Name: Baltazar, Paul Eimar R.	**Date Performed:** November 13, 2024
Course/Section: CPE31S2	**Date Submitted:** Nov 13, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem 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/

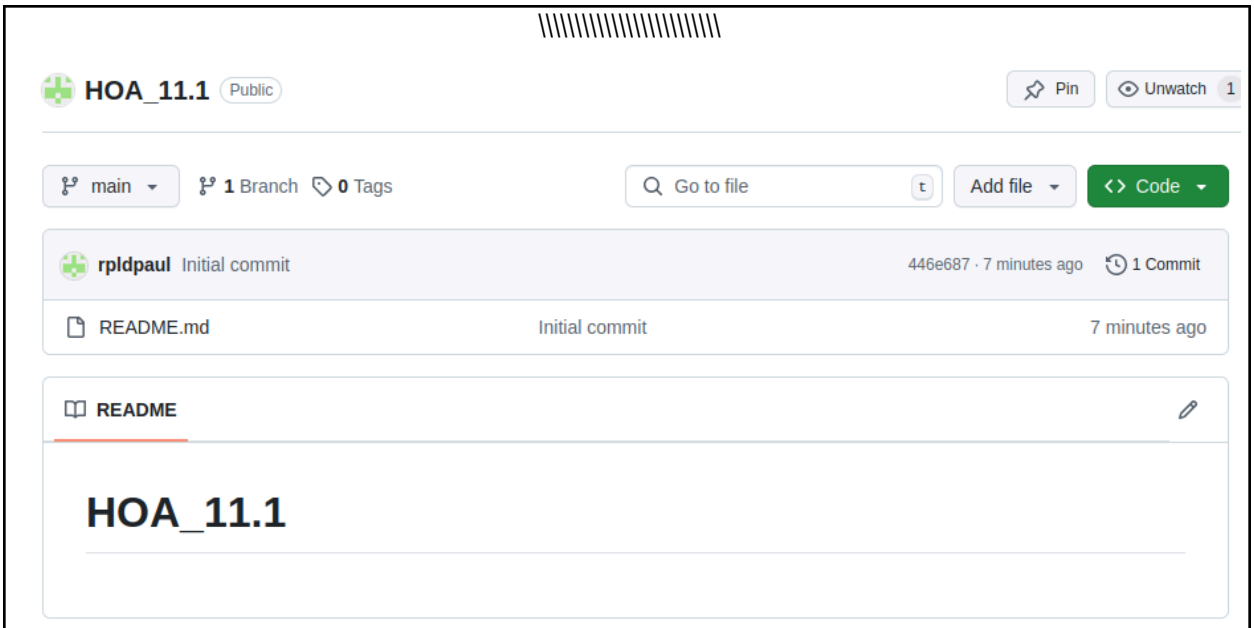
You may also check the difference between containers and virtual machines. Click the link given below.

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.

4. Output (screenshots and explanations)



Creation of repository for the hands-on activity.

```yaml
---
- name: Install Docker
  tags: prep
  become: true
  apt:
    name:
      - docker.io
    state: latest

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

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

TASK [Ubuntu : Install Docker] ****************************
ok: [192.168.56.102]

TASK [Ubuntu : Start the Docker Service in Ubuntu] ********
ok: [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]
```

```
qperbaltazar@server1:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Wed 2024-11-13 20:00:49 +08; 1h 31min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 5192 (dockerd)
      Tasks: 8
     Memory: 30.5M
        CPU: 9.063s
     CGroup: /system.slice/docker.service
             └─5192 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>
```

Installation, enabling, and adding user to docker group to the remote Ubuntu server through ansible playbook.

