Name: Khelvin P. Nicolas	**Date Performed: 12/2/2024**
Course/Section: CPE 212 - CPE31S2	**Date Submitted: 12/2/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem - 3rd Yr.**

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)

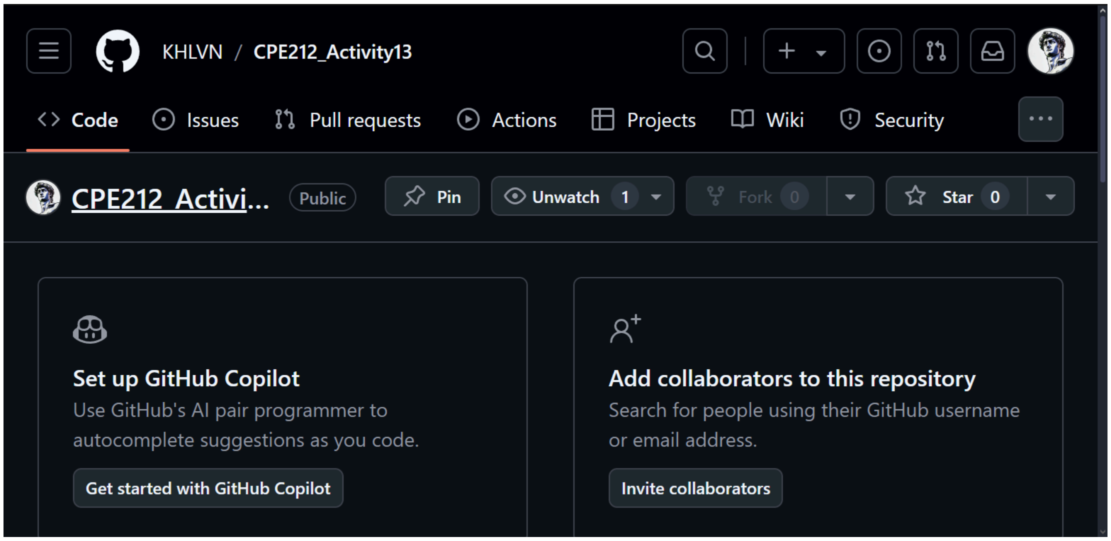
1x Ubuntu VM or Centos VM

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

5. Output (screenshots and explanations)

Setting up the Ansible and Configuration Files



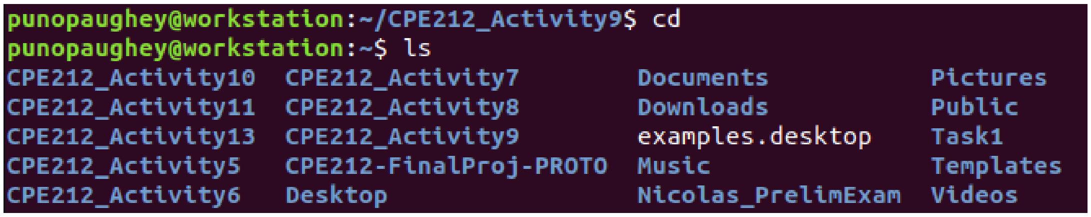
Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

Search for people using their GitHub username or email address.

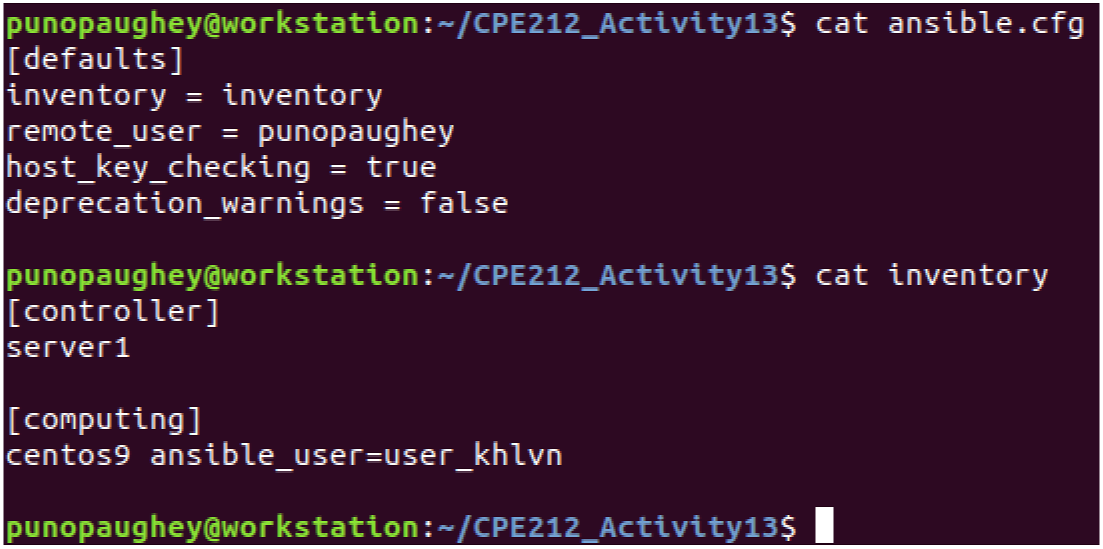
Figure 5.1: Creating GitHub repository for this activity



```
punopaughey@workstation:~/CPE212_Activity9$ cd
punopaughey@workstation:~$ ls
CPE212_Activity10  CPE212_Activity7        Documents          Pictures
CPE212_Activity11  CPE212_Activity8        Downloads          Public
CPE212_Activity13  CPE212_Activity9        examples.desktop   Task1
CPE212_Activity5   CPE212-FinalProj-PROTO  Music              Templates
CPE212_Activity6   Desktop                 Nicolas_PrelimExam Videos
```

