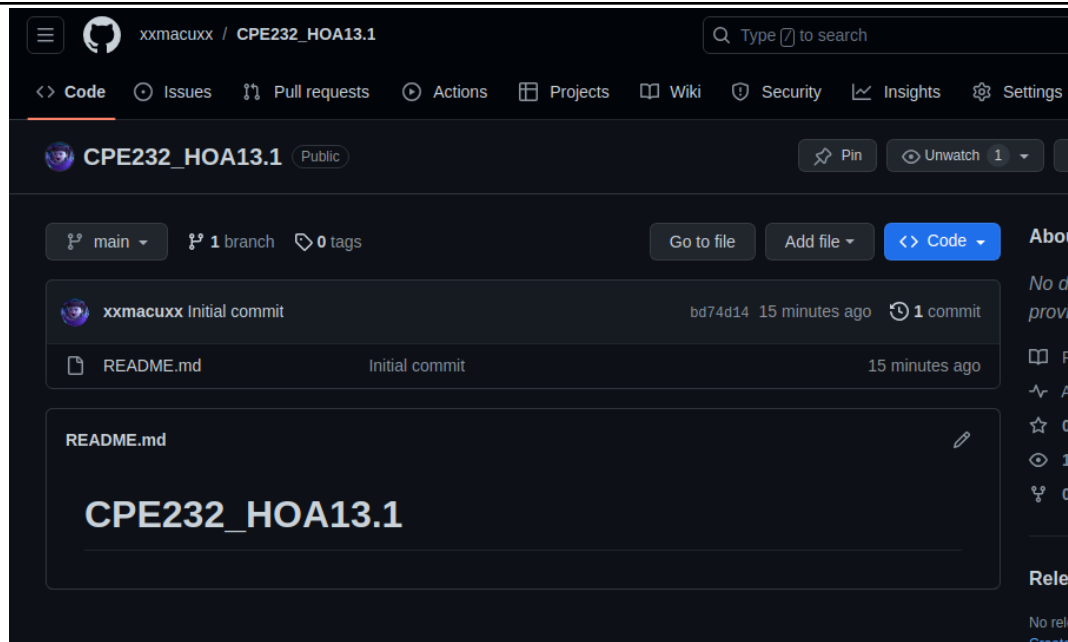


Name: Jaira Biane Maculada	Date Performed: 11/27/23
Course/Section: CPE232/CPE31S6	Date Submitted: 11/28/23
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st Sem(2023-2024)
Activity 13: OpenStack Prerequisite Installation	
1. Objectives	
Create a workflow to install OpenStack using Ansible as your Infrastructure as Code (IaC).	
2. Intended Learning Outcomes	
<ol style="list-style-type: none"> 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	
<p>Oracle VirtualBox (Hypervisor)</p> <p>1x Ubuntu VM or Centos VM</p>	
4. Tasks	
<ol style="list-style-type: none"> 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/ <ol style="list-style-type: none"> 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)	
Task 1: Create a File	
<ol style="list-style-type: none"> 1. Create a new repository for this Hands-On Activity. 	



```
jai@workstation: ~/CPE232_HOA13.1
jai@workstation:~$ git clone git@github.com:xxmacuxx/CPE232_HOA13.1.git
Cloning into 'CPE232_HOA13.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
jai@workstation:~$ cd CPE232_HOA13.1
```

2. Create the ansible.cfg and inventory file (*must include one Ubuntu*)

```
jai@workstation: ~/CPE232_HOA13.1
GNU nano 6.2 ansible.cfg *
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = jai
private_key_file = ~/.ssh/
```

```
jai@workstation: ~/CPE232_HOA13.1
GNU nano 6.2 inventory *
[controller]
192.168.56.110
```

