <certificate-file>]
                 [--os-key <key-file>] [--verify | --insecure]
                 [--os-default-domain <auth-domain>]
                 [--os-interface <interface>]
                 [--os-service-provider <service_provider>]
                 [--os-remote-project-name <remote_project_name> | --os-remote-p
roject-id <remote_project_id>]
                 [--os-remote-project-domain-name <remote_project_domain_name> |
 --os-remote-project-domain-id <remote_project_domain_id>]
                 [--timing] [--os-beta-command]
                 [--os-compute-api-version <compute-api-version>]
                 [--os-identity-api-version <identity-api-version>]
                 [--os-image-api-version <image-api-version>]
                 [--os-network-api-version <network-api-version>]
                 [--os-object-api-version <object-api-version>]
                 [--os-volume-api-version <volume-api-version>]
                 [--os-auth-type <auth-type>]
                 [--os-auth-url <auth-auth-url>]
                 [--os-system-scope <auth-system-scope>]
                 [--os-domain-id <auth-domain-id>]
```

**FIgure 5.18 - 5.19: Verifying if OpenStack is installed.**

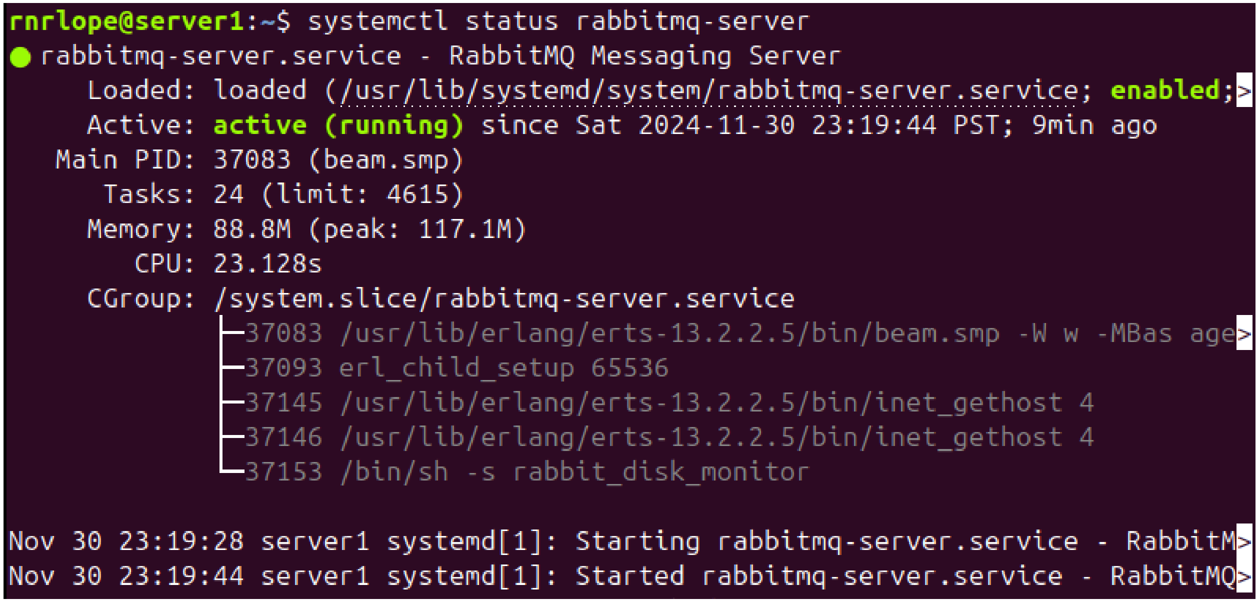


```
rnrlope@server1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.11.8 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset:>
     Active: active (running) since Sat 2024-11-30 23:19:20 PST; 9min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 36920 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 30464)
     Memory: 83.4M (peak: 86.5M)
        CPU: 1.425s
     CGroup: /system.slice/mariadb.service
             └─36920 /usr/sbin/mariadbd

