Name: Kenn Cherwin C. Yu	**Date Performed:** 16/04/2024
Course/Section: CPE31S1	**Date Submitted:** 16/04/2024
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 2nd Sem | 2024

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.

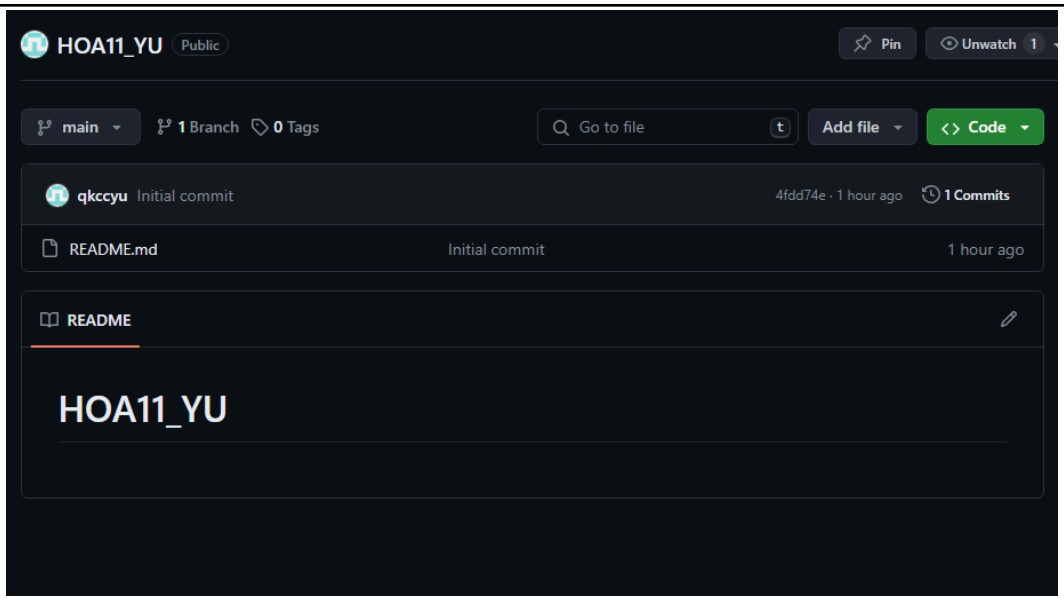
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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.

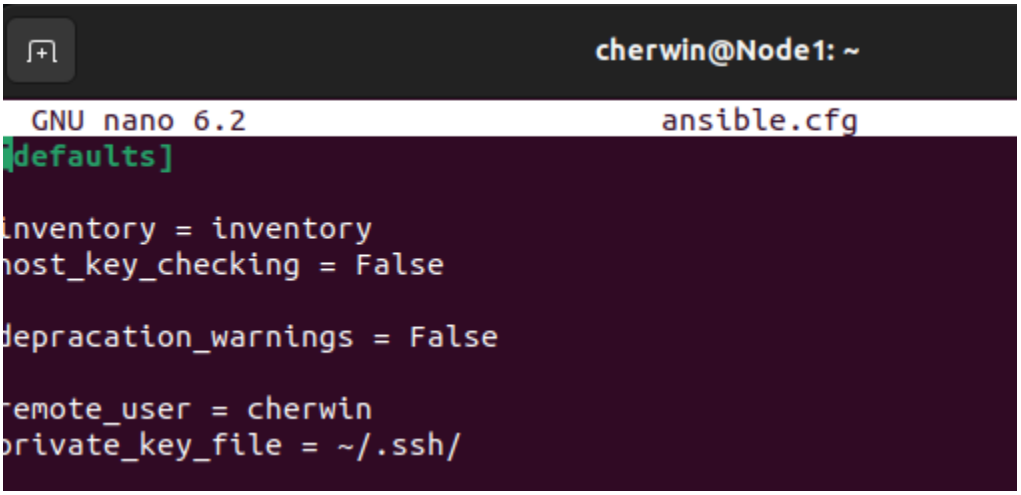
4. Output (screenshots and explanations)