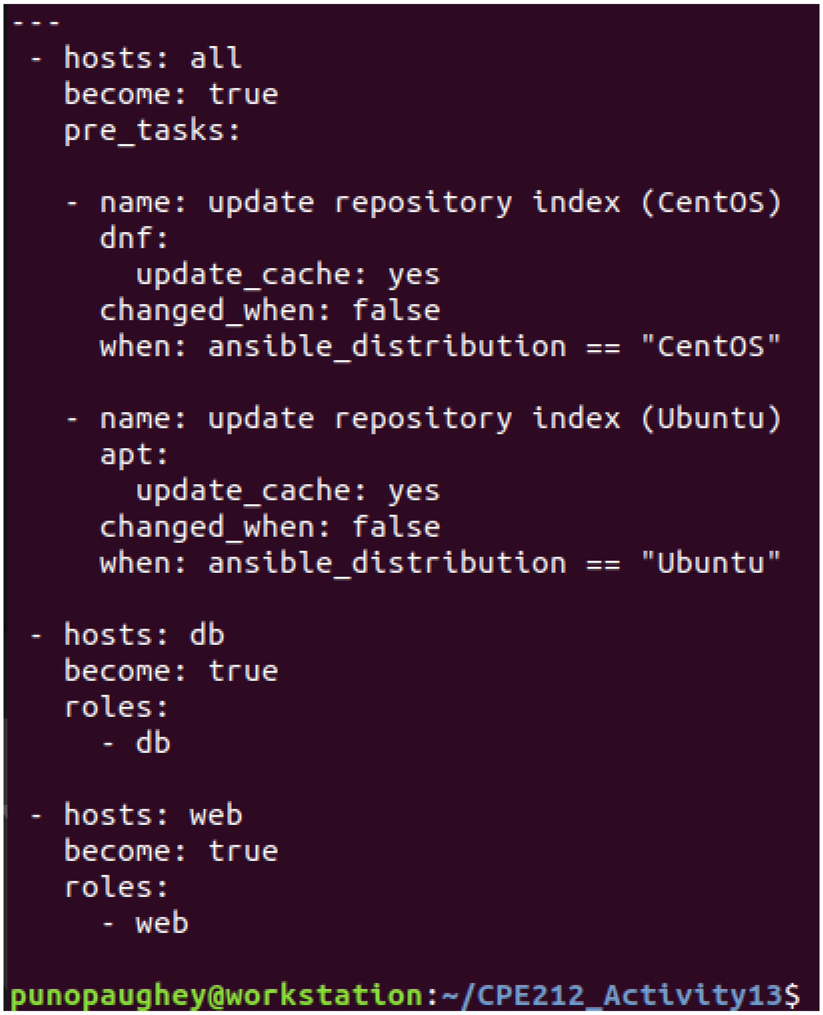
Figure 5.2: list directory of /home/punopaughey

After that, we should clone the empty repository in our local machine. I already cloned the repository prior to the creation of this document.



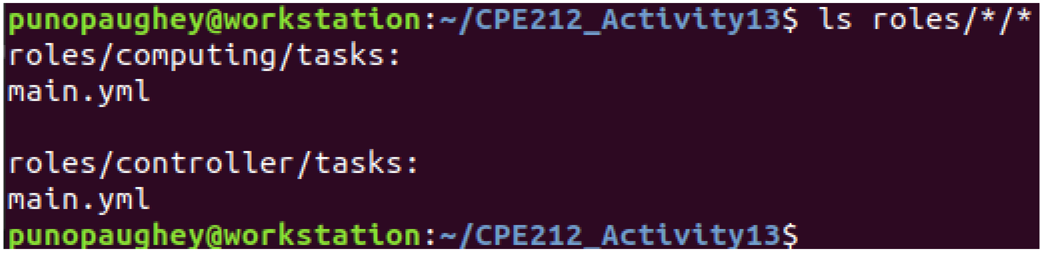
```
punopaughey@workstation:~/CPE212_Activity13$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = punopaughey
host_key_checking = true
deprecation_warnings = false

punopaughey@workstation:~/CPE212_Activity13$ cat inventory
[controller]
server1

[computing]
centos9 ansible_user=user_khlvn

punopaughey@workstation:~/CPE212_Activity13$ 
```

Figure 5.3: Creation of ansible.cfg and inventory files inside the repository



```
---
 - hosts: all
   become: true
   pre_tasks:

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

   - name: update repository index (Ubuntu)
     apt:
       update_cache: yes
     changed_when: false
     when: ansible_distribution == "Ubuntu"

 - hosts: db
   become: true
   roles:
     - db

 - hosts: web
   become: true
   roles:
     - web

punopaughey@workstation:~/CPE212_Activity13$
```

Figure 5.4: Creation of setup.yml file

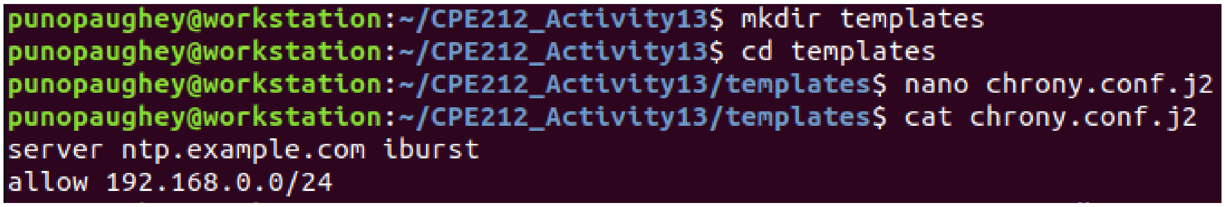


```
punopaughey@workstation:~/CPE212_Activity13$ ls roles/*/*
roles/computing/tasks:
main.yml

roles/controller/tasks:
main.yml
punopaughey@workstation:~/CPE212_Activity13$
```

Figure 5.5: Creation of groups controller and computing

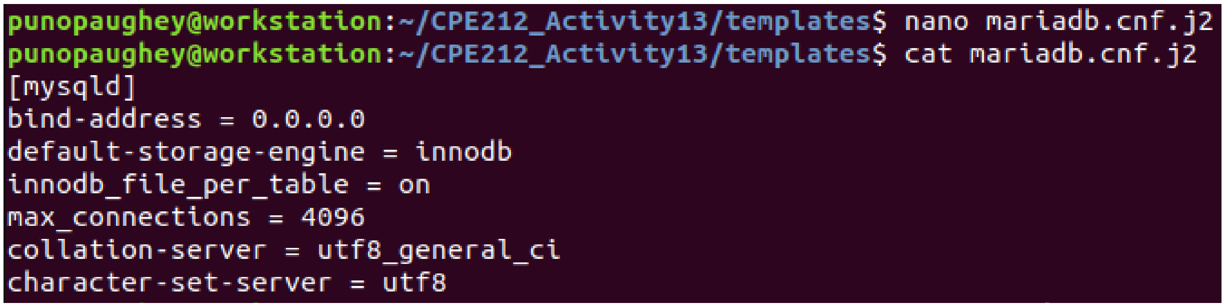
Figure 5.5 shows the structure of directories needed for the groups of ***controller*** and ***computing***. Each group has a tasks directory and each of them contains a main.yml file.

