Name: Ilagan, Carlo Hideki D.	Date Performed: 11/13/2024
Course/Section: CPE31S2	Date Submitted: 11/13/2024
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.

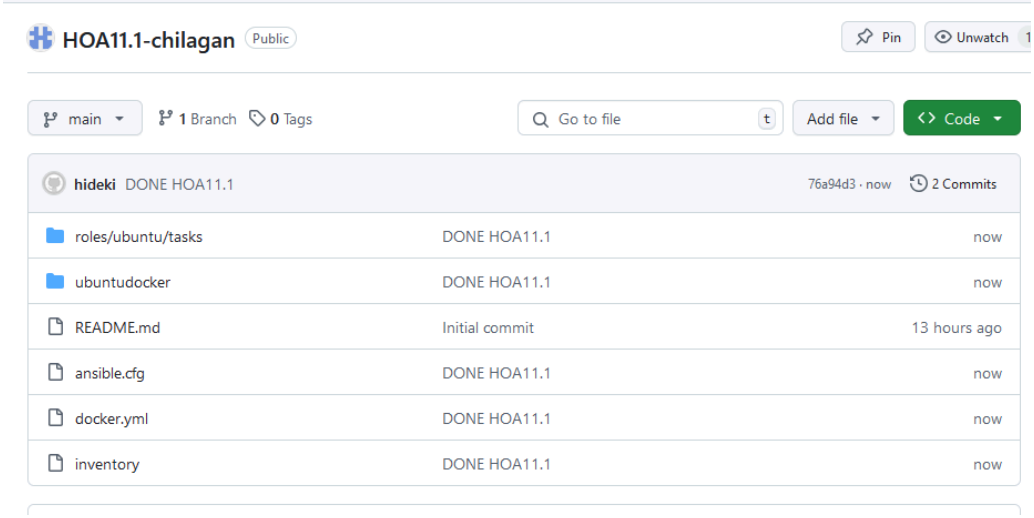
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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. (Automate)
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

Repository:



Install Docker:

```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: started
    enabled: true

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

Add to Docker group:

```
- 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/hideki/docker_files
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/hideki/HOA11.1-chilagan/Ubuntu_Docker/Dockerfile
    dest: /home/hideki/docker_files/
    owner: "{{ ansible_user }}"
```

Install and build the Dockerfile using Ansible.

Full Ansible Playbook:

```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: started
    enabled: true

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

```yaml
- 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/hideki/docker_files
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/hideki/HOA11.1-chilagan/ubuntudocker/Dockerfile
    dest: /home/hideki/docker_files/
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'
```

```
- name: Build Docker image
  become: true
  community.docker.docker_image:
    name: apache-mariadb-image
    tag: latest
    source: build
    build:
      path: /home/{{ ansible_user }}/docker_files/
      dockerfile: Dockerfile
```

Create a Dockerfile to install web and DB server:
Dockerfile:

```
FROM ubuntu:latest
MAINTAINER hideki <qchdilagan@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
```

Process:

```
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]

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] **********************************************
```

```
TASK [ubuntu : Build Docker image] **********************************************
changed: [192.168.56.102]

PLAY RECAP *********************************************************************
192.168.56.102             : ok=10   changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Output from the server:

```
hideki@server1:~$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
apache-mariadb-image    latest    f1b67c81d83b   2 minutes ago   580MB
ubuntu                  latest    59ab366372d5   4 weeks ago     78.1MB
```

For this first final term activity, we were tasked to install docker into our virtual machines, even the control node. It is the first task to do in order to proceed to the activity. As usual, inventory, and ansible cfg files were created to run the playbook. For the dockerfile, there were new commands that were used, it was my first time using these types of query in this course. It was fun since I was able to learn new things that will be helpful for our final project. Lastly, a main yml file was created to provide the following needs such as adding the docker to group, and building the docker image file.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

    - One of the most obvious benefits of implementing containerization is its portability, it is without a doubt will always be mentioned. Building a container will make your file or application be anywhere without having to do it all over again for a new or different environment. It is also an efficient way of virtualization to developers since available resources are always ready and they minimize overhead. Lastly, an improved security in which containerization provides an additional layer of security. Containers are isolated from one another, it means that even if the security of one container is compromised, the others will not be affected and will remain secured.

Conclusions:

In conclusion, I was able to install docker in my control node and 1 ubuntu managed node. I had an error in building the image but managed to fix the said problem. On the server 1, I was able to install the apache mariadb server which is the main task in this activity. Doing this activity gave me a much clear idea on how the project for this course will be done.