Task 2: Create Playbook for Installing OpenStack

1. Create a playbook and name it install_openstack.yml.

```
jai@workstation: ~/CPE232_HOA13.1
GNU nano 6.2 install openstack.yml *
---
- hosts: all
  become: true
  pre_tasks:
    - name: Install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
        changed_when: false
- hosts: controller
  become: true
  roles:
    - NTP
    - OpenStack
    - SQL
    - MessageQ
    - Memcached
    - Etcd
```

Code explanation:

<p>It instructs Ansible to run tasks on all hosts, become a privileged user, and execute a pre-task. The pre-task installs updates on Ubuntu using the 'apt' module, specifying a distribution upgrade while suppressing changes indication for brevity.</p>	<pre> GNU nano 6.2 install openstack.yml --- - hosts: all become: true pre_tasks: - name: Install updates (Ubuntu) apt: upgrade: dist update_cache: yes changed_when: false </pre>
<p>This Ansible playbook configures a server (controller) to take on various roles, including managing NTP, deploying OpenStack, handling SQL databases, dealing with message queues, utilizing Memcached, and using Etcd for distributed key-value storage. The 'become: true' grants elevated privileges for execution.</p>	<pre> - hosts: controller become: true roles: - NTP - OpenStack - SQL - MessageQ - Memcached - Etcd </pre>

2. Save the file and exit.

Task 3: Create Roles

1. Create a new directory and name it "roles". Enter the roles directory and create new directories: NTP, OpenStack, SQL, MessageQ, Memcached, Etcd For each directory, create a directory and name it tasks.

For NTP

```

jai@workstation:~/CPE232_H0A13.1/roles$ mkdir NTP
jai@workstation:~/CPE232_H0A13.1/roles$ cd NTP
jai@workstation:~/CPE232_H0A13.1/roles/NTP$ mkdir tasks
jai@workstation:~/CPE232_H0A13.1/roles/NTP$ cd tasks

```

For OpenStack

```
jai@workstation:~/CPE232_H0A13.1/roles$ mkdir OpenStack
jai@workstation:~/CPE232_H0A13.1/roles$ cd OpenStack
jai@workstation:~/CPE232_H0A13.1/roles/OpenStack$ mkdir tasks
```

```
jai@workstation:~/CPE232_H0A13.1/roles/OpenStack$ cd tasks
```

For SQL

```
jai@workstation:~/CPE232_H0A13.1/roles$ mkdir SQL
jai@workstation:~/CPE232_H0A13.1/roles$ cd SQL
jai@workstation:~/CPE232_H0A13.1/roles/SQL$ mkdir tasks
jai@workstation:~/CPE232_H0A13.1/roles/SQL$ cd tasks
```

For MessageQ

```
jai@workstation:~/CPE232_H0A13.1/roles$ mkdir MessageQ
jai@workstation:~/CPE232_H0A13.1/roles$ cd MessageQ
jai@workstation:~/CPE232_H0A13.1/roles/MessageQ$ mkdir tasks
jai@workstation:~/CPE232_H0A13.1/roles/MessageQ$ cd tasks
```

For Memcached

```
jai@workstation:~/CPE232_H0A13.1/roles$ mkdir Memcached
jai@workstation:~/CPE232_H0A13.1/roles$ cd Memcached
jai@workstation:~/CPE232_H0A13.1/roles/Memcached$ mkdir tasks
jai@workstation:~/CPE232_H0A13.1/roles/Memcached$ cd tasks
jai@workstation:~/CPE232_H0A13.1/roles/Memcached/tasks$
```

For Etcd

```
jai@workstation:~/CPE232_H0A13.1/roles$ mkdir Etcd
jai@workstation:~/CPE232_H0A13.1/roles$ cd Etcd
jai@workstation:~/CPE232_H0A13.1/roles/Etcd$ mkdir tasks
jai@workstation:~/CPE232_H0A13.1/roles/Etcd$ cd tasks
```

2. In each of the tasks for the two directory (*centos_elk* and *ubuntu_elk*), create another file and name it *main.yml*.

