Name: Buduan, Christian Aaron C.	**Date Performed:** 13/11/2024
Course/Section: CpE31S2	**Date Submitted:** 13/11/2024
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/

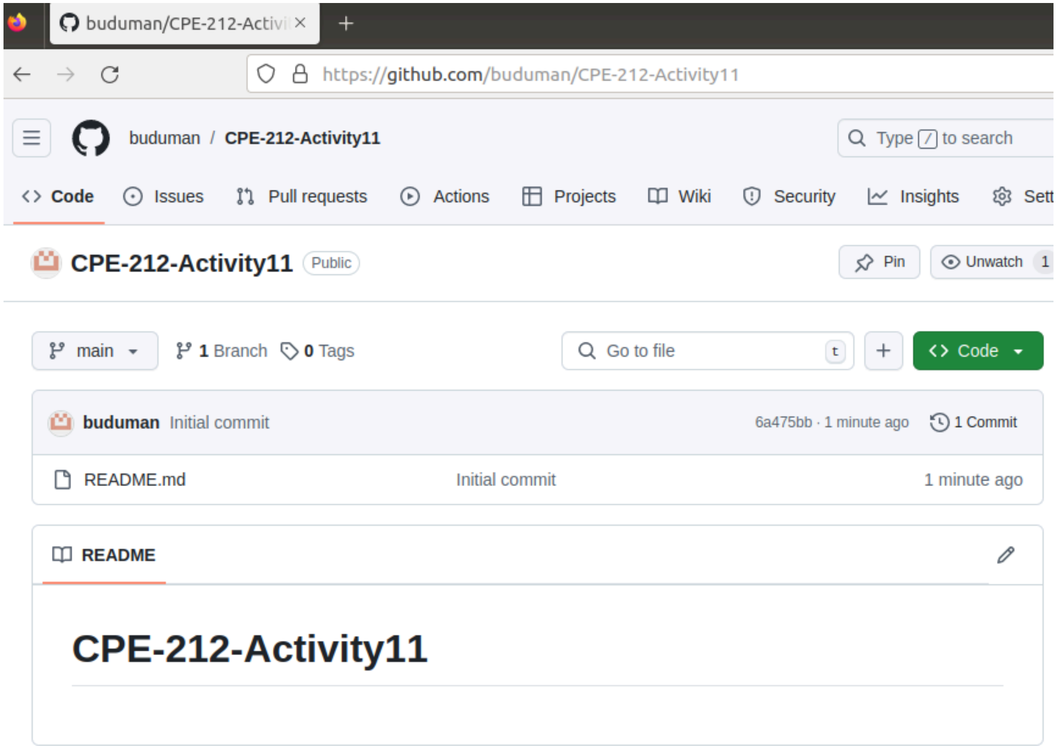
You may also check the difference between containers and virtual machines. Click the link given below.

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

3. Tasks

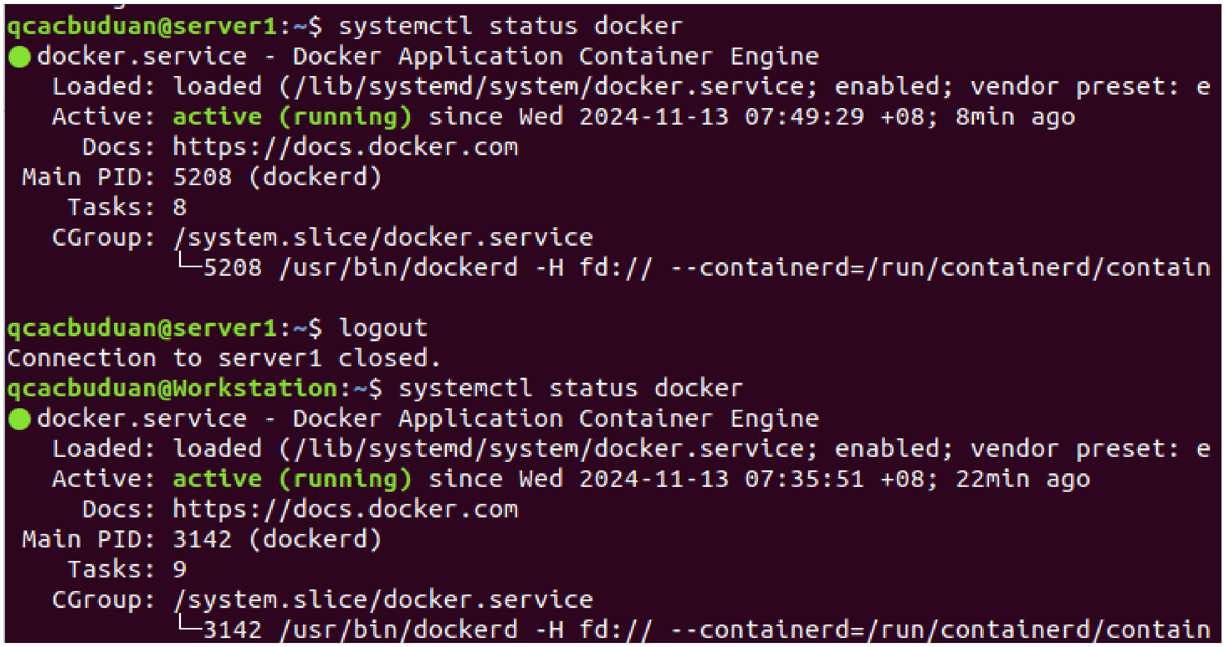
1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. Output (screenshots and explanations)



