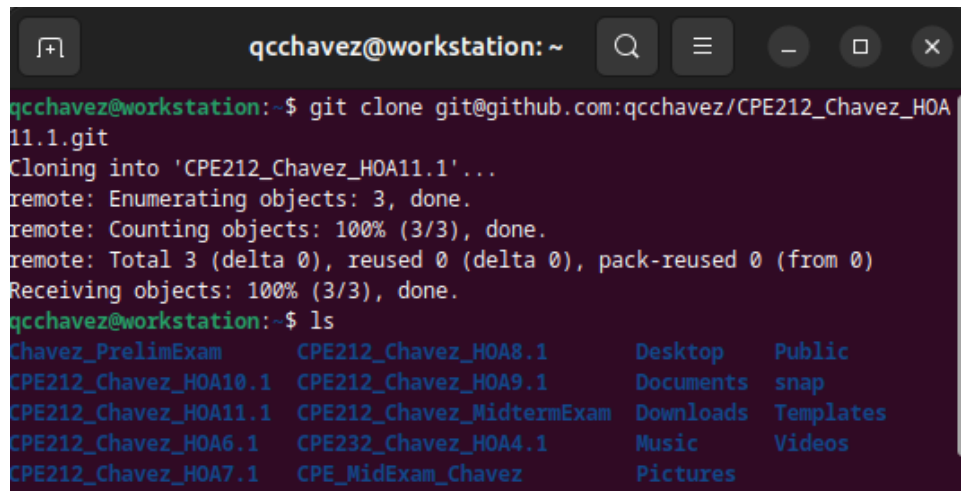


Name: Clarence B. Chavez	Date Performed: November 13, 2024
Course/Section: CPE31S2	Date Submitted: November 13, 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem, 2024-2025
Activity 11: Containerization	
1. Objectives	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process	
2. Discussion	
<p>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.</p> <p>Source: https://docs.docker.com/get-started/overview/</p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</p>	
3. Tasks	
<ol style="list-style-type: none"> 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)

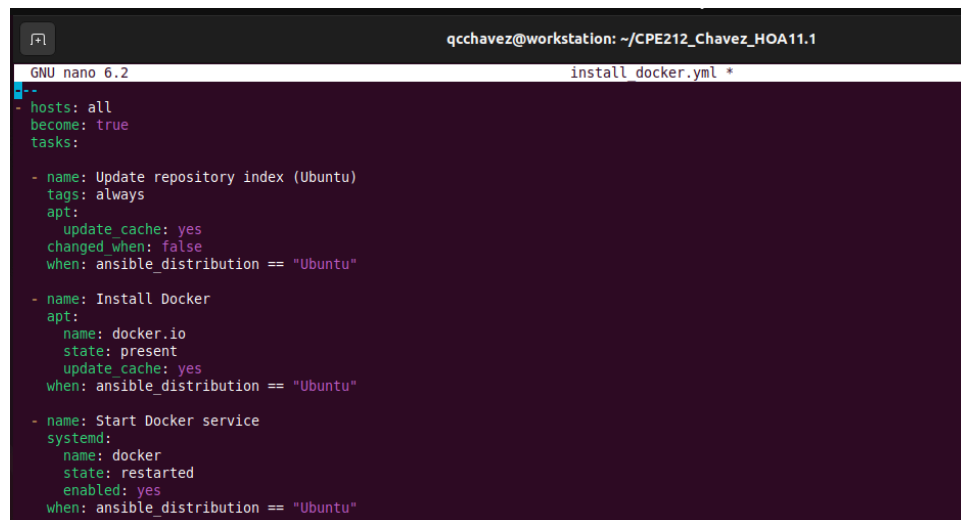
Task 3.1



```
qcchavez@workstation: ~  
qcchavez@workstation:~$ git clone git@github.com:qcchavez/CPE212_Chavez_HOA11.1.git  
Cloning into 'CPE212_Chavez_HOA11.1'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (3/3), done.  
qcchavez@workstation:~$ ls  
Chavez_PrelimExam    CPE212_Chavez_HOA8.1    Desktop    Public  
CPE212_Chavez_HOA10.1 CPE212_Chavez_HOA9.1    Documents  snap  
CPE212_Chavez_HOA11.1 CPE212_Chavez_MidtermExam Downloads  Templates  
CPE212_Chavez_HOA6.1  CPE232_Chavez_HOA4.1    Music      Videos  
CPE212_Chavez_HOA7.1  CPE_MidExam_Chavez      Pictures
```

- In this screenshot, this is the creation of the GitHub repository for this activity.

