





Name: Ian Carlo T. Bello	Date Performed: December 1, 2022
Course/Section: CPE31S24-CPE232	Date Submitted: December 2, 2022
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1 st sem – 3 rd year
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)	
<p>Creating new repository</p> <div data-bbox="727 1732 880 1793">  </div>	

Owner * Repository name *

 qictbello / CPE232_Act13 

Quick setup — if you've done this kind of thing before

 Set up in Desktop or ☐ HTTPS ☒ SSH `git@github.com:qictbello/CPE232_Act13.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#)

```
ubuntuhost@workstation:~$ git clone git@github.com:qictbello/CPE232_Act13.git
Cloning into 'CPE232_Act13'...
warning: You appear to have cloned an empty repository.
ubuntuhost@workstation:~$ cd CPE232_Act13
ubuntuhost@workstation:~/CPE232_Act13$
```

Creating prerequisite for creating ansible playbook

```
ubuntuhost@workstation:~/CPE232_Act13$ touch inventory
ubuntuhost@workstation:~/CPE232_Act13$ touch ansible.cfg
ubuntuhost@workstation:~/CPE232_Act13$
```

```
GNU nano 6.2 ansible.cfg
[defaults]
command_warnings=False
deprecation_warnings=False
inventory=inventory
private_key_file = ~/.ssh/ansible
```

Here we separate our groups in our inventory file

```
GNU nano 6.2
[controller]
server1
[compute]
server1
```

Creating roles for each installation. So, it can be installed separately even we added more servers

```
ubuntuhost@workstation:~/CPE232_Act13$ mkdir roles
ubuntuhost@workstation:~/CPE232_Act13$ cd roles
ubuntuhost@workstation:~/CPE232_Act13/roles$ mkdir -p {NTP,OpenStack,SQL,Message,memcached,etcd}/tasks
ubuntuhost@workstation:~/CPE232_Act13/roles$ ls
etcd  memcached  Message  NTP  OpenStack  SQL
ubuntuhost@workstation:~/CPE232_Act13/roles$
```

Next, we will create commands for each installation using our source.

NTP

```
- name: Install chrony
  apt:
    name: chrony
    state: latest
    update_cache: yes

- name: Start chrony
  systemd:
    name: chrony
    state: started

- name: Enable chrony
  systemd:
    name: chrony
    enabled: yes
```

Openstack

```
GNU nano 6.2 main.yml
- name: Install openstack
  apt:
    name:
      - nova-compute
      - python3-openstackclient
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```

SQL

```
GNU nano 6.2
- name: Install SQL
  apt:
    name:
      - mariadb-server
      - python3-pymysql
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Edit the mariadb file
  copy:
    content: |
      default-storage-engine = innodb
      innodb_file_per_table = on
      max_connections = 4096
      collation-server = utf_general_ci
      character-set-server = utf8
    dest: /etc/mysql/mariadb.conf.d/99-openstack.cnf
    mode: "0755"

- name: Enable mariadb
  systemd:
    name: mariadb
    enabled: yes

- name: Restart mariadb
  systemd:
    name: mariadb
    state: restarted
```

Message Queue

```
- name: Install Message queue
  apt:
    name: rabbitmq-server
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```

Memcached

```
- name: Install Memcached
  apt:
    name:
      - memcached
      - python3-memcache
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Restart memcached
  systemd:
    name: memcached
    state: restarted
```

Etcd

```
- name: Install etcd
  apt:
    name: etcd
    state: latest
    update_cache: yes
    when: ansible_distribution == "Ubuntu"

- name: Enable etcd
  systemd:
    name: etcd
    enabled: yes
```

Next, we will create the playbook itself to run this installation we will name it openstack.yml

```
ubuntu@workstation:~/CPE232_Act13$ nano openstack.yml
```

```
---
- hosts: all
  become: true
  pre_tasks:
    - name: install updates Ubuntu
      tags: always
      apt:
        upgrade: dist
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"

- hosts: controller
  become: true
  roles:
    - NTP
    - OpenStack
    - SQL

- hosts: compute
  become: true
  roles:
    - Message
    - memcached
    - etcd
```

After this we can try and run it in our server1 which is ubuntu

```

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

TASK [memcached : Restart memcached] *****
*****
changed: [server1]

TASK [etcd : Install etcd] *****
*****
changed: [server1]

TASK [etcd : Enable etcd] *****
*****
ok: [server1]

PLAY RECAP *****
*****
server1 : ok=17  changed=8  unreachable
=0      failed=0  skipped=0  rescued=0  ignored=0