Nov 30 23:19:20 server1 mariadbd[36920]: 2024-11-30 23:19:20 0 [Note] InnoDB: >
Nov 30 23:19:20 server1 mariadbd[36920]: 2024-11-30 23:19:20 0 [Warning] You n>
Nov 30 23:19:20 server1 mariadbd[36920]: 2024-11-30 23:19:20 0 [Note] Server s>
Nov 30 23:19:20 server1 mariadbd[36920]: 2024-11-30 23:19:20 0 [Note] InnoDB: >
Nov 30 23:19:20 server1 mariadbd[36920]: 2024-11-30 23:19:20 0 [Note] /usr/sbi>
Nov 30 23:19:20 server1 mariadbd[36920]: Version: '10.11.8-MariaDB-0ubuntu0.24>
Nov 30 23:19:20 server1 systemd[1]: Started mariadb.service - MariaDB 10.11.8 >
Nov 30 23:19:20 server1 /etc/mysql/debian-start[36937]: Upgrading MariaDB tabl>
Nov 30 23:19:21 server1 /etc/mysql/debian-start[36949]: Checking for insecure >
Nov 30 23:19:21 server1 /etc/mysql/debian-start[36953]: Triggering myisam-reco>
rnrlope@server1:~$
```



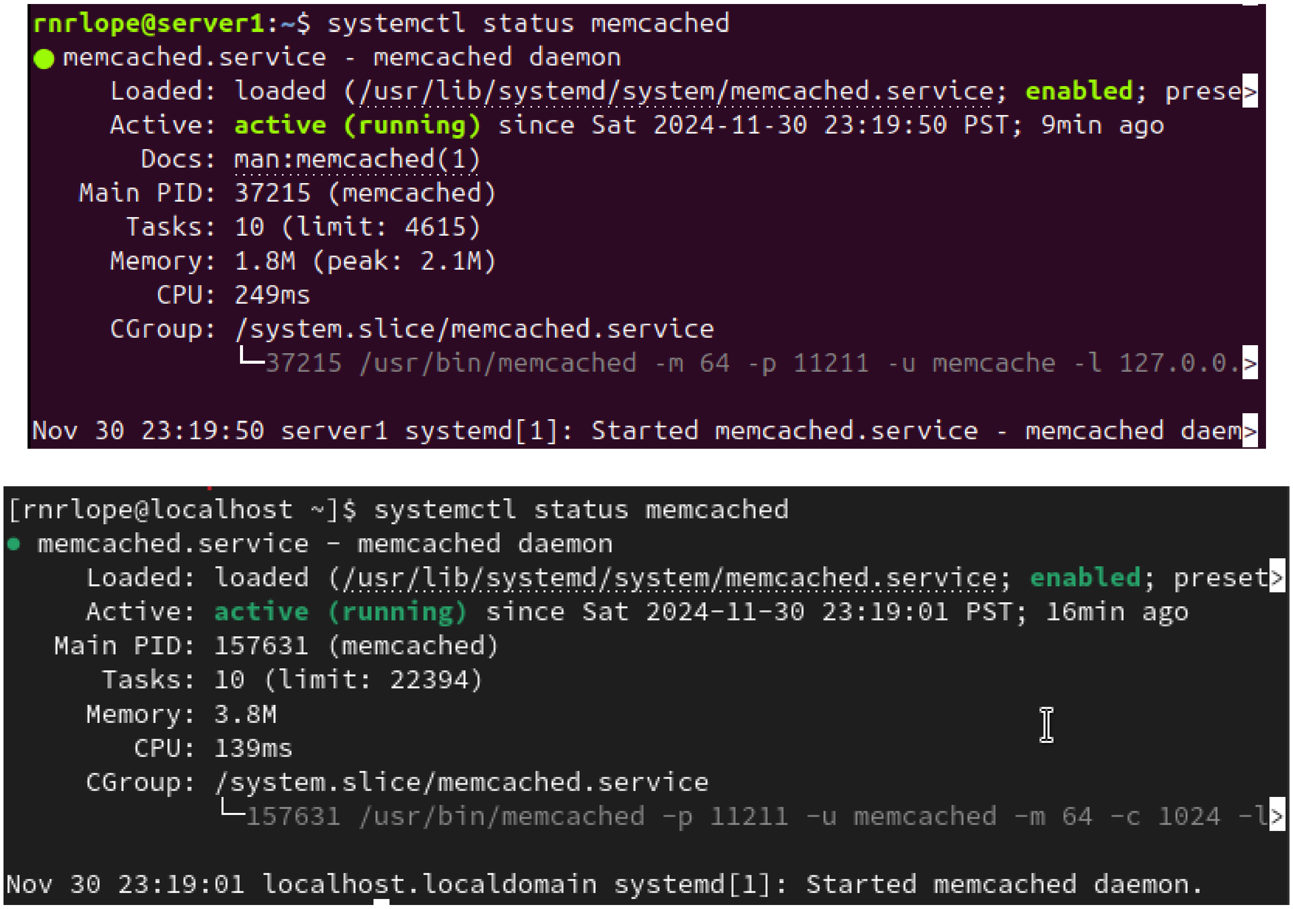
```
[rnrlope@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Sat 2024-11-30 23:18:51 PST; 16min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 157207 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited>
    Process: 157229 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.se>
    Process: 157281 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exit>
   Main PID: 157264 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 22394)
     Memory: 70.6M
        CPU: 598ms
     CGroup: /system.slice/mariadb.service
             └─157264 /usr/libexec/mariadbd --basedir=/usr

