Name: Calderon Ricardo B..	**Date Performed:** 04/16/2024
Course/Section: CPE232-CPE31S1	**Date Submitted:** 04/16/2024
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 2nd Sem Year 2023-2024

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.

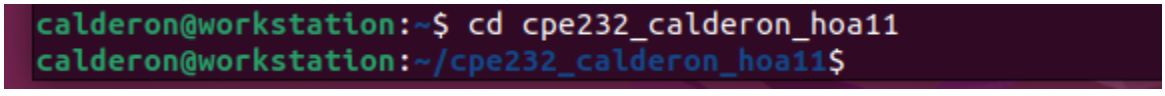
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

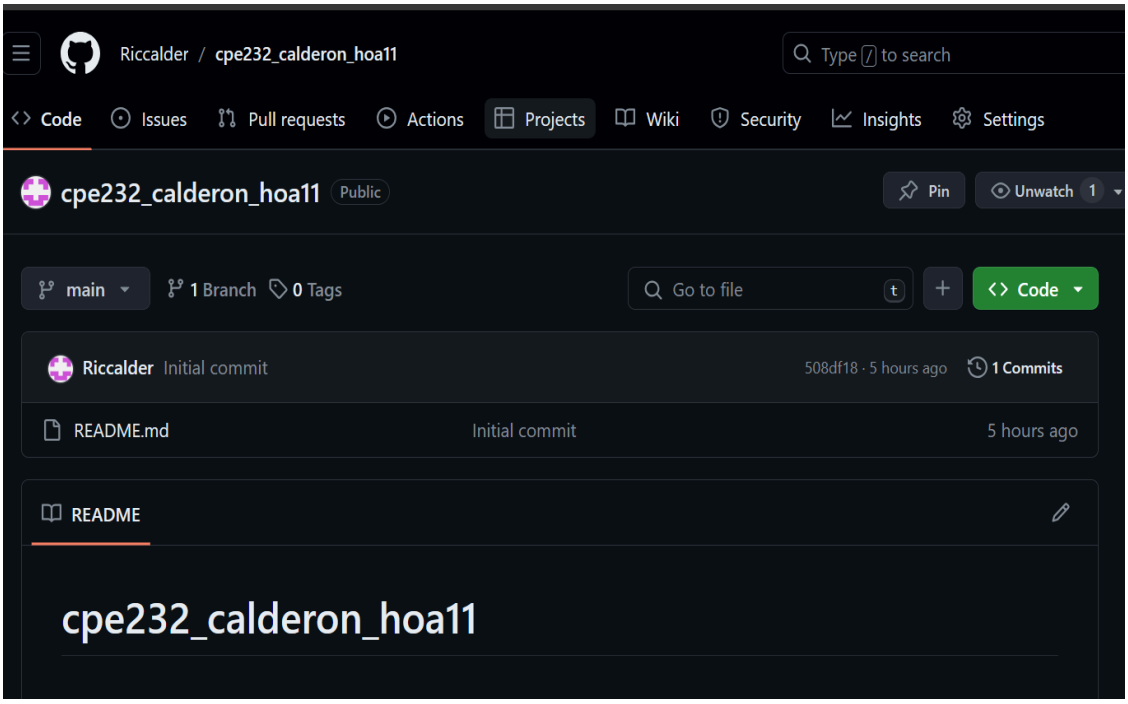
3. Tasks

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)

1. Create a new repository for this activity.






```
calderon@workstation:~/cpe232_calderon_hoa11$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = false

deprecation_warnings = false

remote _user = calderon
private_key_files = ~/.ssh/id_ed25519


calderon@workstation:~/cpe232_calderon_hoa11$
```

