

Name: Maxine Audrey D. Pulao	Date Performed: December 1, 2022
Course/Section: CPE31S2	Date Submitted: December 2, 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022-2023
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. <div data-bbox="300 1077 1364 1257" data-label="Text"> <pre>aud@rey:~/ansible\$ git clone git@github.com:MaxinePulao/Activity-13.git Cloning into 'Activity-13'... 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.</pre> </div> 2. Create a playbook that converts the steps in the following items in https://docs.openstack.org/install-guide/ 	

```
aud@rey:~/ansible/Activity-13/roles$ tree
```

```
.
├── database
│   ├── files
│   │   └── openstack.cnf
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── install.yml
│       └── main.yml
├── etcd
│   ├── files
│   │   ├── etcd.conf.yml
│   │   └── etcd.service
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── install.yml
│       └── main.yml
├── memcached
│   ├── files
│   │   └── memcached
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── install.yml
│       └── main.yml
├── message_queue
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── install.yml
│       └── main.yml
├── ntp
│   ├── files
│   │   └── chrony.conf
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── install.yml
│       └── main.yml
├── packages
│   └── tasks
│       ├── install.yml
│       └── main.yml
└── README.md
```

a. NTP

```
aud@rey:~/ansible/Activity-13/roles/ntp$ tree
```

```
.
├── files
│   └── chrony.conf
├── handlers
│   └── main.yml
└── tasks
    ├── install.yml
    └── main.yml
```

```
3 directories, 4 files
```

Files:

```
aud@rey:~/ansible/Activity-13/roles/ntp/files$ cat chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
#allow 192.168.0.0/16

# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
#keyfile /etc/chrony.keys

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking

allow 192.168.56.126

#server NTP_SERVER iburst
```

Handlers:

```
aud@rey:~/ansible/Activity-13/roles/ntp/handlers$ cat main.yml
- name: Restarting chrony
  service:
    name: chronyd.service
    state: restarted
    enabled: true
```

Tasks:

```

aud@rey:~/ansible/Activity-13/roles/ntp/tasks$ cat install.yml
- name: Installing chrony
  community.general.apt:
    name: chrony
    state: latest

- name: Starting chrony service
  service:
    name: chronyd.service
    state: started
    enabled: true

- name: Editing chrony.conf file
  copy:
    src: chrony.conf
    dest: /etc/chrony.conf
    owner: root
    group: root
    mode: 644

  notify: Restarting chrony

- block:
  - name: Verifying installation of chrony
    command: chronyd --version
    register: chrony_version

    - debug:
      msg="{{ chrony_version }}"

- block:
  - name: Verifying if chronyd is started and running in the background
    command: systemctl status chronyd
    register: chrony_service

    - debug:
      msg="{{ chrony_service }}"

```

b. OpenStack packages

```

aud@rey:~/ansible/Activity-13/roles/packages$ tree
.
├── tasks
│   ├── install.yml
│   └── main.yml
└── 1 directory, 2 files

```

```

aud@rey:~/ansible/Activity-13/roles/packages/tasks$ cat install.yml
- name: Enabling openstack repository
  community.general.apt_repository:
    repo: 'obs://Cloud:OpenStack:Stein/openSUSE_Leap_15.0'
    name: Stein
    auto_import_keys: true
    runrefresh: true

- name: Refreshing the opensuse repository
  community.general.zypper:
    name: '*'
    state: latest
    update_cache: true

- name: Installing openstackclient
  pip:
    name: python-openstackclient

- block:
  - name: Verifying installation of openstackclient
    shell: openstack --version
    register: openstack_version

  - debug:
      msg="{{ openstack_version }}"
aud@rey:~/ansible/Activity-13/roles/packages/tasks$ cat main.yml
- import_tasks: install.yml

```

c. SQL Database

```

aud@rey:~/ansible/Activity-13/roles/database$ tree
.
├── files
│   └── openstack.cnf
├── handlers
│   └── main.yml
└── tasks
    ├── install.yml
    └── main.yml

```

Files:

```

GNU nano 6.2 openstack.cnf
[mysqld]
bind-address = 192.168.56.126

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8

```

Handlers:

```

aud@rey:~/ansible/Activity-13/roles/database/handlers$ cat main.yml

- name: Restarting database service
  service:
    name: mysql
    state: restarted
    enabled: true