Task 3.2



```
qcchavez@workstation: ~/CPE212_Chavez_HOA11.1  
GNU nano 6.2                                install docker.yml *  
- hosts: all  
  become: true  
  tasks:  
  
  - name: Update repository index (Ubuntu)  
    tags: always  
    apt:  
      update_cache: yes  
      changed_when: false  
      when: ansible_distribution == "Ubuntu"  
  
  - name: Install Docker  
    apt:  
      name: docker.io  
      state: present  
      update_cache: yes  
      when: ansible_distribution == "Ubuntu"  
  
  - name: Start Docker service  
    systemd:  
      name: docker  
      state: restarted  
      enabled: yes  
      when: ansible_distribution == "Ubuntu"
```

- In this screenshot, this is the installation of Docker and at the same time, enabling it.

```
qcchavez@server1:~$ sudo docker --version
[sudo] password for qcchavez:
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
qcchavez@server1:~$
```

- In this screenshot, this confirms that docker has been installed in the remote server.

Task 3.3

```
- name: Add user "qcchavez" to Docker group
  user:
    name: "{{ ansible_user }}"
    groups: docker
    append: yes
    state: present
  when: ansible_distribution == "Ubuntu"
```

- In this screenshot, this is where the current user **qcchavez** is added to the Docker group.

```
qcchavez@server1:~$ getent group docker
docker:x:138:qcchavez
qcchavez@server1:~$
```

- In this screenshot, it is confirmed that user **qcchavez** is indeed added to the Docker group successfully.

Task 3.4

```
qcchavez@workstation: ~/CPE212_Chavez_HOA11.1
GNU nano 6.2 Dockerfile
FROM ubuntu:20.04

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y \
    apache2 \
    mysql-server \
    php \
    libapache2-mod-php \
    php-mysql \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

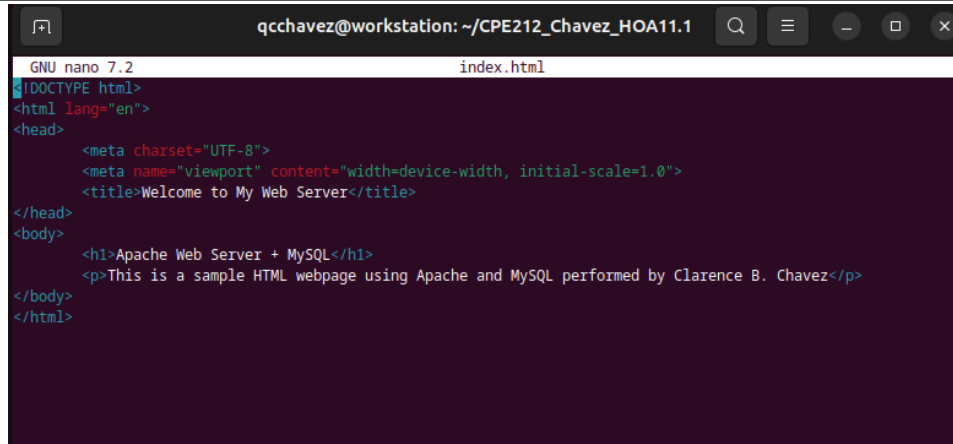
ENV MYSQL_ROOT_PASSWORD=rootpassword
ENV MYSQL_DATABASE=exampledb

COPY ./index.html /var/www/html/

EXPOSE 80 3306

CMD service mysql start && apache2ctl -D FOREGROUND
```