2. Install Docker and enable the docker socket.

```
calderon@workstation:~/cpe232_calderon_hoa11$ sudo apt install docker.io
[sudo] password for calderon:
Reading package lists... Done
Building dependency tree... Done
Files ng state information... Done
  ollowing packages were automatically installed and are no longer requir
  linux-headers-6.5.0-18-generic linux-hwe-6.5-headers-6.5.0-18
  linux-image-6.5.0-18-generic linux-modules-6.5.0-18-generic
  linux-modules-extra-6.5.0-18-generic
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 cgroupfs-mount | cgroup-lite 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 20 not upgraded.
Need to get 69.4 MB of archives.
After this operation, 266 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
calderon@workstation:~/cpe232_calderon_hoa11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Tue 2024-04-16 11:38:07 PST; 2min 40s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 4684 (dockerd)
      Tasks: 9
     Memory: 27.2M
        CPU: 760ms
     CGroup: /system.slice/docker.service
             └─4684 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Apr 16 11:38:04 workstation systemd[1]: Starting Docker Application Container E>
Apr 16 11:38:04 workstation dockerd[4684]: time="2024-04-16T11:38:04.655748265+>
Apr 16 11:38:04 workstation dockerd[4684]: time="2024-04-16T11:38:04.656801621+>
Apr 16 11:38:05 workstation dockerd[4684]: time="2024-04-16T11:38:05.941381129+>
Apr 16 11:38:06 workstation dockerd[4684]: time="2024-04-16T11:38:06.738835797+>
Apr 16 11:38:07 workstation dockerd[4684]: time="2024-04-16T11:38:07.112841841+>
Apr 16 11:38:07 workstation dockerd[4684]: time="2024-04-16T11:38:07.113473340+>
```

3. Add to Docker group to your current user.

```
calderon@workstation:~/cpe232_calderon_hoa11$ sudo usermod -aG docker calde
calderon@workstation:~/cpe232_calderon_hoa11$ grep docker /etc/group
docker:x:138:calderon
calderon@workstation:~/cpe232_calderon_hoa11$
```

4. Create a Dockerfile to install web and DB server.

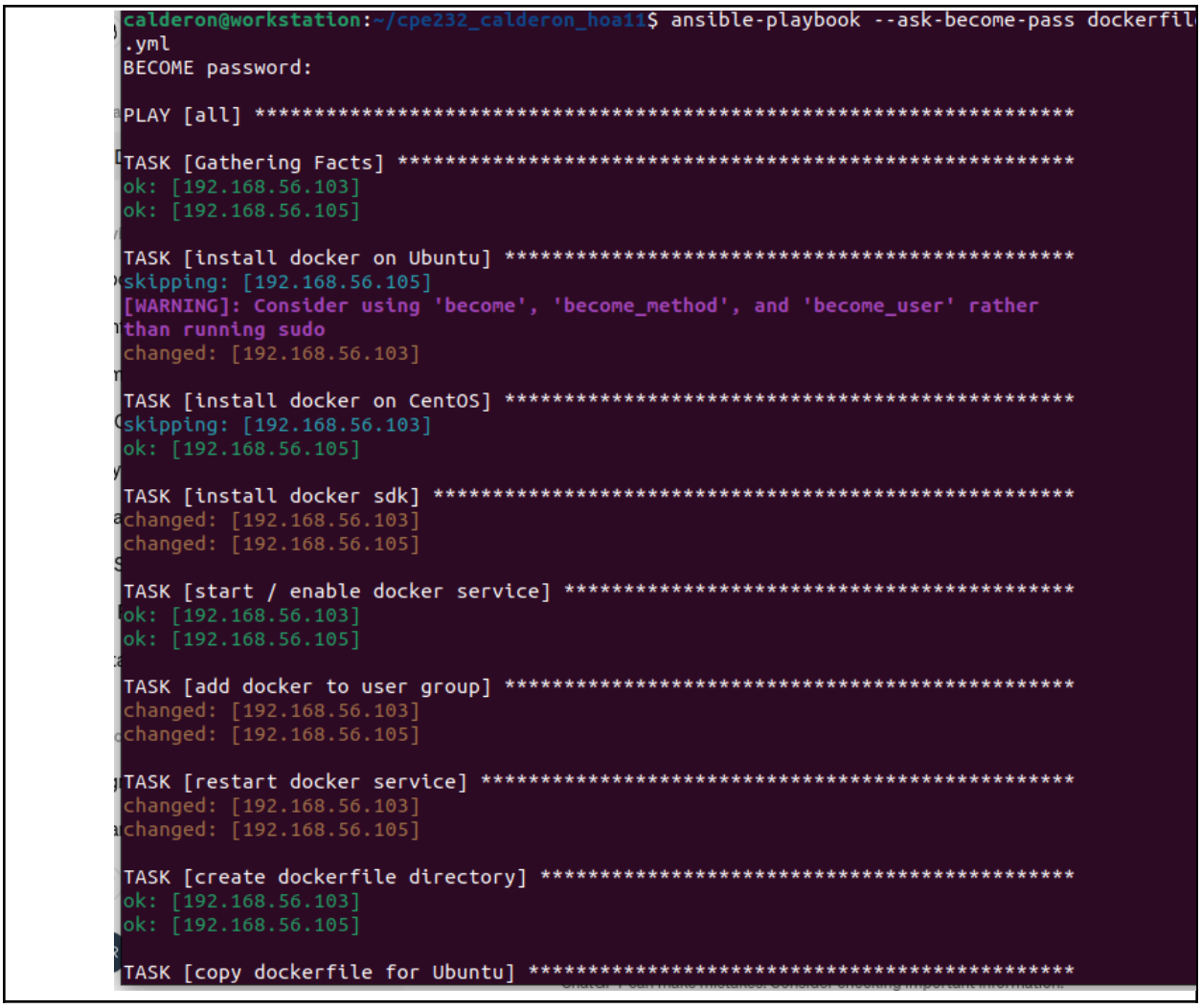
```
calderon@workstation:~/cpe232_calderon_hoa11$ cat dockerfile1
FROM ubuntu:latest
MAINTAINER calderon  <qrbcalderon@tip.edu.ph>

# skip prompts
ARG DEBIAN_FRONTEND=noninteractive

# update packages
RUN apt update
RUN apt upgrade -y

# install packages
RUN apt-get install -y apache2 mariadb-server

# set entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
calderon@workstation:~/cpe232_calderon_hoa11$
```