For NTP

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd NTP/tasks
jai@workstation:~/CPE232_H0A13.1/roles/NTP/tasks$ sudo nano main.yml
```

For OpenStack

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd OpenStack/tasks
jai@workstation:~/CPE232_H0A13.1/roles/OpenStack/tasks$ sudo nano main.yml
```

For SQL

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd SQL/tasks
jai@workstation:~/CPE232_H0A13.1/roles/SQL/tasks$ sudo nano main.yml
```

For MessageQ

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd MessageQ/tasks
jai@workstation:~/CPE232_H0A13.1/roles/MessageQ/tasks$ sudo nano main.yml
```

For Memcached

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd Memcached/tasks
jai@workstation:~/CPE232_H0A13.1/roles/Memcached/tasks$ sudo nano main.yml
```

For Etcd

```
jai@workstation:~/CPE232_H0A13.1/roles$ cd Etcd/tasks
jai@workstation:~/CPE232_H0A13.1/roles/Etcd/tasks$ sudo nano main.yml
```

Tree for roles

```
jai@workstation:~/CPE232_H0A13.1/roles$ tree
.
├── Etcd
│   └── tasks
│       └── main.yml
├── Memcached
│   └── tasks
│       └── main.yml
├── MessageQ
│   └── tasks
│       └── main.yml
├── NTP
│   └── tasks
│       └── main.yml
├── OpenStack
│   └── tasks
│       └── main.yml
└── SQL
    └── tasks
        └── main.yml

12 directories, 6 files
```

3. Copy the code to the main.yml of the each subdirectory.

For NTP

main CPE232_HOA13.1 / roles / NTP / tasks / main.yml

xxmacuxx OpenStack Prerequisite Installation

Code Blame 11 lines (10 loc) · 237 Bytes

```
1 - name: Installing the Network Time Protocol (NTP)
2   apt:
3     name: chrony
4     state: present
5     update_cache: yes
6
7 - name: Enable the chrony
8   service:
9     name: chrony.service
10    state: restarted
11    enabled: yes
```

For OpenStack

main CPE232_HOA13.1 / roles / OpenStack / tasks / main.yml

xxmacuxx OpenStack Prerequisite Installation

Code Blame 7 lines (7 loc) · 150 Bytes

```
1 - name: Install the OpenStack Packages
2   apt:
3     name:
4       - nova-compute
5       - python3-openstackclient
6     state: present
7     update_cache: yes
```

For SQL

main

CPE232_HOA13.1 / roles / SQL / tasks / main.yml

xxmacuxx OpenStack Prerequisite Installation

Code Blame 24 lines (22 loc) · 546 Bytes

```
1  - name: Install the SQL Database
2    apt:
3      name:
4        - mariadb-server
5        - python3-pymysql
6      state: present
7      update_cache: yes
8
9  - name: Edit the maria-db.conf file
10    copy:
11      content: |
12        default-storage-engine = innodb
13        innodb_file_per_table = on
14        max_connections = 4096
15        collation-server = utf_general_ci
16        character-set-server = utf8
17      dest: /etc/mysql/mariadb.conf.d/99-openstack.cnf
18      mode: "0755"
19
20  - name: Restart the mariadb-server
21    service:
22      name: mysql
23      state: restarted
24      enabled: yes
```

For MessageQ

main CPE232_HOA13.1 / roles / MessageQ / tasks / main.yml

xxmacuuxx OpenStack Prerequisite Installation

Code Blame 11 lines (10 loc) · 212 Bytes

```
1 - name: Install Message Queue
2   apt:
3     name: rabbitmq-server
4     state: present
5     update_cache: yes
6
7 - name: Starting service
8   service:
9     name: rabbitmq-server.service
10    state: started
11    enabled: true
```

For Memcached

main CPE232_HOA13.1 / roles / Memcached / tasks / main.yml

xxmacuuxx OpenStack Prerequisite Installation

Code Blame 13 lines (12 loc) · 234 Bytes