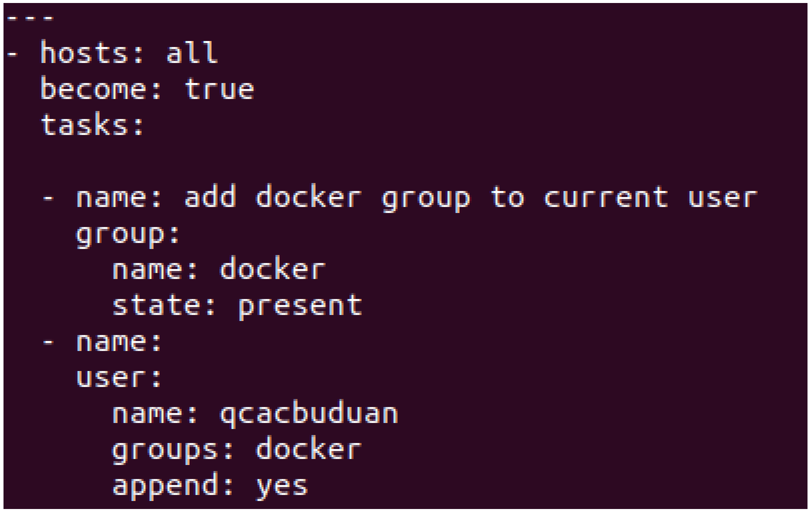
CPE-212-Activity11

Create a new repository for this activity



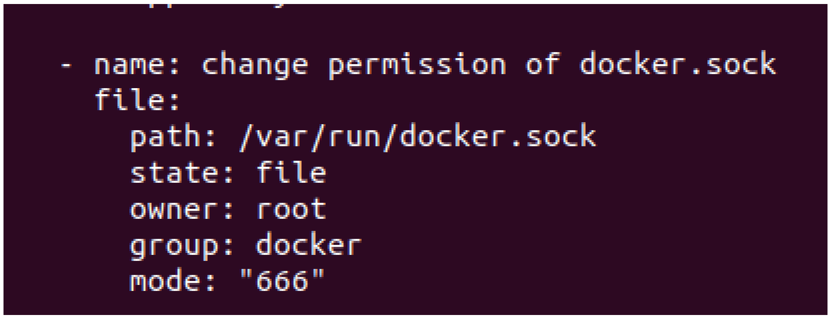
```
qcacbuduan@server1:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Wed 2024-11-13 07:49:29 +08; 8min ago
     Docs: https://docs.docker.com
 Main PID: 5208 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─5208 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain

qcacbuduan@server1:~$ logout
Connection to server1 closed.
qcacbuduan@Workstation:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Wed 2024-11-13 07:35:51 +08; 22min ago
     Docs: https://docs.docker.com
 Main PID: 3142 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           └─3142 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
```

install docker



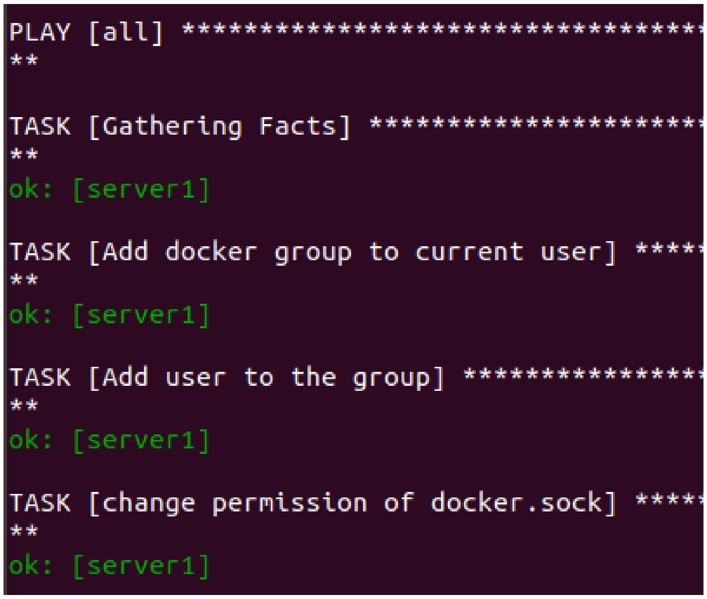
```
---
- hosts: all
  become: true
  tasks:

  - name: add docker group to current user
    group:
      name: docker
      state: present
  - name:
    user:
      name: qcacbuduan
      groups: docker
      append: yes
```