```

First run is successful however my vms froze I will re-run it again and check if the installation is successful

```

ubuntuhost@workstation:~/CPE232_Act13$ ansible-playbook --ask-become-pass openstack.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [server1]

TASK [install updates Ubuntu] *****
ok: [server1]

PLAY [controller] *****

TASK [Gathering Facts] *****
ok: [server1]

TASK [NTP : Install chrony] *****
ok: [server1]

TASK [NTP : Start chrony] *****
ok: [server1]

TASK [NTP : Enable chrony] *****
ok: [server1]

TASK [OpenStack : Install openstack] *****
ok: [server1]

TASK [SQL : Install SQL] *****
ok: [server1]

TASK [SQL : Edit the mariadb file] *****
ok: [server1]

```

```

TASK [SQL : Enable mariadb] *****
ok: [server1]

TASK [SQL : Restart mariadb] *****
changed: [server1]

PLAY [compute] *****

TASK [Gathering Facts] *****
ok: [server1]

TASK [Message : Install Message queue] *****
ok: [server1]

TASK [memcached : Install Memcached] *****
ok: [server1]

TASK [memcached : Restart memcached] *****
changed: [server1]

TASK [etcd : Install etcd] *****
ok: [server1]

TASK [etcd : Enable etcd] *****
ok: [server1]

PLAY RECAP *****
server1 : ok=17  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

Let's check their status now

```

ubuntuhost@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 Thu 2022-12-01 22:02:14 PST; 33min ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
   Process: 1010 ExecStart=/usr/lib/systemd/scripts/chronyd-starter.sh $DAEMON
 Main PID: 1035 (chronyd)
    Tasks: 2 (limit: 2283)
   Memory: 1.1M
      CPU: 148ms
   CGroup: /system.slice/chrony.service
           └─1035 /usr/sbin/chronyd -F 1
             1075 /usr/sbin/chronyd -F 1

Dec 01 22:02:11 server1 systemd[1]: Starting chrony, an NTP client/server...
Dec 01 22:02:12 server1 chronyd[1035]: chronyd version 4.2 starting (+CMDMON +
Dec 01 22:02:14 server1 chronyd[1035]: Using right/UTC timezone to obtain leap
Dec 01 22:02:14 server1 chronyd[1035]: Loaded seccomp filter (level 1)
Dec 01 22:02:14 server1 systemd[1]: Started chrony, an NTP client/server.
Dec 01 22:02:31 server1 chronyd[1035]: Selected source 162.159.200.1 (1.ubuntu
Dec 01 22:02:31 server1 chronyd[1035]: System clock TAI offset set to 37 second
Dec 01 22:02:33 server1 chronyd[1035]: Source 91.189.94.4 replaced with 185.12
lines 1-23/23 (END)

```

```

ubuntuhost@server1:~$ systemctl status nova-compute.service
● nova-compute.service - OpenStack Compute
   Loaded: loaded (/lib/systemd/system/nova-compute.service; enabled; vendor
   Active: active (running) since Thu 2022-12-01 22:03:30 PST; 33min ago
 Main PID: 3555 (nova-compute)
    Tasks: 2 (limit: 2283)
   Memory: 33.2M
      CPU: 4.748s
   CGroup: /system.slice/nova-compute.service
           └─3555 /usr/bin/python3 /usr/bin/nova-compute --config-file=/etc/

Dec 01 22:03:30 server1 systemd[1]: Started OpenStack Compute.
Dec 01 22:05:15 server1 nova-compute[3555]: Modules with known eventlet monkey
lines 1-12/12 (END)

```

```

● mariadb.service - MariaDB 10.6.11 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-01 22:30:00 PST; 7min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 46451 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/lib/mysql (code=exited, status=0/SUCCESS)
   Process: 46452 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_MODE (code=exited, status=0/SUCCESS)
   Process: 46454 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && mv /usr/bin/galera_recovery_recovery.sh /usr/bin/galera_recovery_recovery.sh 2>/dev/null (code=exited, status=0/SUCCESS)
   Process: 46493 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_MODE (code=exited, status=0/SUCCESS)
   Process: 46495 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 46483 (mariabdd)
    Status: "Taking your SQL requests now..."
      Tasks: 7 (limit: 2283)
    Memory: 79.5M
       CPU: 486ms
    CGroup: /system.slice/mariadb.service
            └─46483 /usr/sbin/mariabdd