```
1 - name: Install the Memcached
2   apt:
3     name:
4       - memcached
5       - python3-memcache
6     state: present
7     update_cache: yes
8
9 - name: Restart the Memcached
10  service:
11    name: memcached
12    state: restarted
13    enabled: yes
```

For Etcd



xxmacuxx OpenStack Prerequisite Installation

Code

Blame

25 lines (23 loc) · 695 Bytes

```
1  - name: Install the Etcd
2    apt:
3      name: etcd
4      state: present
5      update_cache: yes
6
7  - name: Edit the Etcd file
8    copy:
9      content: |
10         ETCD_NAME="controller"
11         ETCD_DATA_DIR="/var/lib/etcd"
12         ETCD_INITIAL_CLUSTER_STATE="new"
13         ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster-01"
14         ETCD_INITIAL_CLUSTER="controller=http://10.0.0.11:2380"
15         ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.0.0.11:2380"
16         ETCD_ADVERTISE_CLIENT_URLS="http://10.0.0.11:2379"
17         ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
18         ETCD_LISTEN_CLIENT_URLS="http://10.0.0.11:2379"
19     dest: /etc/default/etcd
20     mode: "0755"
21
22  - name: Enable the Etcd
23    service:
24      name: etcd
25      enabled: yes
```

Task 4: Run and Verify

1. Run the command `ansible-playbook - - ask-become-pass install_openstack.yml` to completely install the OpenStack in the controller node.

```
jai@workstation: ~/CPE232_HOA13.1
jai@workstation:~/CPE232_HOA13.1$ ansible-playbook --ask-become-pass install_openstack.y
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.110]

TASK [Install updates (Ubuntu)] *****
ok: [192.168.56.110]

PLAY [controller] *****

TASK [Gathering Facts] *****
ok: [192.168.56.110]

TASK [NTP : Installing the Network Time Protocol (NTP)] *****
ok: [192.168.56.110]

TASK [NTP : Enable the chrony] *****
changed: [192.168.56.110]

TASK [OpenStack : Install the OpenStack Packages] *****
ok: [192.168.56.110]

TASK [SQL : Install the SQL Database] *****
ok: [192.168.56.110]

TASK [SQL : Edit the maria-db.conf file] *****
ok: [192.168.56.110]

TASK [SQL : Restart the mariadb-server] *****
changed: [192.168.56.110]

TASK [MessageQ : Install Message Queue] *****
ok: [192.168.56.110]

TASK [MessageQ : Starting service] *****
ok: [192.168.56.110]

TASK [Memcached : Install the Memcached] *****
ok: [192.168.56.110]

TASK [Memcached : Restart the Memcached] *****
changed: [192.168.56.110]

TASK [Etcd : Install the Etcd] *****
ok: [192.168.56.110]

TASK [Etcd : Edit the Etcd file] *****
ok: [192.168.56.110]

TASK [Etcd : Enable the Etcd] *****
ok: [192.168.56.110]

PLAY RECAP *****
192.168.56.110      : ok=16   changed=3   unreachable=0   failed=0   skipped=0
rescued=0   ignored=0

jai@workstation:~/CPE232_HOA13.1$
```

2. Show the screenshot of the NTP, OpenStack, SQL, MessageQ, Memcached, and Etcd that are working.

NTP

```
jai@server2:~$ ntpq -p
      remote                       refid              st t when poll reach   delay   offset  jitter
=====
0.ubuntu.pool.n .POOL.             16 p   -   64    0    0.000  +0.000  0.000
1.ubuntu.pool.n .POOL.             16 p   -   64    0    0.000  +0.000  0.000
2.ubuntu.pool.n .POOL.             16 p   -   64    0    0.000  +0.000  0.000
3.ubuntu.pool.n .POOL.             16 p   -   64    0    0.000  +0.000  0.000
ntp.ubuntu.com  .POOL.             16 p   -   64    0    0.000  +0.000  0.000
185.125.190.56  194.121.207.249    2 u   61   64    1   270.667 -10.856  0.000
185.125.190.57  201.68.88.106     2 u   60   64    1   278.944 -10.846  0.000
185.125.190.58  37.15.221.189     2 u   62   64    1   289.412 -18.471  0.000
alphyn.canonica 132.163.96.1       2 u   63   64    1   255.653  +3.172  0.000
```