```

Tasks:

```
aud@rey:~/ansible/Activity-13/roles/database/tasks$ cat install.yml
```

```
- name: Installing database components
  community.general.zypper:
    name:
      - mariadb-client
      - mariadb

- name: Installing python-PyMySQL (components)
  expect:
    command: zypper install python-PyMySQL
    responses:
      (?i)Choose from above solutions by number or cancel: 2
      (?i)Overall download size: y

- name: Copying openstack.cnf file
  copy:
    src: openstack.cnf
    dest: /etc/my.cnf.d/openstack.cnf
    owner: root
    group: root
    mode: 644

  notify: Restarting database service

- name: Starting and enabling database service
  service:
    name: mysql
    state: started
    enabled: true

- name: Set the root password
  mysql_user: login_user=root login_password="{{ root_password }}" user=
sword="{{ root_password }}"

- name: Secure the root user for IPV6 localhost (:::1)
  mysql_user: login_user=root login_password="{{ root_password }}" user=
sword="{{ root_password }}" host=":::1"

- name: Secure the root user for IPV4 localhost (127.0.0.1)
  mysql_user: login_user=root login_password="{{ root_password }}" user=
sword="{{ root_password }}" host="127.0.0.1"

- name: Secure the root user for localhost domain
  mysql_user: login_user=root login_password="{{ root_password }}" user=
sword="{{ root_password }}" host="localhost"

- name: Secure the root user for server_hostname domain
  mysql_user: login_user=root login_password="{{ root_password }}" user=
sword="{{ root_password }}" host="{{ ansible_fqdn }}"
```

```

- name: Deletes anonymous server user
  mysql_user: login_user=root login_password="{{ root_password }}" user:
all=yes state=absent

- name: Removes the test database
  mysql_db: login_user=root login_password="{{ root_password }}" db=tes:
bsent

- block:
  - name: Verifying installation of mysql
    shell: mysql --version
    register: mysql_version

  - debug:
    msg="{{ mysql_version }}"

- block:
  - name: Verifying if mysql is started and running in the background
    shell: systemctl status mysql
    register: mysql_service

  - debug:
    msg="{{ mysql_service }}"
aud@rey:~/ansible/Activity-13/roles/database/tasks$ cat main.yml

- import_tasks: install.yml

```

d. Message Queue

```

aud@rey:~/ansible/Activity-13/roles/message_queue$ tree
.
├── handlers
│   └── main.yml
└── tasks
    ├── install.yml
    └── main.yml

2 directories, 3 files

```

Handlers:

```

aud@rey:~/ansible/Activity-13/roles/message_queue/handlers$ cat main.yml

- name: Configuring rabbitmq-server
  shell: |
    rabbitmqctl add_user openstack server54321
    rabbitmqctl set_permissions openstack ".*" ".*" ".*"

```

Tasks:

```

aud@rey:~/ansible/Activity-13/roles/message_queue/tasks$ cat install.yml
- name: Installing rabbitmq-server
  community.general.apt:
    name: rabbitmq-server
    state: latest

- name: Starting service
  service:
    name: rabbitmq-server.service
    state: started
    enabled: true

  notify: Configuring rabbitmq-server

- block:
  - name: Verifying rabbitmq-server installation
    command: rabbitmq-server --version
    register: rabbitmq_version

    - debug:
      msg="{{ rabbitmq_version }}"

- block:
  - name: Verifying rabbitmq-server installation
    command: sudo systemctl status rabbitmq-server
    register: rabbitmq_service

    - debug:
      msg="{{ rabbitmq_service }}"

aud@rey:~/ansible/Activity-13/roles/message_queue/tasks$ cat main.yml
- import_tasks: install.yml

```

e. Memcached

```

aud@rey:~/ansible/Activity-13/roles/memcached$ tree
.
├── files
│   └── memcached
├── handlers
│   └── main.yml
└── tasks
    ├── install.yml
    └── main.yml

3 directories, 4 files

```

Files:


```
aud@rey:~/ansible/Activity-13/roles/memcached/files$ cat memcached
# Path:      Network/WWW/Memcached
## Description: start parameters for memcached.
## Type:      string
## Default:    "-l 127.0.0.1"
## Config:     memcached
#
# start parameters for memcached.
#
# see man 1 memcached for more
#
MEMCACHED_PARAMS="-l 192.168.56.126"

## Path:      Network/WWW/Memcached
## Description: username memcached should run as
## Type:      string
## Default:    "memcached"
## Config:     memcached
#
# username memcached should run as
#
MEMCACHED_USER="memcached"

## Path:      Network/WWW/Memcached
## Description: group memcached should be run as
## Type:      string
## Default:    "memcached"
## Config:     memcached
#
```

