Name: Juanson, Aliya Dane P.	Date Performed: November 13, 2024
Course/Section: CPE31S2	Date Submitted: November 13, 2024
Instructor: Sir Robin Valenzuela	Semester and SY: 2024-2025

Activity 11: Containerization

1. Objectives

Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process

2. Discussion

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Source: https://docs.docker.com/get-started/overview/

You may also check the difference between containers and virtual machines. Click the link given below.

Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

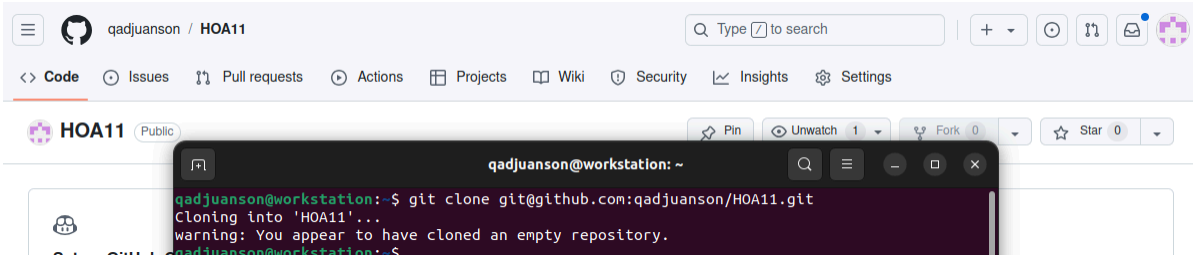
3. Tasks

1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

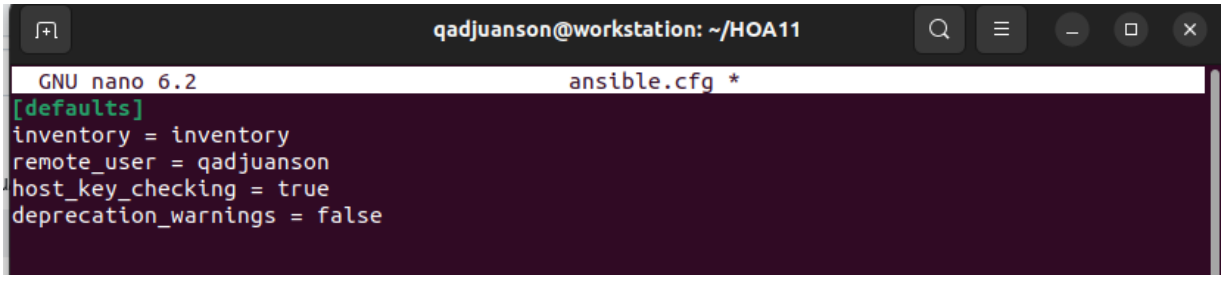
```
qadjuanson@workstation:~/HOA11$ git add .
qadjuanson@workstation:~/HOA11$ git commit -m "DONE"
[main (root-commit) 638653c] DONE
 6 files changed, 184 insertions(+)
 create mode 100644 Docker/Dockerfile
 create mode 100644 ansible.cfg
 create mode 100644 docker.yml
 create mode 100644 inventory
 create mode 100644 roles/ubuntu/tasks/main.yml
 create mode 100644 test.yml
qadjuanson@workstation:~/HOA11$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 2.19 KiB | 2.19 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qadjuanson/HOA11.git
 * [new branch]      main -> main
```

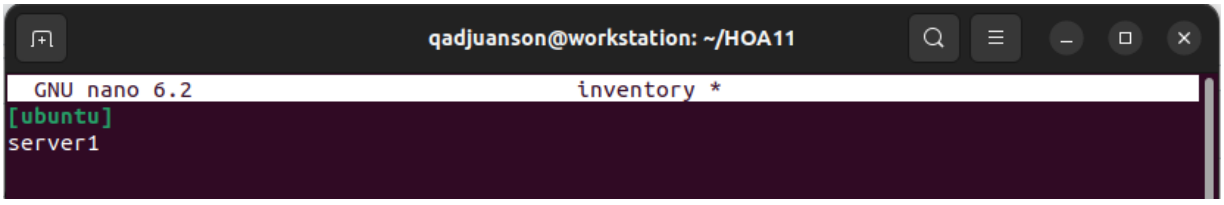
https://github.com/qadjuanson/HOA11

4. **Output** (screenshots and explanations).