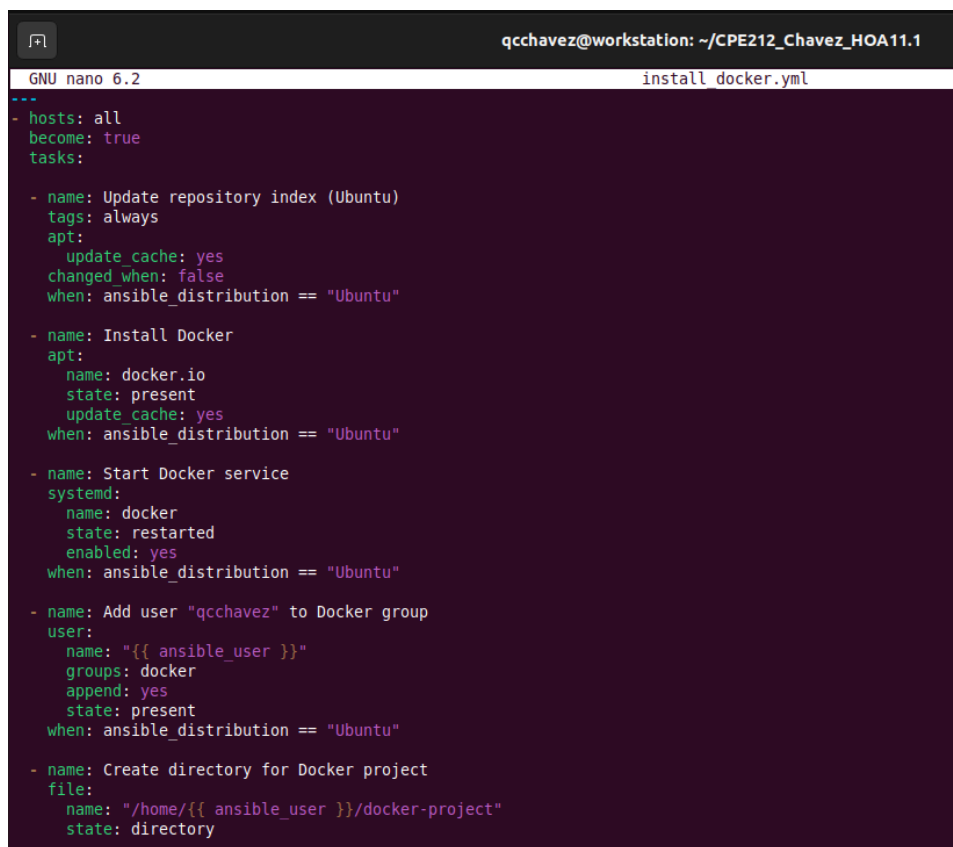
- In this screenshot, this is the codes for the creation of the Dockerfile



```
qccchavez@workstation: ~/CPE212_Chavez_HOA11.1
GNU nano 7.2 index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome to My Web Server</title>
</head>
<body>
  <h1>Apache Web Server + MySQL</h1>
  <p>This is a sample HTML webpage using Apache and MySQL performed by Clarence B. Chavez</p>
</body>
</html>
```