Nov 30 23:18:50 localhost.localdomain systemd[1]: Starting MariaDB 10.5 databas>
Nov 30 23:18:50 localhost.localdomain mariadb-prepare-db-dir[157229]: Database >
Nov 30 23:18:50 localhost.localdomain mariadb-prepare-db-dir[157229]: If this i>
Nov 30 23:18:51 localhost.localdomain systemd[1]: Started MariaDB 10.5 database>
```

**FIgure 5.20 - 5.21: Verifying if SQL Database (mariadb) is installed.**



```
rnrlope@server1:~$ systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ Messaging Server
     Loaded: loaded (/usr/lib/systemd/system/rabbitmq-server.service; enabled;>
     Active: active (running) since Sat 2024-11-30 23:19:44 PST; 9min ago
   Main PID: 37083 (beam.smp)
      Tasks: 24 (limit: 4615)
     Memory: 88.8M (peak: 117.1M)
        CPU: 23.128s
     CGroup: /system.slice/rabbitmq-server.service
             ├─37083 /usr/lib/erlang/erts-13.2.2.5/bin/beam.smp -W w -MBas age>
             ├─37093 erl_child_setup 65536
             ├─37145 /usr/lib/erlang/erts-13.2.2.5/bin/inet_gethost 4
             ├─37146 /usr/lib/erlang/erts-13.2.2.5/bin/inet_gethost 4
             └─37153 /bin/sh -s rabbit_disk_monitor