```
jai@server2:~$ sudo systemctl start chrony.service
jai@server2:~$ sudo systemctl status chrony.service
● chrony.service - chrony, an NTP client/server
   Loaded: loaded (/lib/systemd/system/chrony.service; enabled; vendor preset:
   Active: active (running) since Mon 2023-11-27 23:22:29 PST; 24s ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
  Main PID: 69522 (chronyd)
    Tasks: 2 (limit: 4599)
   Memory: 1.3M
      CPU: 91ms
   CGroup: /system.slice/chrony.service
           └─69522 /usr/sbin/chronyd -F 1
             └─69523 /usr/sbin/chronyd -F 1

Nov 27 23:22:29 server2 systemd[1]: Starting chrony, an NTP client/server...
Nov 27 23:22:29 server2 chronyd[69522]: chronyd version 4.2 starting (+CMDMON +>
Nov 27 23:22:29 server2 chronyd[69522]: Frequency -4.880 +/- 23.476 ppm read fr>
Nov 27 23:22:29 server2 chronyd[69522]: Using right/UTC timezone to obtain leap>
Nov 27 23:22:29 server2 chronyd[69522]: Loaded seccomp filter (level 1)
Nov 27 23:22:29 server2 systemd[1]: Started chrony, an NTP client/server.
Nov 27 23:22:36 server2 chronyd[69522]: Selected source 91.189.91.157 (ntp.ubun>
Nov 27 23:22:36 server2 chronyd[69522]: System clock TAI offset set to 37 secon>
lines 1-22/22 (END)...skipping...
```

OpenStack

```
jai@server2:~$ dpkg -l | grep openstack
ii python3-openstackclient      5.8.0-0ubuntu1
   all OpenStack Command-line Client - Python 3.x
ii python3-openstacksdk         0.61.0-0ubuntu1
   all SDK for building applications to work with OpenStack - Python 3.x
jai@server2:~$
```

SQL

```
jai@server2:~$ sudo systemctl status mysql
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-27 23:40:18 PST; 51s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 72938 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 12 (limit: 4599)
    Memory: 61.0M
       CPU: 547ms
    CGroup: /system.slice/mariadb.service
            └─72938 /usr/sbin/mariabdd

Nov 27 23:40:18 server2 systemd[1]: Started MariaDB 10.6.12 database server.
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72953]: Upgrading MySQL tables if necessary.
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: error: Found option without preceding group
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: Looking for 'mariadb' as: /usr/bin/mariadb
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: This installation of MariaDB is already up-to-date
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: There is no need to run mysql_upgrade again
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72956]: You can use --force if you still want to run the upgrade
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72964]: Checking for insecure root accounts.
Nov 27 23:40:18 server2 /etc/mysql/debian-start[72968]: Triggering myisam-recover for all MyISAM tables
lines 1-23/23 (END)
```

MessageQ

```
jai@server2:~$ sudo systemctl status rabbitmq-server
● rabbitmq-server.service - RabbitMQ Messaging Server
   Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-27 23:06:58 PST; 40min ago
     Main PID: 64361 (beam.smp)
        Tasks: 23 (limit: 4599)
    Memory: 86.8M
       CPU: 24.970s
    CGroup: /system.slice/rabbitmq-server.service
            └─64361 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas ageffcbf -MHas ageffcbf -
              └─64372 erl_child_setup 65536
                └─64423 inet_gethost 4
                  └─64424 inet_gethost 4