We must create a template directory to put inside the configuration files required for the applications of OpenStack (chrony configuration, mariadb configuration, etcd).



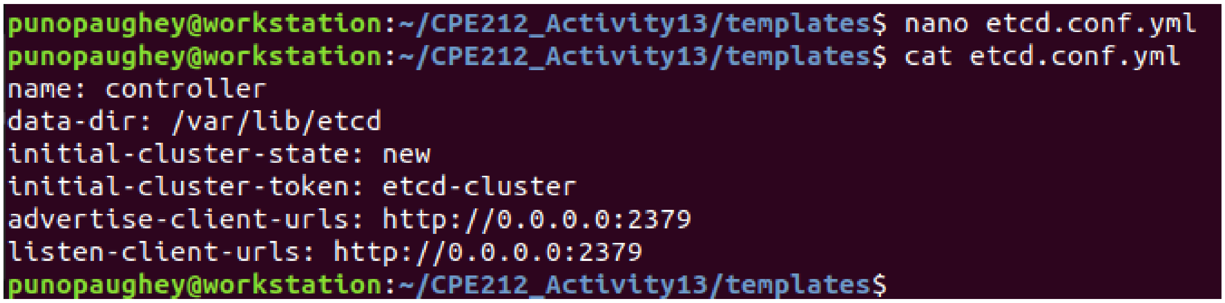
```
punopaughey@workstation:~/CPE212_Activity13$ mkdir templates
punopaughey@workstation:~/CPE212_Activity13$ cd templates
punopaughey@workstation:~/CPE212_Activity13/templates$ nano chrony.conf.j2
punopaughey@workstation:~/CPE212_Activity13/templates$ cat chrony.conf.j2
server ntp.example.com iburst
allow 192.168.0.0/24
```

Figure 5.6.1: Creating templates directory and writing chrony.conf.j2 config file



```
punopaughey@workstation:~/CPE212_Activity13/templates$ nano mariadb.cnf.j2
punopaughey@workstation:~/CPE212_Activity13/templates$ cat mariadb.cnf.j2
[mysqld]
bind-address = 0.0.0.0
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Figure 5.6.2: Creating MariaDB config file

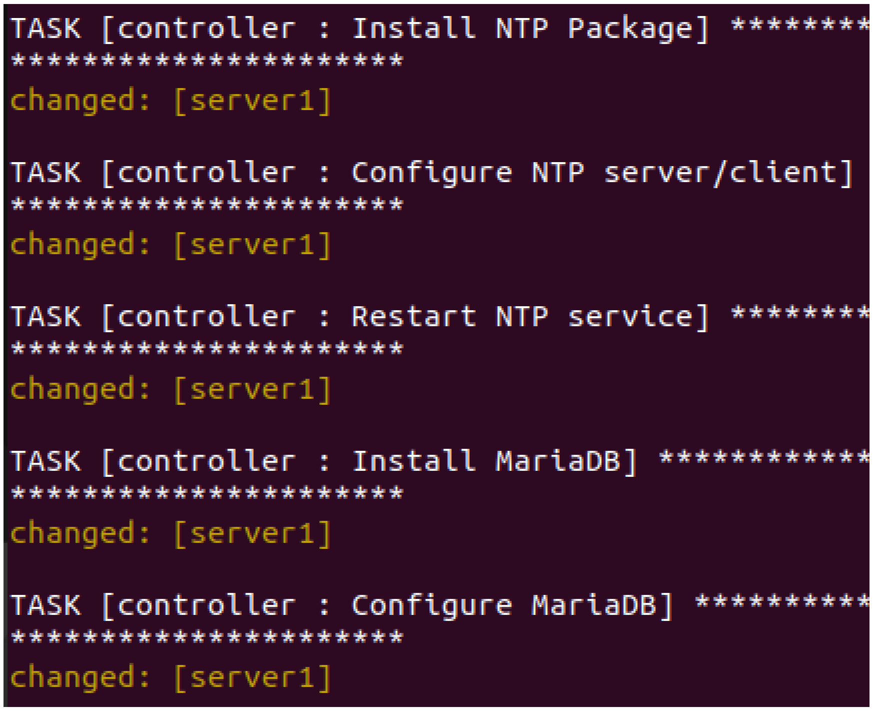