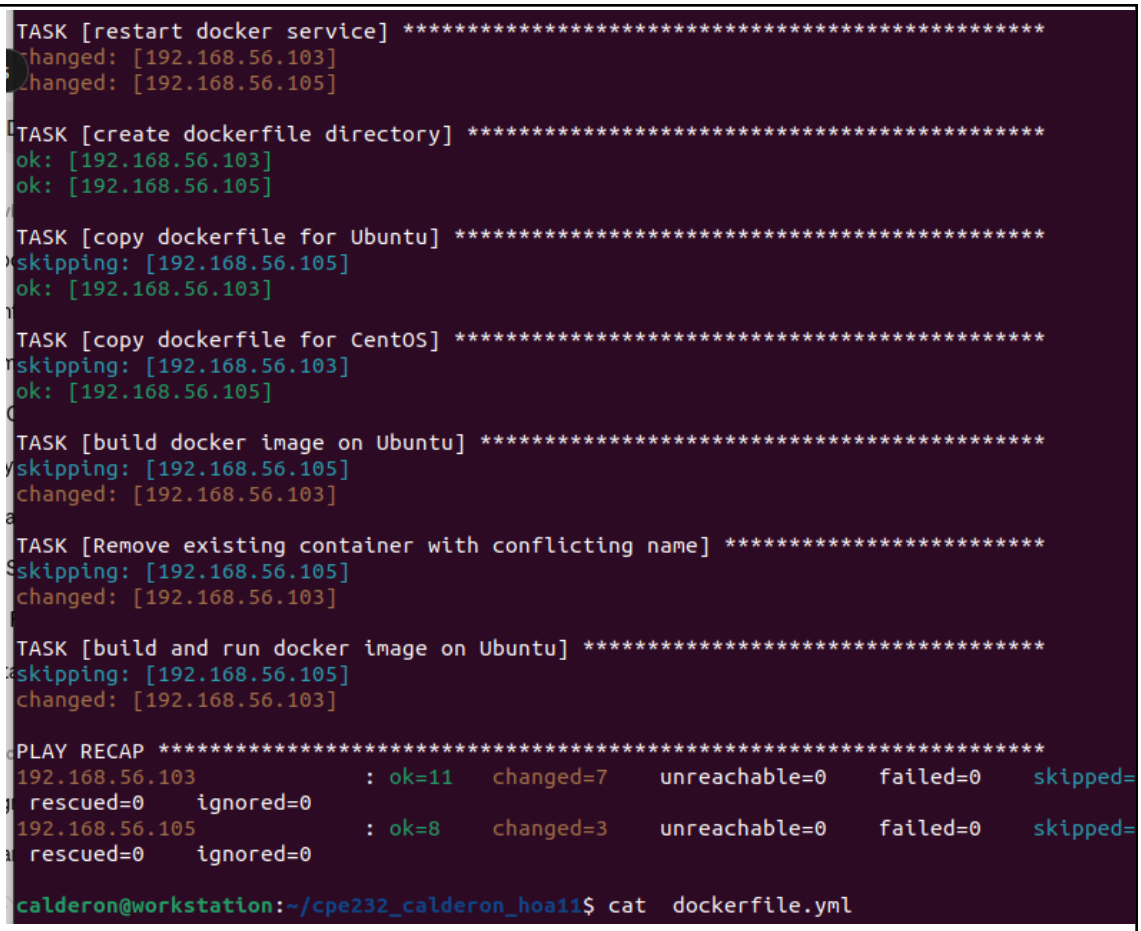
```
calderon@workstation:~/cpe232_calderon_hoa11$ cat dockerfile2
FROM centos:latest
MAINTAINER calderon  <qrbcalderon@tip.edu.ph>

# skip prompts
ARG DEBIAN_FRONTEND=noninteractive

# update packages
RUN dnf -y install epel-release && dnf -y update

# install packages
RUN yum install -y httpd mariadb-server

# set entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
calderon@workstation:~/cpe232_calderon_hoa11$
```

```
calderon@workstation:~/cpe232_calderon_hoa11$ cat  dockerfile.yml
---
- hosts: all
  become: true
  pre_tasks:

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

  - name: install docker on CentOS
    yum:
      name: docker
      state: present
    when: ansible_distribution == "CentOS"

  - name: install docker sdk
    shell:
      pip3 install docker-py

  - name: start / enable docker service
    service:
      name: docker
      state: started
      enabled: true

  - name: add docker to user group
    shell:
      usermod -aG docker calderon

  - name: restart docker service
    service:
      name: docker
      state: restarted
      enabled: true

  - name: create dockerfile directory
    file:
      path: /root/demo-dockerfile
```

```
  - name: copy dockerfile for CentOS
    copy:
      src: dockerfile2
      dest: /root/demo-dockerfile/dockerfile
      owner: root
      group: root
      mode: '0755'
    when: ansible_distribution == "CentOS"

  - name: build docker image on Ubuntu
    shell:
      cmd: docker build -t docker_image /root/demo-dockerfile
    when: ansible_distribution == "Ubuntu"

  - name: Remove existing container with conflicting name
    shell: docker rm -f docker_container5
    ignore_errors: yes  # Ignore errors if the container does not exist or is not running
    when: ansible_distribution == "Ubuntu"

  - name: build and run docker image on Ubuntu
    shell:
      cmd: docker run -d -p 8080 --name docker_container5 docker_image
    when: ansible_distribution == "Ubuntu"

calderon@workstation:~/cpe232_calderon_hoa11$ nano dockerfile.yml
```