Nov 27 23:06:49 server2 systemd[1]: Starting RabbitMQ Messaging Server...
Nov 27 23:06:58 server2 systemd[1]: Started RabbitMQ Messaging Server.
lines 1-15/23 (END)
```

Memcached

```
jai@server2:~$ sudo systemctl start memcached.service
jai@server2:~$ sudo systemctl status memcached.service
● memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-27 23:07:23 PST; 18min ago
     Docs: man:memcached(1)
  Main PID: 65312 (memcached)
     Tasks: 10 (limit: 4599)
    Memory: 2.0M
       CPU: 229ms
    CGroup: /system.slice/memcached.service
            └─65312 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1 -P /var/run/memcac

Nov 27 23:07:23 server2 systemd[1]: Stopped memcached daemon.
Nov 27 23:07:23 server2 systemd[1]: Started memcached daemon.
lines 1-13/13 (END)
```

Etc

```

jai@server2:~$ sudo systemctl enable etcd
Synchronizing state of etcd.service with SysV service script with /lib/systemd/systemd-sysv-instal
.
Executing: /lib/systemd/systemd-sysv-install enable etcd
jai@server2:~$ sudo systemctl start etcd
jai@server2:~$ sudo systemctl status etcd
● etcd.service - etcd - highly-available key value store
   Loaded: loaded (/lib/systemd/system/etcd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-27 23:08:00 PST; 19min ago
     Docs: https://etcd.io/docs
    Man: etcd
   Main PID: 66130 (etcd)
      Tasks: 9 (limit: 4599)
     Memory: 6.0M
        CPU: 3.696s
    CGroup: /system.slice/etcd.service
            └─66130 /usr/bin/etcd

Nov 27 23:08:00 server2 etcd[66130]: 8e9e05c52164694d received MsgVoteResp from 8e9e05c52164694d
Nov 27 23:08:00 server2 etcd[66130]: 8e9e05c52164694d became leader at term 2
Nov 27 23:08:00 server2 etcd[66130]: raft.node: 8e9e05c52164694d elected leader 8e9e05c52164694d
Nov 27 23:08:00 server2 etcd[66130]: published {Name:server2 ClientURLs:[http://localhost:2379]}
Nov 27 23:08:00 server2 etcd[66130]: setting up the initial cluster version to 3.3
Nov 27 23:08:00 server2 etcd[66130]: ready to serve client requests
Nov 27 23:08:00 server2 etcd[66130]: servicing insecure client requests on 127.0.0.1:2379, this is
Nov 27 23:08:00 server2 systemd[1]: Started etcd - highly-available key value store.
Nov 27 23:08:00 server2 etcd[66130]: set the initial cluster version to 3.3
Nov 27 23:08:00 server2 etcd[66130]: enabled capabilities for version 3.3
lines 1-22/22 (END)