- After creating a repository, I clone the repository in my control node.





- Create an *ansible. cfg* and *inventory file.*

```
qadjuanson@workstation:~/HOA11$ mkdir Docker
qadjuanson@workstation:~/HOA11$ cd Docker
qadjuanson@workstation:~/HOA11/Docker$ sudo nano Dockerfile
```

- Create a directory named *Docker* and inside of it create a *Dockerfile.*

```
qadjuanson@workstation: ~/HOA11/Docker
  GNU nano 6.2                                          Dockerfile *
FROM ubuntu:latest
MAINTAINER cpmiranda <qadjuanson@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get update -y
RUN apt-get upgrade -y

RUN apt-get install apache2 -y
RUN apt-get install php libapache2-mod-php -y
RUN apt-get install mariadb-server mariadb-client -y

RUN /etc/init.d/apache2 start

ENTRYPOINT apache2ctl -D FOREGROUND
```

- This is the content of *Dockerfile.*

```
qadjuanson@workstation:~/HOA11$ cd roles
qadjuanson@workstation:~/HOA11/roles$ mkdir ubuntu
qadjuanson@workstation:~/HOA11/roles$ cd ubuntu
qadjuanson@workstation:~/HOA11/roles/ubuntu$ mkdir tasks
qadjuanson@workstation:~/HOA11/roles/ubuntu$ cd tasks
```

- Create a directory named *roles* and inside of it create another directory named *ubuntu*. Inside the directory *ubuntu*, create again a directory named *tasks*.

qadjuanson@workstation: ~/HOA11/roles/ubuntu/tasks

GNU nano 6.2 main.yml *

```
---
- name: Start the Docker Service in Ubuntu
  tags: prep
  become: true
  service:
    name: docker
    state: started
    enabled: true

- name: Ensure group docker exists
  tags: prep
  become: true
  group:
    name: docker
    state: present

- name: Adding the current user to the docker group
  tags: prep
  user:
    name: "{{ ansible_user }}"
    groups: docker
    append: yes

- name: Create a docker directory
  file:
    path: /home/qadjuanson/docker_files
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/qadjuanson/HOA11/Docker/Dockerfile
```

```
    dest: /home/qadjuanson/docker_files
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'
- name: Build Docker Image
  become: true
  docker_image:
    path: /home/qadjuanson/docker_files/
    name: docker-image-apache-mariadb
    tag: latest
    state: present
```

- Create a playbook inside *~/HOA11/roles/ubuntu/tasks.*

```
qadjuanson@workstation:~/HOA11$ sudo nano docker.yml
```

```
qadjuanson@workstation: ~/HOA11
  GNU nano 6.2                                                  docker.yml *
---
 - hosts: all
   become: true
   pre_tasks:

   - name: Install Updates (Ubuntu)
     tags: always
     apt:
       update_cache: yes
     changed_when: false
     when: ansible_distribution == "Ubuntu"


 - hosts: all
   become: true
   roles:
     - ubuntu
```