5. Install and build the Dockerfile using Ansible.

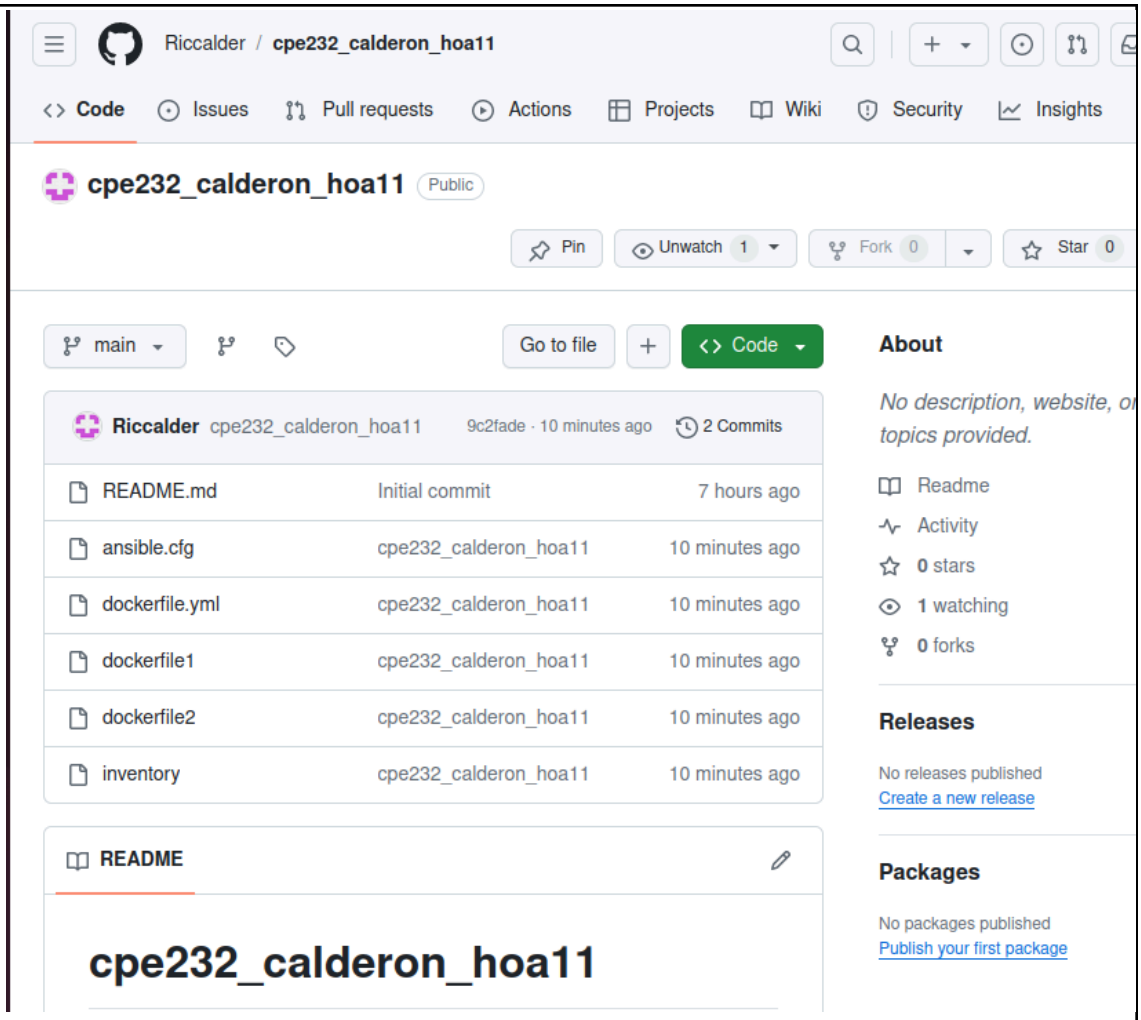

calderon@workstation:~/cpe232_calderon_hoa11$ ansible-playbook --ask-become-pass dockerfil
.yml
BECOME password:
PLAY [all] ***
TASK [Gathering Facts] ***
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [install docker on Ubuntu] **
skipping: [192.168.56.105]
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
changed: [192.168.56.103]
TASK [install docker on CentOS] **
skipping: [192.168.56.103]
ok: [192.168.56.105]
TASK [install docker sdk] **
changed: [192.168.56.103]
changed: [192.168.56.105]
TASK [start / enable docker service] ***
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [add docker to user group] **
changed: [192.168.56.103]
changed: [192.168.56.105]
TASK [restart docker service] **
changed: [192.168.56.103]
changed: [192.168.56.105]
TASK [create dockerfile directory] ***
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [copy dockerfile for Ubuntu] **



6. Add, commit and push it to your repository.

```
calderon@workstation:~/cpe232_calderon_hoa11$ git push origin main
Username for 'https://github.com': Riccalder
Password for 'https://Riccalder@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.50 KiB | 1.50 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Riccalder/cpe232_calderon_hoa11.git
   508df18..9c2fade  main -> main
calderon@workstation:~/cpe232_calderon_hoa11$
```

```
calderon@workstation:~/cpe232_calderon_hoa11$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ansible.cfg
        dockerfile.yml
        dockerfile1
        dockerfile2
        inventory

nothing added to commit but untracked files present (use "git add" to track)
calderon@workstation:~/cpe232_calderon_hoa11$ git add *
calderon@workstation:~/cpe232_calderon_hoa11$ git commit -m "cpe232_calderon_hoa11
[main 9c2fade] cpe232_calderon_hoa11
 5 files changed, 124 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile.yml
 create mode 100644 dockerfile1
 create mode 100644 dockerfile2
 create mode 100644 inventory
```



Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization offers a transformative approach to software development, delivering consistency, isolation, portability, efficiency, agility, and scalability. By embracing containerization, organizations can streamline their workflows, enhance security, and accelerate innovation, ultimately driving business success in today's fast-paced digital landscape.

Conclusions:

In this activity, I have learned about the importance and benefits of containerization. It enables us to perform software development with ease and much faster than before. Containerization simplifies our complicated problems and tasks by providing contemporary software delivery techniques.