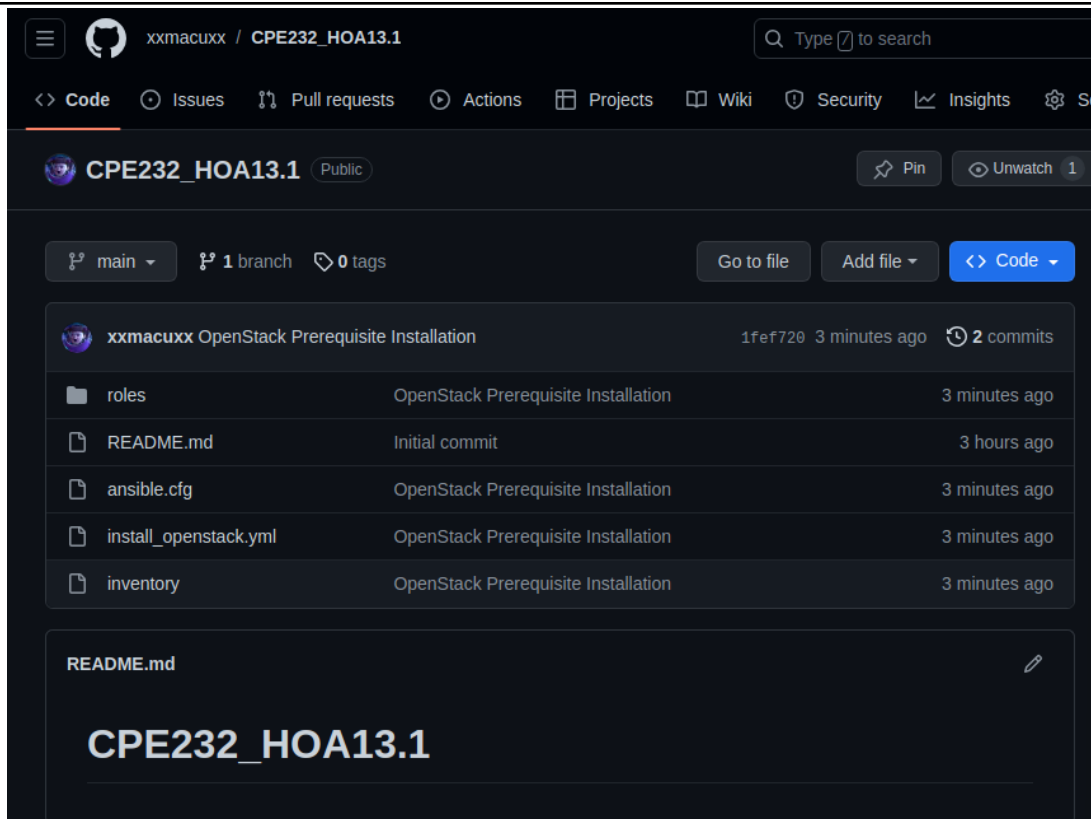
```

3. Upload it in the github.

```

jai@workstation:~/CPE232_H0A13.1$ git add *
jai@workstation:~/CPE232_H0A13.1$ git commit -m "OpenStack Prerequisite Installation"
[main 1fef720] OpenStack Prerequisite Installation
 9 files changed, 122 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 install_openstack.yml
 create mode 100644 inventory
 create mode 100644 roles/Etcd/tasks/main.yml
 create mode 100644 roles/Memcached/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
jai@workstation:~/CPE232_H0A13.1$ git push origin
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 2 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (24/24), 2.55 KiB | 261.00 KiB/s, done.
Total 24 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:xxmacuux/CPE232_H0A13.1.git
 bd74d14..1fef720  main -> main

```



GITHUB LINK: https://github.com/xxmacuxx/CPE232_HOA13.1.git

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?

Implementing OpenStack brings three key benefits. First, it fosters flexibility by enabling seamless management of diverse cloud resources. Second, it promotes cost efficiency through resource optimization and reduced vendor lock-in. Lastly, OpenStack enhances scalability, empowering businesses to effortlessly scale their infrastructure as needed. In a nutshell, it's about versatility, savings, and growth – a triple win for businesses embracing OpenStack.

Conclusions:

In this activity, I was able to encounter the OpenStack, wherein the main focus of this Hands-On Activity is to install the OpenStack. After performing this activity, I realized that Organized steps for NTP, OpenStack packages, SQL Database, Message

Queue, Memcached, and Etcd ensure a systematic deployment. Grouping them in the inventory file adds clarity, paving the way for efficient and customized OpenStack implementations across diverse server roles. Furthermore, delving into cloud services offered valuable insights. We weighed the pros and cons, considering diverse deployments and service models. Crafting a streamlined workflow using Ansible for OpenStack installation proved effective, doubling as documentation. This journey underscored the dynamic landscape of cloud technology, emphasizing adaptability and informed decision-making in harnessing its potential.