```
punopaughey@workstation:~/CPE212_Activity13/templates$ nano etcd.conf.yml
punopaughey@workstation:~/CPE212_Activity13/templates$ cat etcd.conf.yml
name: controller
data-dir: /var/lib/etcd
initial-cluster-state: new
initial-cluster-token: etcd-cluster
advertise-client-urls: http://0.0.0.0:2379
listen-client-urls: http://0.0.0.0:2379
punopaughey@workstation:~/CPE212_Activity13/templates$
```

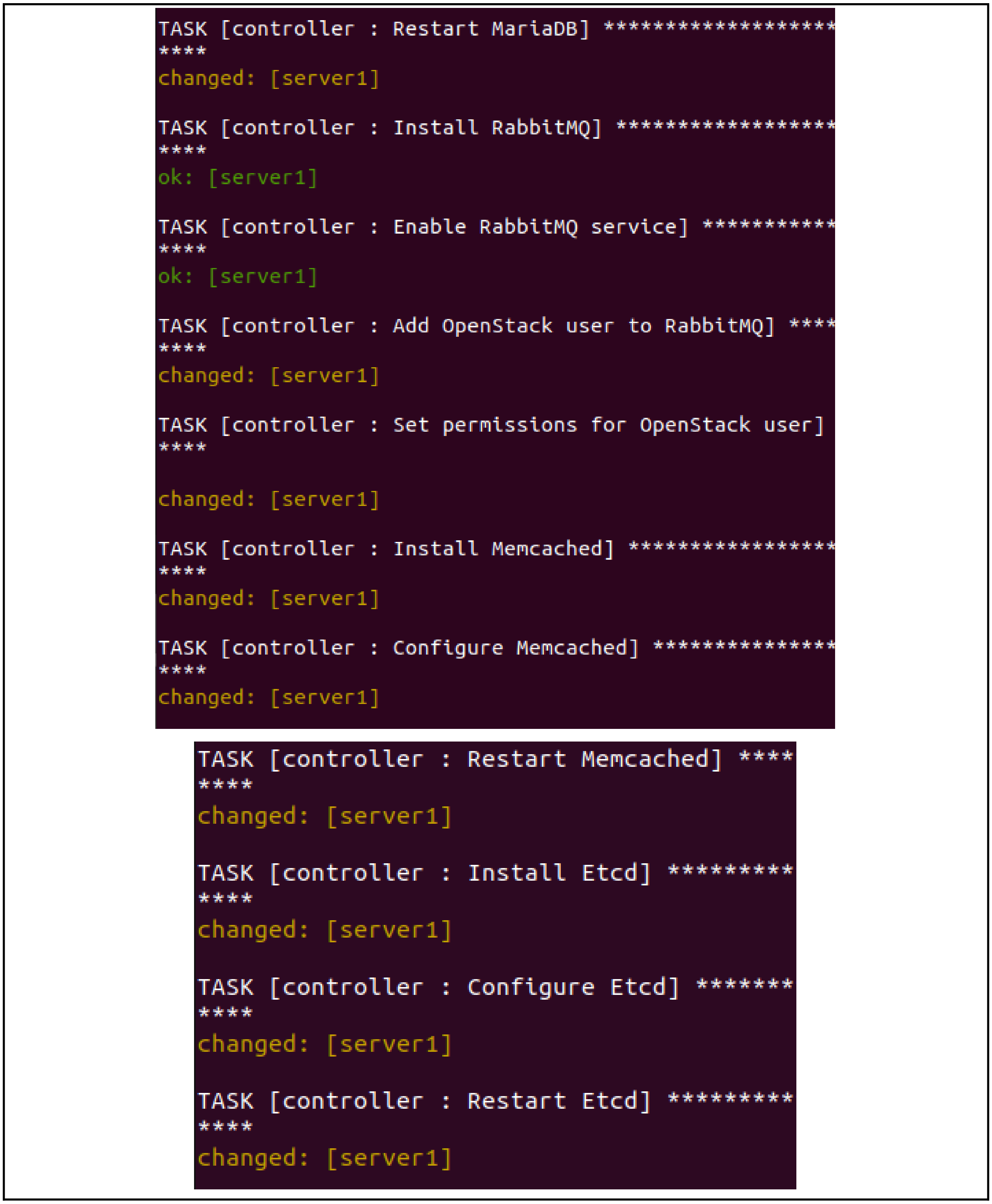
Figure 5.6.3: Creating Etcd config file

6. Running the Playbook

roles/controller/tasks/main.yml
roles/computing/tasks/main.yml

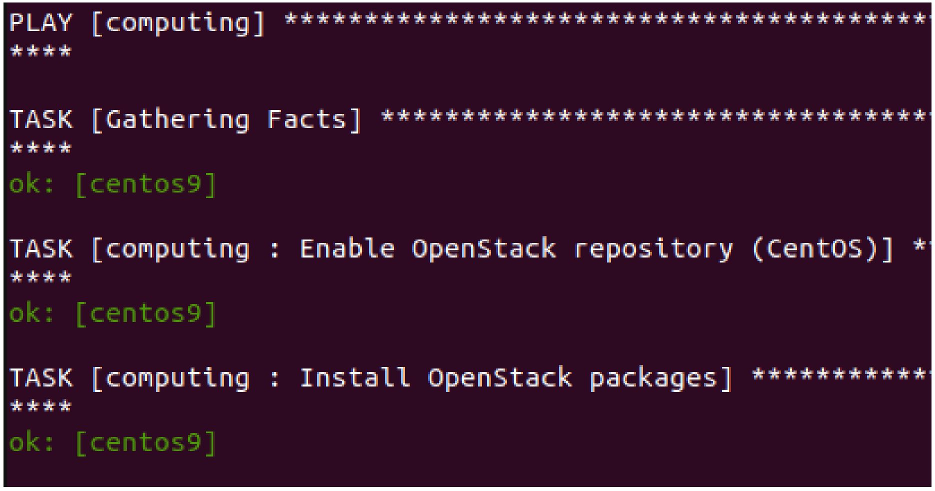


```
TASK [controller : Install NTP Package] ********
**********************
changed: [server1]

TASK [controller : Configure NTP server/client]
**********************
changed: [server1]

TASK [controller : Restart NTP service] ********
**********************
changed: [server1]

TASK [controller : Install MariaDB] ************
**********************
changed: [server1]

TASK [controller : Configure MariaDB] **********
**********************
changed: [server1]
```



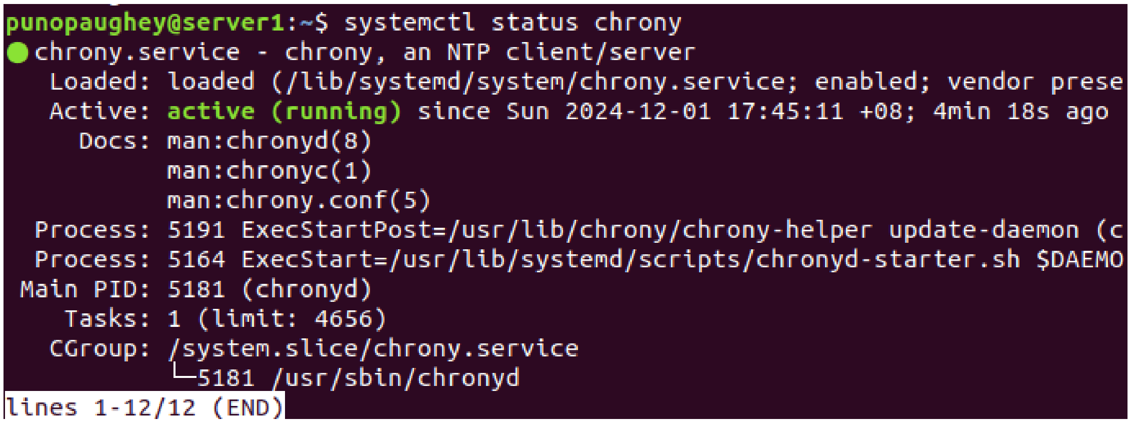
```
TASK [controller : Restart MariaDB] *******************
****
changed: [server1]