Dec 01 22:30:00 server1 mariabdd[46483]: 2022-12-01 22:30:00 0 [Note] InnoDB: Starting shutdown...
Dec 01 22:30:00 server1 mariabdd[46483]: 2022-12-01 22:30:00 0 [Warning] You have 16 unflushed changes not yet synced to disk. In normal situations with using buffers you should not see this warning.
Dec 01 22:30:00 server1 mariabdd[46483]: 2022-12-01 22:30:00 0 [Note] InnoDB: Shutdown: flushing datafiles (fs: [A] /var/lib/mysql)
Dec 01 22:30:00 server1 mariabdd[46483]: 2022-12-01 22:30:00 0 [Note] Server shutdown complete
Dec 01 22:30:00 server1 mariabdd[46483]: 2022-12-01 22:30:00 0 [Note] /usr/sbin/mariadbd: ready for connections
Dec 01 22:30:00 server1 mariabdd[46483]: Version: '10.6.11-MariaDB-0ubuntu0.22.04.1' for Linux on x86_64
Dec 01 22:30:00 server1 systemd[1]: Started MariaDB 10.6.11 database server.
Dec 01 22:30:00 server1 /etc/mysql/debian-start[46497]: Upgrading MySQL tables for 10.6.11-MariaDB
Dec 01 22:30:00 server1 /etc/mysql/debian-start[46531]: Checking for insecure permissions
Dec 01 22:30:01 server1 /etc/mysql/debian-start[46535]: Triggering mysqld to recheck permissions

```

```

ubuntuhost@server1:~$ systemctl status rabbitmq-server.service
● rabbitmq-server.service - RabbitMQ Messaging Server
   Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-01 22:03:30 PST; 31min ago
     Main PID: 1027 (beam.smp)
       Tasks: 21 (limit: 2283)
    Memory: 19.5M
       CPU: 17.036s
    CGroup: /system.slice/rabbitmq-server.service
            └─1027 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas ageffc
              └─1270 erl_child_setup 65536
                └─2761 inet_gethost 4
                  └─2762 inet_gethost 4

Dec 01 22:02:12 server1 systemd[1]: Starting RabbitMQ Messaging Server...
Dec 01 22:03:30 server1 systemd[1]: Started RabbitMQ Messaging Server.

```

```

ubuntuhost@server1:~$ systemctl status memcached.service
● memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-01 22:30:15 PST; 8min ago
     Docs: man:memcached(1)
   Main PID: 47531 (memcached)
      Tasks: 10 (limit: 2283)
    Memory: 5.4M
       CPU: 150ms
    CGroup: /system.slice/memcached.service
            └─47531 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1

Dec 01 22:30:15 server1 systemd[1]: Started memcached daemon.

```

```

ubuntuhost@server1:~$ 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 Thu 2022-12-01 22:02:43 PST; 36min ago
     Docs: https://etcd.io/docs
           man:etcd
   Main PID: 1021 (etcd)
      Tasks: 7 (limit: 2283)
    Memory: 8.8M
       CPU: 8.096s
    CGroup: /system.slice/etcd.service
            └─1021 /usr/bin/etcd

Dec 01 22:02:41 server1 etcd[1021]: enabled capabilities for version 3.3
Dec 01 22:02:43 server1 etcd[1021]: 8e9e05c52164694d is starting a new election
Dec 01 22:02:43 server1 etcd[1021]: 8e9e05c52164694d became candidate at term 4
Dec 01 22:02:43 server1 etcd[1021]: 8e9e05c52164694d received MsgVoteResp from 8e9e05c52164694d at term 4
Dec 01 22:02:43 server1 etcd[1021]: 8e9e05c52164694d became leader at term 4
Dec 01 22:02:43 server1 etcd[1021]: raft.node: 8e9e05c52164694d elected leader
Dec 01 22:02:43 server1 etcd[1021]: published {Name:server1 ClientURLs:[http://127.0.0.1:2379, http://127.0.0.1:2380]}
Dec 01 22:02:43 server1 etcd[1021]: ready to serve client requests
Dec 01 22:02:43 server1 systemd[1]: Started etcd - highly-available key value store.
Dec 01 22:02:43 server1 etcd[1021]: serving insecure client requests on 127.0.0.1

```


Now every service is running we will update our repository by committing and pushing it

```
ubuntuhost@workstation:~/CPE232_Act13$ git add -A
ubuntuhost@workstation:~/CPE232_Act13$ git commit -m "activity 13"
[main (root-commit) 4db88eb] activity 13
 9 files changed, 118 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 openstack.yml
 create mode 100644 roles/Message/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
 create mode 100644 roles/etcd/tasks/main.yml
 create mode 100644 roles/memcached/tasks/main.yml
ubuntuhost@workstation:~/CPE232_Act13$ git push
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (24/24), 2.34 KiB | 797.00 KiB/s, done.
Total 24 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qictbello/CPE232_Act13.git
 * [new branch]      main -> main
ubuntuhost@workstation:~/CPE232_Act13$
```

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?

OpenStack increases business agility and efficiency in core processes. OpenStack enhances the availability of resources and adds flexibility and adaptability to our workarounds. It is faster to innovate and doesn't need to commit to a single vendor. It is simpler to comply with changes.

Conclusions:

In conclusion, we created an Ansible playbook that installed OpenStack and its tools. We explored the environment of this app and its advantages. We created a workflow for how the service will be installed using Ansible. We met different errors that were easier to debug but took time since we needed heavy updates to run. We finished with working services installed in an Ubuntu environment.