Handlers:

```
aud@rey:~/ansible/Activity-13/roles/memcached/handlers$ cat main.yml
- name: Restarting memcached
  service:
    name: memcached
    state: restarted
    enabled: true
```

Tasks:

```

aud@rey:~/ansible/Activity-13/roles/memcached/tasks$ cat install.yml
- name: Installing memcached
  community.general.apt:
    name:
      - memcached
      - python-python-memcached
    state: latest

- name: Editing memcached.conf file
  copy:
    src: memcached
    dest: /etc/sysconfig/memcached
    owner: root
    group: root
    mode: 644

  notify: Restarting memcached

- name: Starting memcached service
  service:
    name: memcached
    state: started
    enabled: true

- block:
  - name: Verifying installation of memcached
    command: memcached --version
    register: memcached_version

    - debug:
        msg="{{ memcached_version }}"

- block:
  - name: Verifying if memcached is started and running in the background
    command: systemctl status memcached
    register: memcached_service

    - debug:
        msg="{{ memcached_service }}"

```

f. Etcd

```

aud@rey:~/ansible/Activity-13/roles/etcd$ tree
.
├── files
│   ├── etcd.conf.yml
│   └── etcd.service
├── handlers
│   └── main.yml
└── tasks
    ├── install.yml
    └── main.yml

3 directories, 5 files

```

Files:

```
[0] + stopped nano etcd.service
aud@rey:~/ansible/Activity-13/roles/etcd/files$ cat etcd.conf.yml
name: controller
data-dir: /var/lib/etcd
initial-cluster-state: 'new'
initial-cluster-token: 'etcd-cluster-01'
initial-cluster: controller=http://192.168.30.160:2380
initial-advertise-peer-urls: http://192.168.30.160:2380
advertise-client-urls: http://192.168.30.160:2379
listen-peer-urls: http://0.0.0.0:2380
listen-client-urls: http://192.168.30.160:2379
aud@rey:~/ansible/Activity-13/roles/etcd/files$ cat etcd.service
[Unit]
After=network.target
Description=etcd - highly-available key value store

[Service]
# Uncomment this on ARM64.
# Environment="ETCD_UNSUPPORTED_ARCH=arm64"
LimitNOFILE=65536
Restart=on-failure
Type=notify
ExecStart=/usr/bin/etcd --config-file /etc/etcd/etcd.conf.yml
User=etcd

[Install]
WantedBy=multi-user.target
```

Handlers:

```
aud@rey:~/ansible/Activity-13/roles/etcd/handlers$ cat main.yml
- name: Reloading systemd service files
  systemd:
    daemon_reload: yes
    ignore_errors: yes
```

Tasks:

```
aud@rey:~/ansible/Activity-13/roles/etcd/tasks$ cat install.yml
```

```
- name: Creating etcd user
  group:
    name: etcd
    system: true
    state: present

- name: Creating user for etcd
  user:
    name: etcd
    home: "/var/lib/etcd"
    shell: /bin/false
    group: etcd
    system: true

- name: Creating /etc/etcd directory
  file:
    path: /etc/etcd
    state: directory
    owner: etcd
    group: etcd

- name: Creating /var/lib/etcd directory
  file:
    path: /var/lib/etcd
    state: directory
    owner: etcd
    group: etcd

- name: Installing etcd tarball for x86_64/amd64
  shell: |
    ETCD_VER=v3.2.7
    rm -rf /tmp/etcd && mkdir -p /tmp/etcd
    curl -L https://github.com/coreos/etcd/releases/download/${ETCD_VER}/etcd-
ETCD_VER}-linux-amd64.tar.gz -o /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz
    tar xzvf /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz -C /tmp/etcd --strip-com
nents=1
    cp /tmp/etcd/etcd /usr/bin/etcd
    cp /tmp/etcd/etcdctl /usr/bin/etcdctl

- name: Creating a config file for etcd
  copy:
    src: etcd.conf.yml
    dest: /etc/etcd/etcd.conf.yml
    owner: root
    group: root
    mode: 644
```

```

notify: Reloading systemd service files

- name: Starting and enabling service of etcd
  service:
    name: etcd
    state: started
    enabled: true

- block:
  - name: Verifying installation of etcd
    command: etcd --version
    register: etcd_version

  - debug:
    msg="{{ etcd_version }}"

- block:
  - name: Verifying if etcd is started and running in the background
    command: systemctl status etcd
    register: etcd_service

  - debug:
    msg="{{ etcd_service }}"