TASK [controller : Install RabbitMQ] ******************
****
ok: [server1]

TASK [controller : Enable RabbitMQ service] ***********
****
ok: [server1]

TASK [controller : Add OpenStack user to RabbitMQ] ****
****
changed: [server1]

TASK [controller : Set permissions for OpenStack user]
****

changed: [server1]

TASK [controller : Install Memcached] *****************
****
changed: [server1]

TASK [controller : Configure Memcached] ***************
****
changed: [server1]
```

```
TASK [controller : Restart Memcached] ****
****
changed: [server1]

TASK [controller : Install Etcd] *********
****
changed: [server1]

TASK [controller : Configure Etcd] *******
****
changed: [server1]

TASK [controller : Restart Etcd] *********
****
changed: [server1]
```



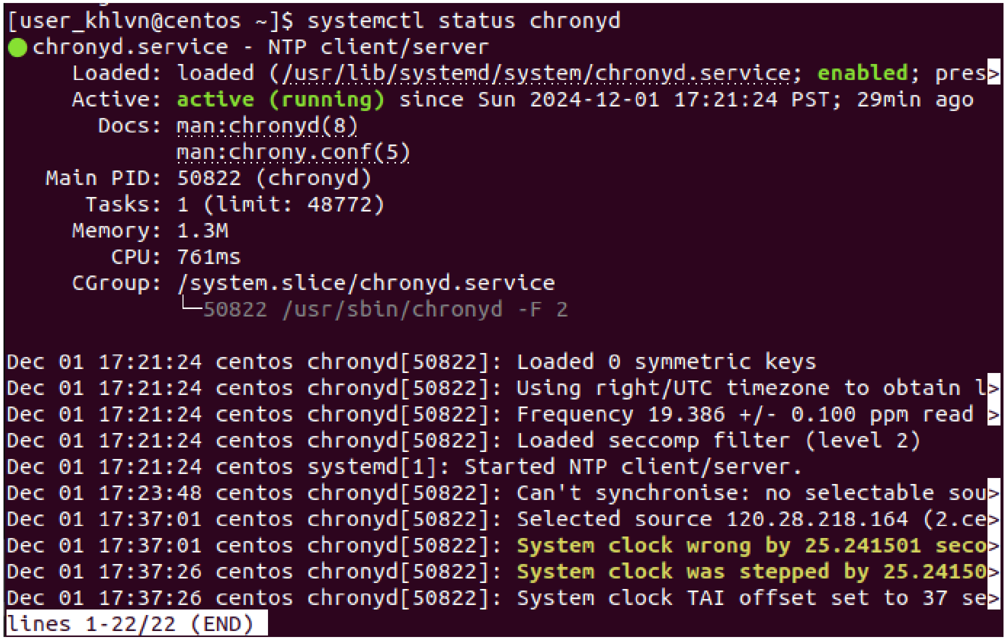
```
PLAY [computing] ******************************************
****

TASK [Gathering Facts] ************************************
****
ok: [centos9]

TASK [computing : Enable OpenStack repository (CentOS)] *
****
ok: [centos9]