- In this screenshot, this is the file content of my **index.html**

Task 3.5



```
qccchavez@workstation: ~/CPE212_Chavez_HOA11.1
GNU nano 6.2 install_docker.yml
---
- hosts: all
  become: true
  tasks:

  - name: Update repository index (Ubuntu)
    tags: always
    apt:
      update_cache: yes
      changed_when: false
      when: ansible_distribution == "Ubuntu"

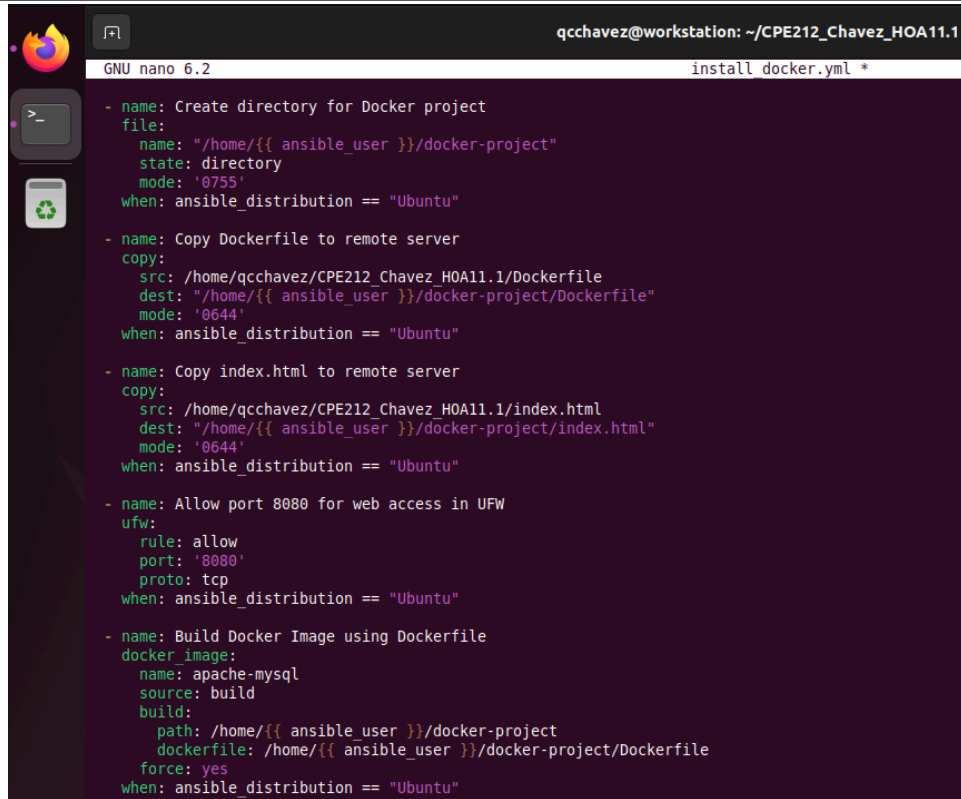
  - name: Install Docker
    apt:
      name: docker.io
      state: present
      update_cache: yes
      when: ansible_distribution == "Ubuntu"

  - name: Start Docker service
    systemd:
      name: docker
      state: restarted
      enabled: yes
      when: ansible_distribution == "Ubuntu"

  - name: Add user "qccchavez" to Docker group
    user:
      name: "{{ ansible_user }}"
      groups: docker
      append: yes
      state: present
      when: ansible_distribution == "Ubuntu"

  - name: Create directory for Docker project
    file:
      name: "/home/{{ ansible_user }}/docker-project"
      state: directory
```

- In this screenshot, these are the code of the ansible playbook where it installs **docker**, adds the user **qccchavez** to Docker group.



```
GNU nano 6.2                                install_docker.yml *
- name: Create directory for Docker project
  file:
    name: "/home/{{ ansible_user }}/docker-project"
    state: directory
    mode: '0755'
  when: ansible_distribution == "Ubuntu"

- name: Copy Dockerfile to remote server
  copy:
    src: /home/qcchavez/CPE212_Chavez_HOA11.1/Dockerfile
    dest: "/home/{{ ansible_user }}/docker-project/Dockerfile"
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Copy index.html to remote server
  copy:
    src: /home/qcchavez/CPE212_Chavez_HOA11.1/index.html
    dest: "/home/{{ ansible_user }}/docker-project/index.html"
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Allow port 8080 for web access in UFW
  ufw:
    rule: allow
    port: '8080'
    proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Build Docker Image using Dockerfile
  docker_image:
    name: apache-mysql
    source: build
    build:
      path: /home/{{ ansible_user }}/docker-project
      dockerfile: /home/{{ ansible_user }}/docker-project/Dockerfile
    force: yes
  when: ansible_distribution == "Ubuntu"
```