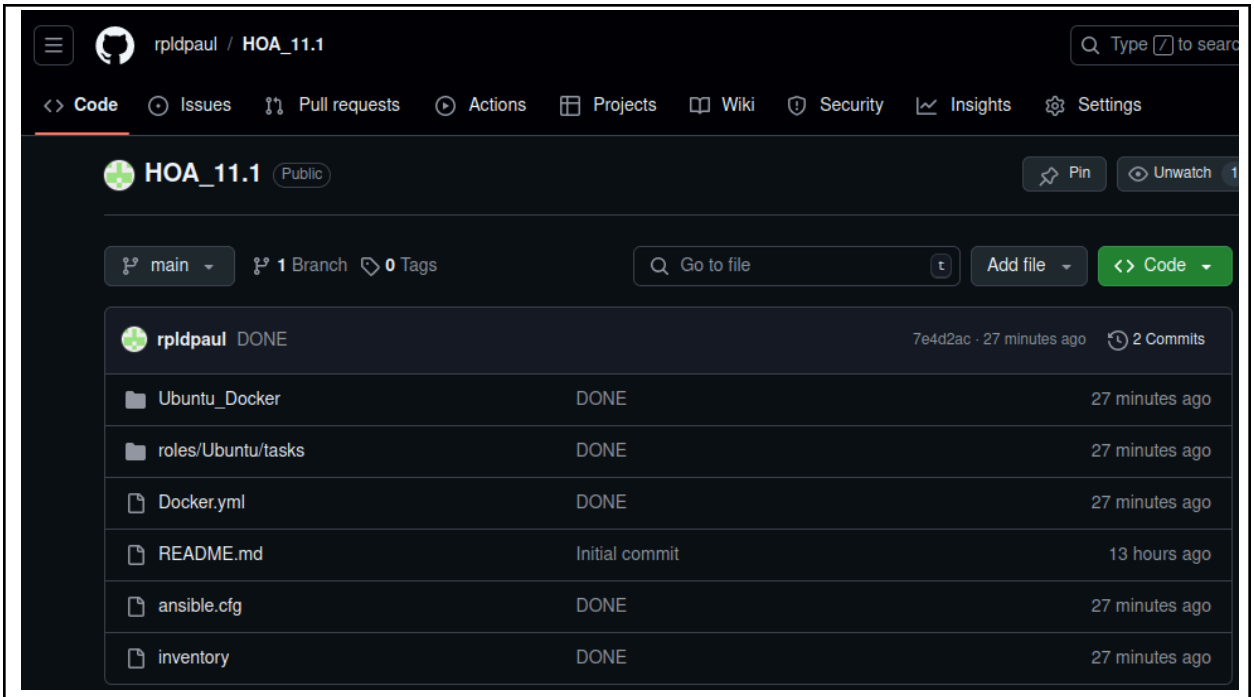
```yaml
- name: Create a docker directory
  file:
    path: /home/qperbaltazar/docker_files
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/qperbaltazar/HOA_11.1/Ubuntu_Docker/Dockerfile
    dest: /home/qperbaltazar/docker_files/
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Build Docker Image
  become: true
  docker_image:
    path: /home/qperbaltazar/docker_files/
    name: apache-mariadb-image
    tag: latest
    state: present
  vars:
    ansible_python_interpreter: /usr/bin/python3
```

Building of docker image

```
qperbaltazar@server1:~$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
apache-mariadb-image    latest    13c9b0f4c1e1   43 minutes ago   580MB
ubuntu                  latest    59ab366372d5   4 weeks ago      78.1MB
```



activity is pushed on the repository.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   Containerization has several advantages that make it useful in software development. It allows applications to run consistently across different environments by creating isolated, lightweight units called containers. Unlike virtual machines, containers share the host's operating system, making them faster to start and less resource-intensive. This efficiency lets multiple containers run smoothly on the same server, saving costs and resources. Containers also make it easy to scale applications up or down, as needed, by quickly adding or removing containers. Additionally, they allow for a modular design, where different parts of an application can be managed separately, making updates and maintenance simpler and more secure.

Conclusions:

In this activity, we installed docker and also created a docker image through ansible. This gave me hands-on experience on how to work with ansible and docker and also what docker does.