TASK [computing : Install OpenStack packages] ************
****
ok: [centos9]
```

7. Verifying Services



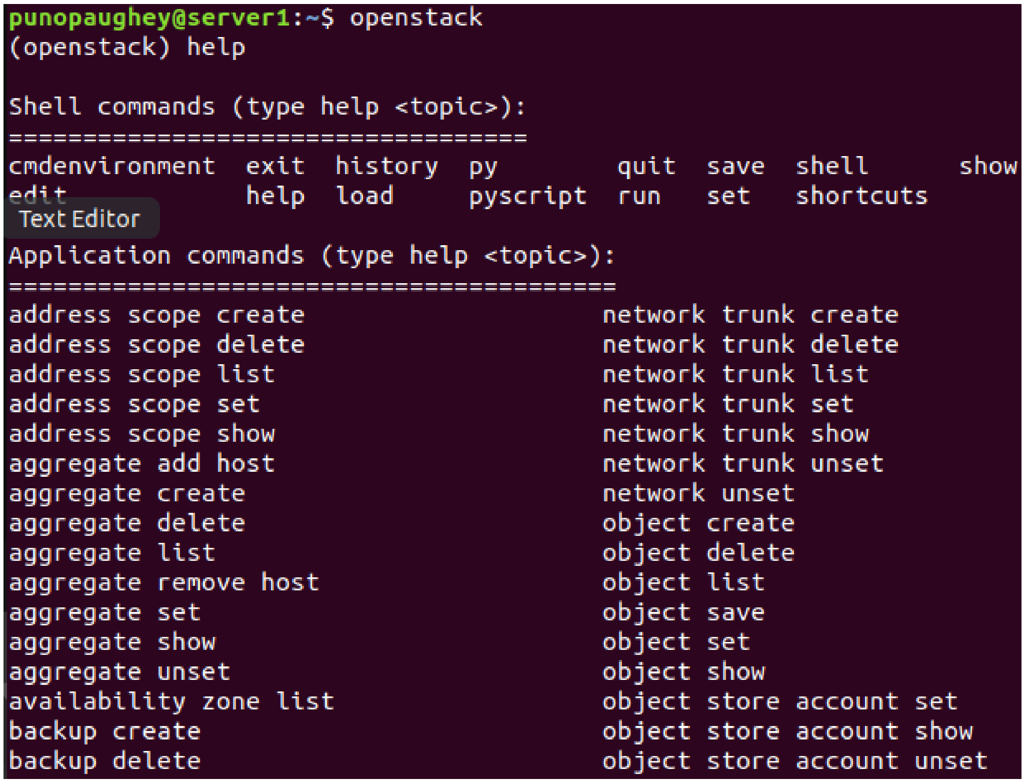
```
punopaughey@server1:~$ systemctl status chrony
● chrony.service - chrony, an NTP client/server
   Loaded: loaded (/lib/systemd/system/chrony.service; enabled; vendor prese
   Active: active (running) since Sun 2024-12-01 17:45:11 +08; 4min 18s ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
  Process: 5191 ExecStartPost=/usr/lib/chrony/chrony-helper update-daemon (c
  Process: 5164 ExecStart=/usr/lib/systemd/scripts/chronyd-starter.sh $DAEMO
 Main PID: 5181 (chronyd)
    Tasks: 1 (limit: 4656)
   CGroup: /system.slice/chrony.service
           └─5181 /usr/sbin/chronyd
lines 1-12/12 (END)
```

Figure 7.1.1: Verifying Chrony on Controller node



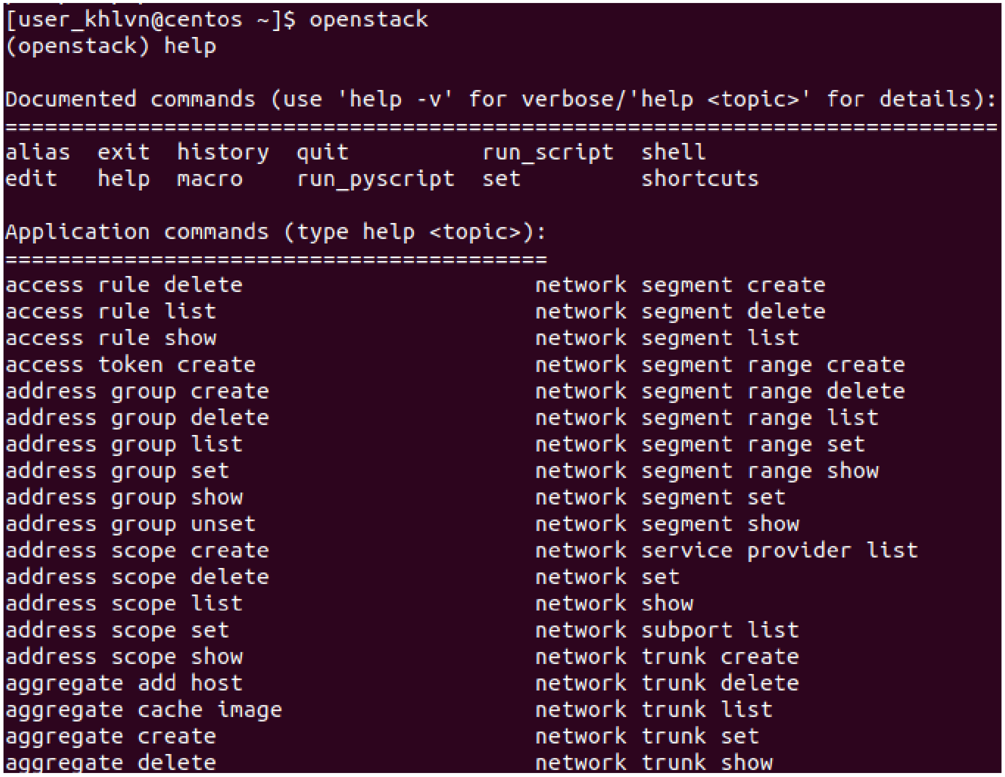
```
[user_khlvn@centos ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
     Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; pres>
     Active: active (running) since Sun 2024-12-01 17:21:24 PST; 29min ago
       Docs: man:chronyd(8)
             man:chrony.conf(5)
   Main PID: 50822 (chronyd)
      Tasks: 1 (limit: 48772)
     Memory: 1.3M
        CPU: 761ms
     CGroup: /system.slice/chronyd.service
             └─50822 /usr/sbin/chronyd -F 2

Dec 01 17:21:24 centos chronyd[50822]: Loaded 0 symmetric keys
Dec 01 17:21:24 centos chronyd[50822]: Using right/UTC timezone to obtain l>
Dec 01 17:21:24 centos chronyd[50822]: Frequency 19.386 +/- 0.100 ppm read >
Dec 01 17:21:24 centos chronyd[50822]: Loaded seccomp filter (level 2)
Dec 01 17:21:24 centos systemd[1]: Started NTP client/server.
Dec 01 17:23:48 centos chronyd[50822]: Can't synchronise: no selectable sou>
Dec 01 17:37:01 centos chronyd[50822]: Selected source 120.28.218.164 (2.ce>
Dec 01 17:37:01 centos chronyd[50822]: System clock wrong by 25.241501 seco>
Dec 01 17:37:26 centos chronyd[50822]: System clock was stepped by 25.24150>
Dec 01 17:37:26 centos chronyd[50822]: System clock TAI offset set to 37 se>
lines 1-22/22 (END)
```

Figure 7.1.2: Verifying Chrony on Compute node



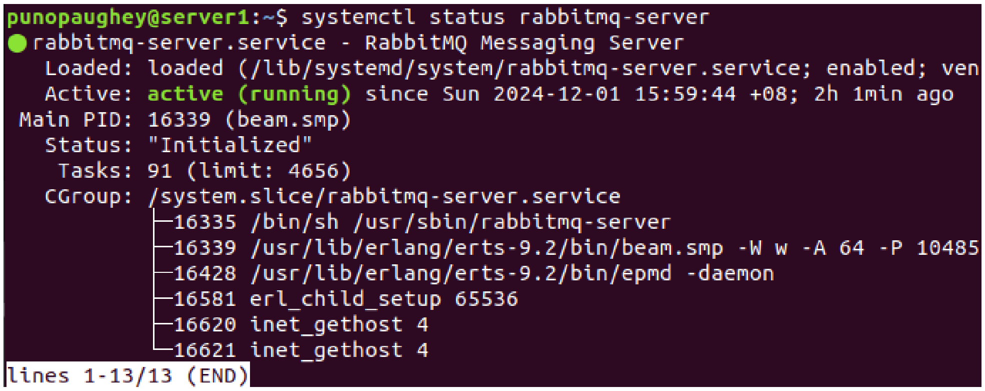
```
punopaughey@server1:~$ openstack
(openstack) help