Nov 30 23:19:28 server1 systemd[1]: Starting rabbitmq-server.service - RabbitM>
Nov 30 23:19:44 server1 systemd[1]: Started rabbitmq-server.service - RabbitMQ>
```

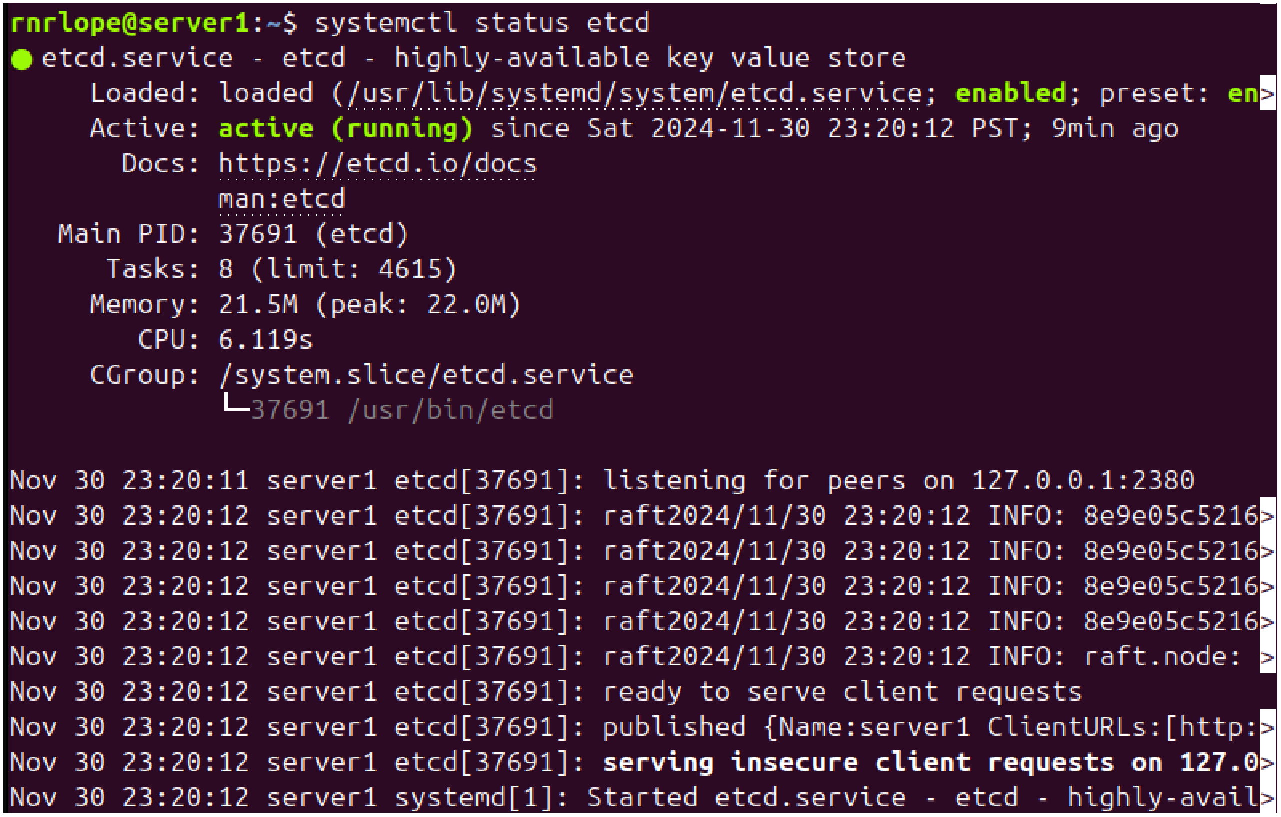
**FIgure 5.22: Verifying if Message Queue (RabbitMQ) is installed.**



```
rnrlope@server1:~$ systemctl status memcached
● memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; prese>
     Active: active (running) since Sat 2024-11-30 23:19:50 PST; 9min ago
       Docs: man:memcached(1)
   Main PID: 37215 (memcached)
      Tasks: 10 (limit: 4615)
     Memory: 1.8M (peak: 2.1M)
        CPU: 249ms
     CGroup: /system.slice/memcached.service
             └─37215 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.>

Nov 30 23:19:50 server1 systemd[1]: Started memcached.service - memcached daem>
```

```
[rnrlope@localhost ~]$ systemctl status memcached
● memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset>
     Active: active (running) since Sat 2024-11-30 23:19:01 PST; 16min ago
   Main PID: 157631 (memcached)
      Tasks: 10 (limit: 22394)
     Memory: 3.8M
        CPU: 139ms
     CGroup: /system.slice/memcached.service
             └─157631 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l>