- Create a docker.yml to run the playbook.

```
qadjuanson@workstation:~/HOA11$ sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 2 not upgraded.
Need to get 75.2 MB of archives.
After this operation, 283 MB of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.12-0ubuntu2~22.04.1 [8,405 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.12-0ubuntu2~22.04.1 [37.8 MB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu2~22.04.1 [28.8 MB]
Get:6 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 75.2 MB in 9s (8,418 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 280339 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12-0ubuntu2~22.04.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu2~22.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.12-0ubuntu2~22.04.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu2~22.04.1) ...
```

```
qadjuanson@workstation:~/HOA11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-11-13 22:05:38 +08; 1min 0s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 5730 (dockerd)
      Tasks: 8
     Memory: 61.1M
        CPU: 270ms
     CGroup: /system.slice/docker.service
             └─5730 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 13 22:05:37 workstation systemd[1]: Starting Docker Application Container Engine...
Nov 13 22:05:37 workstation dockerd[5730]: time="2024-11-13T22:05:37.671338356+08:00" level=info msg="Starting up"
Nov 13 22:05:37 workstation dockerd[5730]: time="2024-11-13T22:05:37.675218243+08:00" level=info msg="detected 127.0.0.53 nameserve>
Nov 13 22:05:37 workstation dockerd[5730]: time="2024-11-13T22:05:37.814016573+08:00" level=info msg="Loading containers: start."
Nov 13 22:05:38 workstation dockerd[5730]: time="2024-11-13T22:05:38.466887172+08:00" level=info msg="Loading containers: done."
Nov 13 22:05:38 workstation dockerd[5730]: time="2024-11-13T22:05:38.581609436+08:00" level=info msg="Docker daemon" commit="24.0.7>
Nov 13 22:05:38 workstation dockerd[5730]: time="2024-11-13T22:05:38.584637182+08:00" level=info msg="Daemon has completed initiali>
Nov 13 22:05:38 workstation dockerd[5730]: time="2024-11-13T22:05:38.685705202+08:00" level=info msg="API listen on /run/docker.soc>
Nov 13 22:05:38 workstation systemd[1]: Started Docker Application Container Engine.
lines 1-21/21 (END)
```

```
qadjuanson@workstation:~/HOA11$ sudo systemctl enable docker.socket
qadjuanson@workstation:~/HOA11$ sudo systemctl start docker.socket
qadjuanson@workstation:~/HOA11$ sudo systemctl status docker.socket
● docker.socket - Docker Socket for the API
     Loaded: loaded (/lib/systemd/system/docker.socket; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-11-13 22:05:37 +08; 2min 18s ago
   Triggers: ● docker.service
     Listen: /run/docker.sock (Stream)
      Tasks: 0 (limit: 1063)
     Memory: 0B
        CPU: 594us
     CGroup: /system.slice/docker.socket

Nov 13 22:05:37 workstation systemd[1]: Starting Docker Socket for the API...
Nov 13 22:05:37 workstation systemd[1]: Listening on Docker Socket for the API.
qadjuanson@workstation:~/HOA11$
```

- Install the docker.io and enable the docker.socket

```
qadjuanson@workstation:~/HOA11$ ansible-playbook docker.yml --ask-become-pass
BECOME password:

PLAY [all] ********************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [192.168.56.102]

TASK [Install Updates (Ubuntu)] ***********************************************************************************************
ok: [192.168.56.102]

PLAY [all] ********************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [192.168.56.102]

TASK [ubuntu : Start the Docker Service in Ubuntu] ****************************************************************************
changed: [192.168.56.102]

TASK [ubuntu : Ensure group docker exists] *************************************************************************************
ok: [192.168.56.102]

TASK [ubuntu : Adding the current user to the docker group] ********************************************************************
ok: [192.168.56.102]

TASK [ubuntu : Create a docker directory] **************************************************************************************
ok: [192.168.56.102]

TASK [ubuntu : Copy Dockerfile to Ubuntu] **************************************************************************************
ok: [192.168.56.102]

TASK [ubuntu : Build Docker Image] ********************************************************************************************
changed: [192.168.56.102]

PLAY RECAP ********************************************************************************************************************
192.168.56.102             : ok=9    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

qadjuanson@workstation:~/HOA11$
```

- Run the playbook *docker.yml.*

```
qadjuanson@server1:~$ sudo docker images
REPOSITORY                   TAG       IMAGE ID       CREATED         SIZE
docker-image-apache-mariadb  latest    7ec50e844af7   4 minutes ago   580MB
ubuntu                       latest    59ab366372d5   4 weeks ago     78.1MB
qadjuanson@server1:~$
```

- Go to server1 and verify if the docker images are working.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   - Containerization makes deploying and managing applications much easier by packaging them with all their necessary files and dependencies. This way, the application will run the same way across different environments, such as a developer's computer, testing systems, or production servers, without requiring adjustments. Containers are lightweight and share the host system's operating system, which makes them faster and more efficient than virtual machines. This means we can run more applications on the same hardware and scale them up or down quickly to meet demand. Containers are isolated, so issues in one container don't impact others, which also improves security. They're ideal for building microservices, where different parts of an application can be developed, tested, and deployed separately, making updates easier. In short, containerization helps developers work faster and more consistently, and it supports modern development practices like continuous integration and deployment, where frequent updates and changes are essential.

Conclusions:

- In this activity, I am able to install docker and create a docker file using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process.This activity is important for us because it will be our foundation in our final project. This will help us in the future making of our final project.