Shell commands (type help <topic>):
===================================
cmdenvironment  exit  history  py         quit  save  shell      show
edit            help  load     pyscript   run   set   shortcuts

Application commands (type help <topic>):
=========================================
address scope create                    network trunk create
address scope delete                    network trunk delete
address scope list                      network trunk list
address scope set                       network trunk set
address scope show                      network trunk show
aggregate add host                      network trunk unset
aggregate create                        network unset
aggregate delete                        object create
aggregate list                          object delete
aggregate remove host                   object list
aggregate set                           object save
aggregate show                          object set
aggregate unset                         object show
availability zone list                  object store account set
backup create                           object store account show
backup delete                           object store account unset
```

Figure 7.2.1: Verifying OpenStack in Controller node



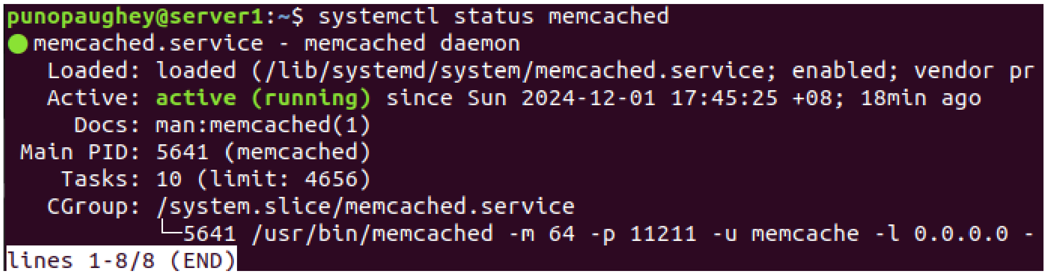
```
[user_khlvn@centos ~]$ openstack
(openstack) help

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 segment create
access rule list                        network segment delete
access rule show                        network segment list
access token create                     network segment range create
address group create                    network segment range delete
address group delete                    network segment range list
address group list                      network segment range set
address group set                       network segment range show
address group show                      network segment set
address group unset                     network segment show
address scope create                    network service provider list
address scope delete                    network set
address scope list                      network show
address scope set                       network subport list
address scope show                      network trunk create
aggregate add host                      network trunk delete
aggregate cache image                   network trunk list
aggregate create                        network trunk set
aggregate delete                        network trunk show
```

Figure 7.2.2: Verifying OpenStack in Compute node



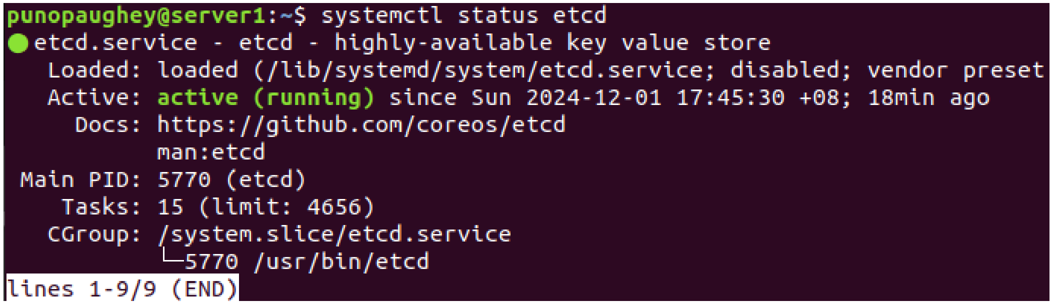
```
punopaughey@server1:~$ systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ Messaging Server
   Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; ven
   Active: active (running) since Sun 2024-12-01 15:59:44 +08; 2h 1min ago
 Main PID: 16339 (beam.smp)
   Status: "Initialized"
    Tasks: 91 (limit: 4656)
   CGroup: /system.slice/rabbitmq-server.service
           ├─16335 /bin/sh /usr/sbin/rabbitmq-server
           ├─16339 /usr/lib/erlang/erts-9.2/bin/beam.smp -W w -A 64 -P 10485
           ├─16428 /usr/lib/erlang/erts-9.2/bin/epmd -daemon
           ├─16581 erl_child_setup 65536
           ├─16620 inet_gethost 4
           └─16621 inet_gethost 4
lines 1-13/13 (END)
```

Figure 7.3: Verifying RabbitMQ in Controller node



```
punopaughey@server1:~$ systemctl status memcached
● memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor pr
   Active: active (running) since Sun 2024-12-01 17:45:25 +08; 18min ago
     Docs: man:memcached(1)
 Main PID: 5641 (memcached)
    Tasks: 10 (limit: 4656)
   CGroup: /system.slice/memcached.service
           └─5641 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 0.0.0.0 -
lines 1-8/8 (END)
```

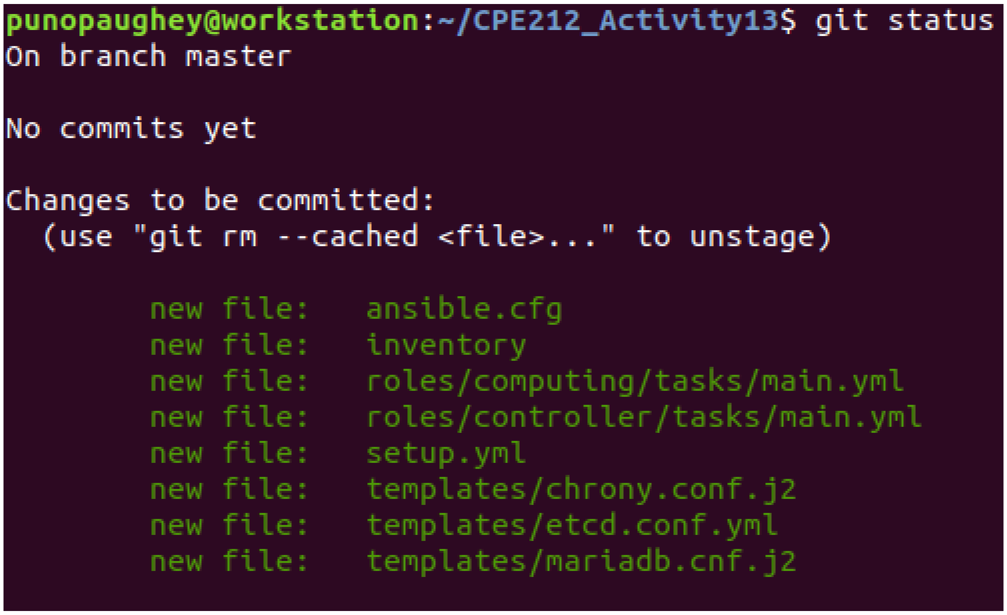
Figure 7.4: Verifying Memcached in Controller node