Nov 30 23:19:01 localhost.localdomain systemd[1]: Started memcached daemon.
```

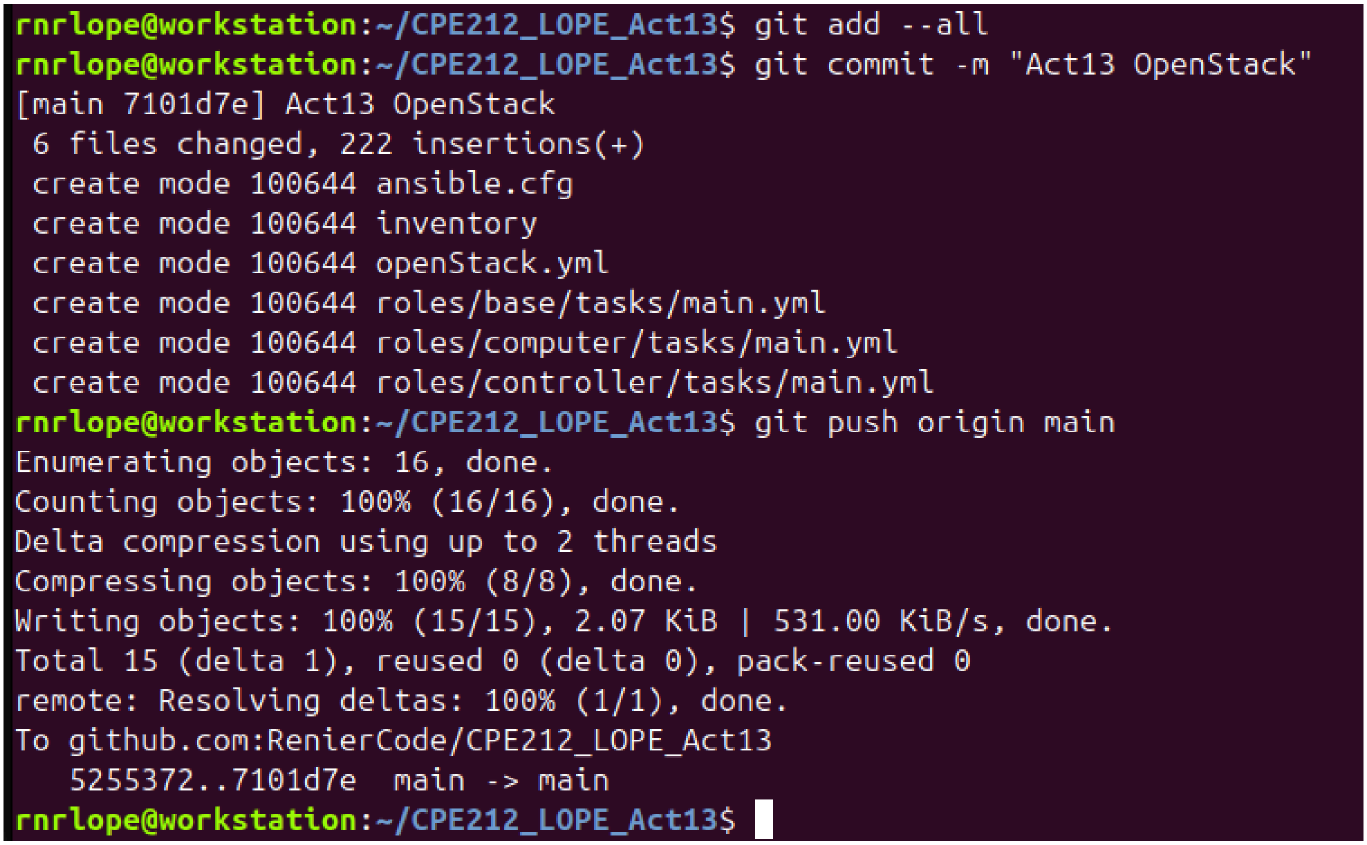
**FIgure 5.23 - 5.24: Verifying if Memcached is installed.**