Repository

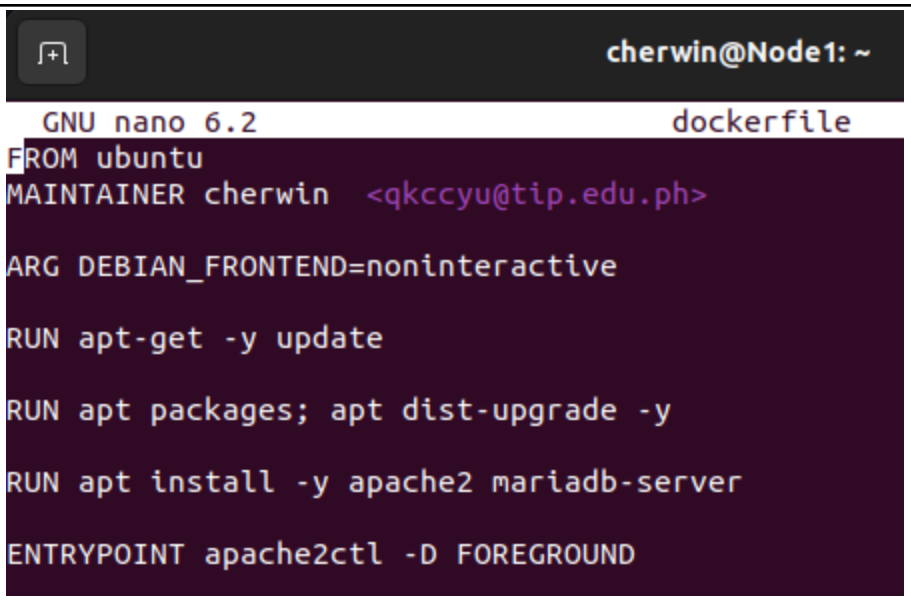



```
cherwin@Node1:~$ git clone https://github.com/qkccyu/HOA11_YU.git
Cloning into 'HOA11_YU'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 21 (delta 6), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), 6.31 KiB | 6.31 MiB/s, done.
Resolving deltas: 100% (6/6), done.
cherwin@Node1:~$
```

Ansible.cfg



Dockerfile



dockerfile.yml

cherwin@Node1: ~

GNU nano 6.2 dockerfile.yml *

```
- hosts: web_servers
  become: true
  pre_tasks:

    - name: install docker
      shell:
        sudo apt-get install docker.io -y
      when: ansible_distribution == "Ubuntu"

    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: Install Docker (Ubuntu)
      apt:
        name: docker
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install SDK (Ubuntu)
      shell:
        pip3 install docker-py

    - name: Adding group to Docker
      shell:
        usermod -aG docker cherwin

    - name: Enable/Restart Docker (Ubuntu)
      service:
        name: docker
        state: started
        enabled: true

    - name: Creating Directory for Dockerfile
      file:
```

Server

cherwin@Node1: ~/HOA11_YU

GNU nano 6.2 inventory

```
[web_servers]
192.168.56.8

[db_servers]
192.168.122.1
```

Tree

```
cherwin@Node1:~/HOA11_YU$ tree
.
├── ansible.cfg
├── dockerfile
├── dockerfile.yml
├── inventory
└── README.md

0 directories, 5 files
cherwin@Node1:~/HOA11_YU$
```

```
   Active: active (running) since Thu   2024-16-04 05:07:09 EST; 7min ago
     Docs: https://docs.docker.com
 Main PID: 8702 (dockerd)
    Tasks: 9
   Memory: 105.6M
   CGroup: /system.slice/docker.service
           └─8702 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd....
```

REPOSITORY: https://github.com/qkccyu/HOA11_YU

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Conclusions:

- This module taught us how to use containers alongside Dockers. We installed and pre-configured Docker in two different operating systems, mainly Ubuntu Desktop and CentOS. The skills and knowledge that we learned in this module will be beneficial in the future as we grow system administrators.