```
punopaughey@server1:~$ systemctl status etcd
● etcd.service - etcd - highly-available key value store
   Loaded: loaded (/lib/systemd/system/etcd.service; disabled; vendor preset
   Active: active (running) since Sun 2024-12-01 17:45:30 +08; 18min ago
     Docs: https://github.com/coreos/etcd
           man:etcd
 Main PID: 5770 (etcd)
    Tasks: 15 (limit: 4656)
   CGroup: /system.slice/etcd.service
           └─5770 /usr/bin/etcd
lines 1-9/9 (END)
```

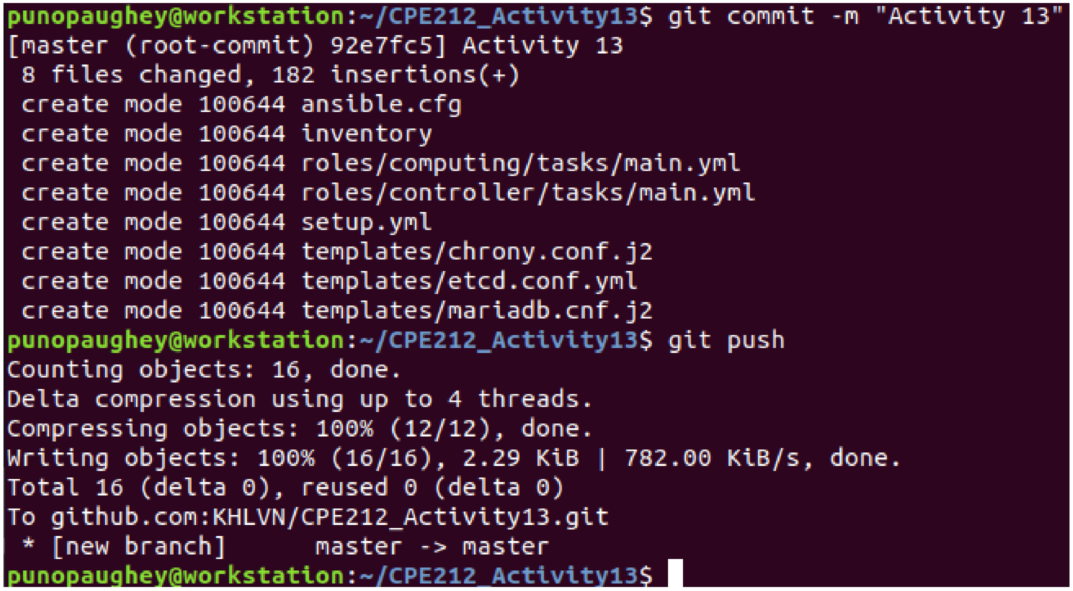
Figure 7.5: Verifying Etcd in Controller node

8. GitHub



```
punopaughey@workstation:~/CPE212_Activity13$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   ansible.cfg
        new file:   inventory
        new file:   roles/computing/tasks/main.yml
        new file:   roles/controller/tasks/main.yml
        new file:   setup.yml
        new file:   templates/chrony.conf.j2
        new file:   templates/etcd.conf.yml
        new file:   templates/mariadb.cnf.j2
```

Figure 8.1: Git Status



```
punopaughey@workstation:~/CPE212_Activity13$ git commit -m "Activity 13"
[master (root-commit) 92e7fc5] Activity 13
 8 files changed, 182 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 roles/computing/tasks/main.yml
 create mode 100644 roles/controller/tasks/main.yml
 create mode 100644 setup.yml
 create mode 100644 templates/chrony.conf.j2
 create mode 100644 templates/etcd.conf.yml
 create mode 100644 templates/mariadb.cnf.j2
punopaughey@workstation:~/CPE212_Activity13$ git push
Counting objects: 16, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 2.29 KiB | 782.00 KiB/s, done.
Total 16 (delta 0), reused 0 (delta 0)
To github.com:KHLVN/CPE212_Activity13.git
 * [new branch]      master -> master
punopaughey@workstation:~/CPE212_Activity13$ 
```

Figure 8.2: Committing, and Pushing to the remote repository.

GitHub:

https://github.com/KHLVN/CPE212_Activity13

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?
   - Implementing OpenStack provides numerous benefits for organizations seeking to build and manage cloud infrastructure. As an open-source platform, OpenStack significantly reduces costs, as it eliminates licensing fees typically associated with proprietary cloud services like AWS or Microsoft Azure. Moreover, OpenStack's flexibility allows organizations to scale their infrastructure as needed, accommodating growth without the need for costly hardware investments. The platform's multi-vendor support enables businesses to avoid vendor lock-in, empowering them to choose hardware from different suppliers, thus optimizing cost and performance.

Conclusions:

- In conclusion, working through the OpenStack playbook and troubleshooting the issues that arose during its implementation has been a valuable learning experience in my system administration course. Through this project, I gained hands-on knowledge of setting up key OpenStack components like MariaDB, RabbitMQ, Memcached, and Etcd on both CentOS and Ubuntu servers. The task required me to understand the architecture of OpenStack and its dependencies, which was key to successfully configuring the environment.