```
rnrlope@server1:~$ systemctl status etcd
● etcd.service - etcd - highly-available key value store
     Loaded: loaded (/usr/lib/systemd/system/etcd.service; enabled; preset: en>
     Active: active (running) since Sat 2024-11-30 23:20:12 PST; 9min ago
       Docs: https://etcd.io/docs
             man:etcd
   Main PID: 37691 (etcd)
      Tasks: 8 (limit: 4615)
     Memory: 21.5M (peak: 22.0M)
        CPU: 6.119s
     CGroup: /system.slice/etcd.service
             └─37691 /usr/bin/etcd

Nov 30 23:20:11 server1 etcd[37691]: listening for peers on 127.0.0.1:2380
Nov 30 23:20:12 server1 etcd[37691]: raft2024/11/30 23:20:12 INFO: 8e9e05c5216>
Nov 30 23:20:12 server1 etcd[37691]: raft2024/11/30 23:20:12 INFO: 8e9e05c5216>
Nov 30 23:20:12 server1 etcd[37691]: raft2024/11/30 23:20:12 INFO: 8e9e05c5216>
Nov 30 23:20:12 server1 etcd[37691]: raft2024/11/30 23:20:12 INFO: 8e9e05c5216>
Nov 30 23:20:12 server1 etcd[37691]: raft2024/11/30 23:20:12 INFO: raft.node: >
Nov 30 23:20:12 server1 etcd[37691]: ready to serve client requests
Nov 30 23:20:12 server1 etcd[37691]: published {Name:server1 ClientURLs:[http:>
Nov 30 23:20:12 server1 etcd[37691]: serving insecure client requests on 127.0>
Nov 30 23:20:12 server1 systemd[1]: Started etcd.service - etcd - highly-avail>
```

**FIgure 5.25: Verifying if Etcd is installed.**

**GITPUSH:**



```
rnrlope@workstation:~/CPE212_LOPE_Act13$ git add --all
rnrlope@workstation:~/CPE212_LOPE_Act13$ git commit -m "Act13 OpenStack"
[main 7101d7e] Act13 OpenStack
 6 files changed, 222 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 openStack.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
rnrlope@workstation:~/CPE212_LOPE_Act13$ git push origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (15/15), 2.07 KiB | 531.00 KiB/s, done.
Total 15 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:RenierCode/CPE212_LOPE_Act13
   5255372..7101d7e  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act13$ 
```

**GITHUB LINK:**

https://github.com/RenierCode/CPE212_LOPE_Act13.git

**Reflections:**

**Answer the following:**

1. **What are the benefits of implementing OpenStack?**
    - OpenStack is an open-source cloud computing platform that provides for both private and public, and less reliance on proprietary vendors and lowering costs. Implementing OpenStack provides organizations a scalable, robust, and flexible cloud infrastructure solution. OpenStack guarantees compatibility, supports a variety of workloads, and connects with a large number of technologies. Its modular architecture improves agility and creativity by enabling customized deployments to match particular company demands. It is also a future-ready option for contemporary IT infrastructures because of its vibrant community, which guarantees ongoing development, security updates, and access to a multitude of resources.

**Conclusions:**

In this activity, I manage to create and demonstrate how to automate the process of Installing OpenStack Prerequisites using Ansible Playbooks and also using roles to group related tasks. I created a playbook named “openStack.yml” that will run the tasks inside the specified roles such as “base”, “controller”, and “computer”. Utilizing roles makes sure that the task will run smoothly due to segregating related tasks either based on groups, operating systems, or even types of packages. Ansible Roles provides reusability and flexibility of managing various tasks by separating the tasks into various containers also known as roles. Overall I manage to accomplish the objectives that are provided in this activity and at the same time further my proficiency in using ansible by learning from it.