```

```

aud@rey:~/ansible/Activity-13/roles/etcd/tasks$ cat main.yml
- import_tasks: install.yml

```

- g. Create different plays in installing per server type (controller, compute etc.) and identify it as a group in Inventory file.

```

aud@rey:~/ansible/Activity-13$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = False
deprecation_warnings = False
private_key_file = ~/.ssh/id_rsa

```

```

aud@rey:~/ansible/Activity-13$ cat inventory
[controller_node]
192.168.56.126 ansible_user=ubuntu

```

```
aud@grey:~/ansible/Activity-13$ cat install_openstack.yml
---

- hosts: all
  become: true
  pre_tasks:

    - name: Updating and upgrading the operating system
      community.general.zypper:
        name: "*"
        state: latest
        update_cache: true

- hosts: controller_node
  become: true
  roles:
    - ntp
    - packages
    - database
    - message_queue
    - memcached
    - etcd
```

5. Output (screenshots and explanations)

```

aud@rey:~/ansible/Hands_on_Activity_13.1$ ansible-playbook --ask-become-pass
nstack.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.120]

TASK [Dpkg fixing in Ubuntu Servers] *****
changed: [192.168.56.120]

TASK [Update and Upgrade remote in Ubuntu servers] *****
[WARNING]: The value "True" (type bool) was converted to "'True'" (type
string). If this does not look like what you expect, quote the entire value
ensure it does not change.
ok: [192.168.56.120]

PLAY [controller_node] *****

TASK [Gathering Facts] *****
ok: [192.168.56.120]

TASK [ntp : Installing the NTP] *****
ok: [192.168.56.120]

TASK [ntp : Enable the chrony] *****
changed: [192.168.56.120]

TASK [os-p : Installing the OpenStack-Packages] *****
ok: [192.168.56.120]

TASK [sql-d : Installing the SQL-Database] *****
changed: [192.168.56.120]

TASK [sql-d : editing the maria-db.conf file] *****
changed: [192.168.56.120]

TASK [sql-d : Restarting the mariadb-server] *****
changed: [192.168.56.120]

TASK [m-q : Installing Message-Queue] *****

PLAY RECAP *****
192.168.56.121      : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.122      : ok=5    changed=0    unreachable=0    failed=0
PLAY RECAP *****
192.168.56.121      : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.122      : ok=5    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.128      : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0

```

Reflections:

Answer the following:

1. What are the benefits of implementing OpenStack?

By offering a platform with on-demand, resource pooling, self-service, highly elastic, and measurable services characteristics, OpenStack increases corporate agility, availability, and efficiency. That's because cloud computing is what this sounds like. Your primary tool for creating your own cloud infrastructure is OpenStack. Developers and IT personnel may use IT resources more quickly and effectively thanks to OpenStack's orchestration and self-service capabilities. Developers can drastically shorten development and testing times and have greater freedom to try out novel ideas because they can quickly and on-demand provision computers. End users and business units won't have to wait days or weeks to start using the network services and apps they require thanks to faster IT resource deployment. They would then be better equipped to launch and complete projects earlier than before. OpenStack can aid in efforts to maintain regulatory compliance since it makes it possible to build private, on-premise clouds. You will have more control over access privileges, security protocols, and security rules if your cloud is located in your own data center. You can personally be in charge of making sure that regulations for protecting personal information, financial information, and other sensitive information are truly followed and not just written down on a piece of paper.

Reference: <https://www.rutter-net.com/blog/the-benefits-and-challenges-of-using-openstack-in-your-business>

Conclusions:

In this activity, I have successfully analyzed the advantages and disadvantages of cloud services. I also evaluated different Cloud deployment and service models. With this activity, it taught me how to effectively create a workflow to install and configure OpenStack base services using Ansible as documentation and execution despite being hard. I have learned to troubleshoot and determine how to come up with solutions to new problems. With that, I have sharpened my engineering skills which may be useful in my future endeavors as a computer engineer.