- In this screenshot, these are the code of the ansible playbook where it creates a directory for **docker** to the remote server, copies the **Dockerfile** and **index.html** to the remote server. The playbook also requires port 8080 in the firewall in order to access the **index.html**.

```
qccchavez@workstation: ~/CPE212_Chavez_HOA11.1
GNU nano 6.2 install_docker.yml *

when: ansible_distribution == "Ubuntu"

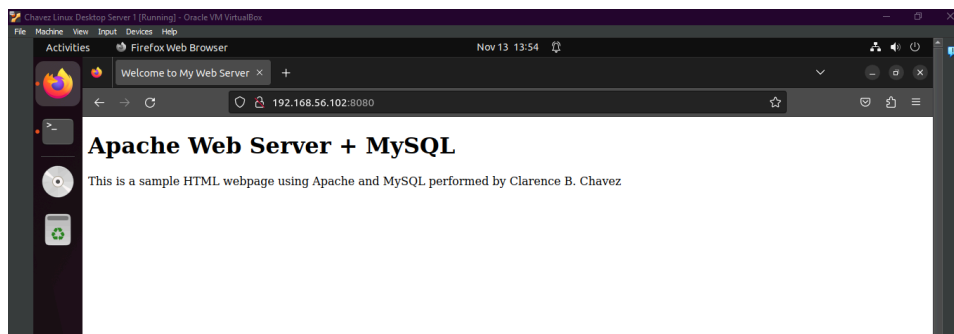
- name: Copy index.html to remote server
  copy:
    src: /home/qccchavez/CPE212_Chavez_HOA11.1/index.html
    dest: "/home/{{ ansible_user }}/docker-project/index.html"
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Allow port 8080 for web access in UFW
  ufw:
    rule: allow
    port: '8080'
    proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Build Docker Image using Dockerfile
  docker_image:
    name: apache-mysql
    source: build
    build:
      path: /home/{{ ansible_user }}/docker-project
      dockerfile: /home/{{ ansible_user }}/docker-project/Dockerfile
    force: yes
  when: ansible_distribution == "Ubuntu"

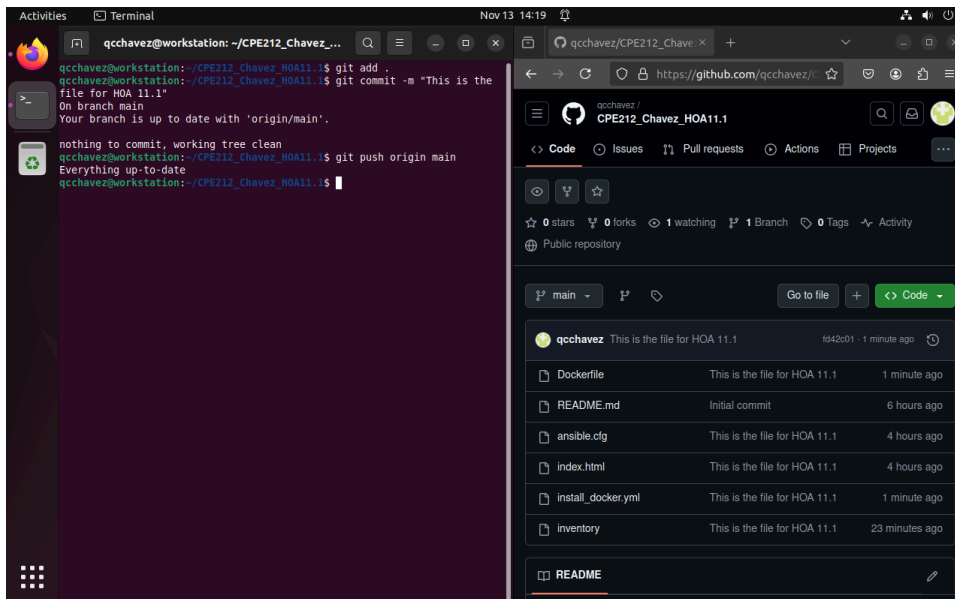
- name: Run Docker Container
  docker_container:
    name: apache-mysql-container
    image: apache-mysql:latest
    state: started
    restart_policy: always
    ports:
      - "8080:80"
      - "3306:3306"
  when: ansible_distribution == "Ubuntu"
```

- In this screenshot, these are the code for building the **Docker Image** with the use of **Dockerfile (which includes the web server and the database)**, and also running the **Docker Container**.



- In this screenshot, it shows that the **web server** worked alongside MySQL by prompting **192.168.56.102:8080** in the **web browser**.

Task 3.6



- In this screenshot, I've added, committed, and pushed my files to my Github repository.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- There are numerous benefits when implementing containerization, such as, portability, ensuring consistent operation across different environments. It provides an isolation between applications, which allows for more efficient resource usage, especially for low-end computers.

Conclusions:

- In this activity, it demonstrated the process of setting up Docker on a remote server using Ansible. Using Docker is indeed an efficient way to bundle several packages which makes deployment and application management much easier. Also, make sure to test and verify deployments step by step to prevent potential issues like permission problems which I've encountered.

Github Link: https://github.com/qcchavez/CPE212_Chavez_HOA11.1.git