Create a new file named *main.yml* and Automate task #3 using ansible playbook with the following commands



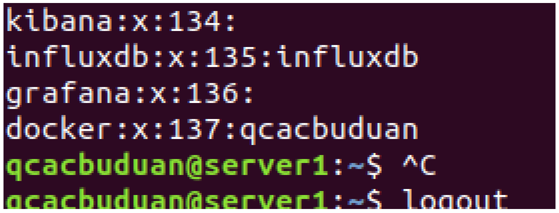
```
  - name: change permission of docker.sock
    file:
      path: /var/run/docker.sock
      state: file
      owner: root
      group: docker
      mode: "666"
```

In the same file, use the following prompt above to enable the docker socket.



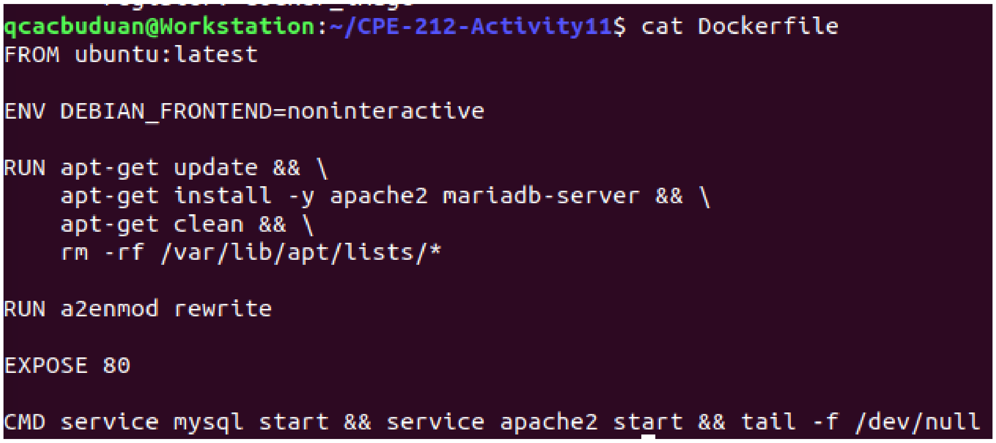
```
PLAY [all] ***********************************
**

TASK [Gathering Facts] ***********************
**
ok: [server1]

TASK [Add docker group to current user] *****
**
ok: [server1]

TASK [Add user to the group] ****************
**
ok: [server1]

TASK [change permission of docker.sock] *****
**
ok: [server1]
```

run the ansible-playbook using the command *ansible-playbook –ask-become-pass main.yml*



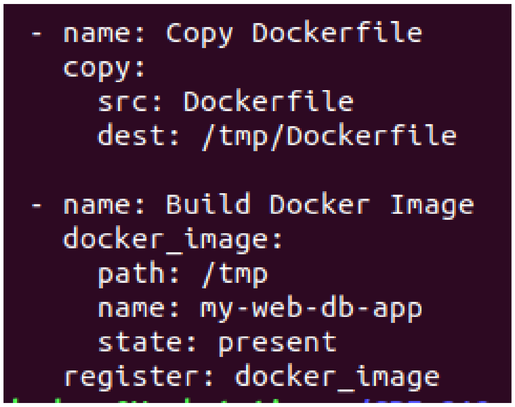
```
kibana:x:134:
influxdb:x:135:influxdb
grafana:x:136:
docker:x:137:qcacbuduan
qcacbuduan@server1:~$ ^C
qcacbuduan@server1:~$ logout
```

verify if the group you created is added to the user



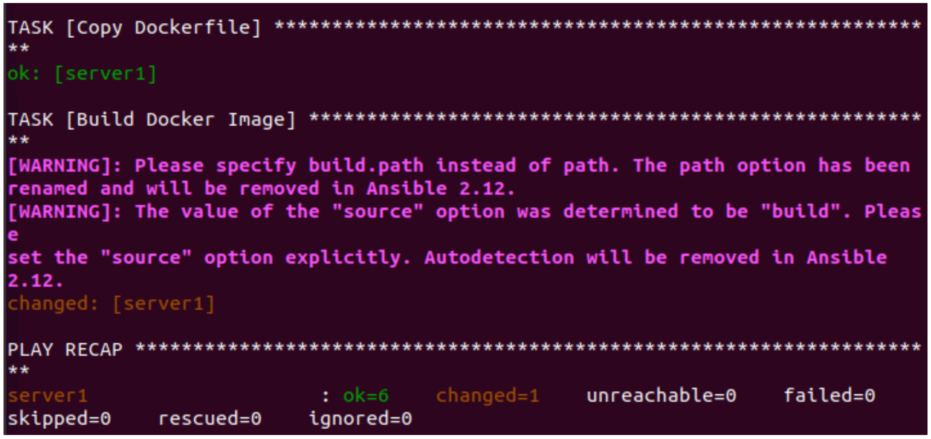
```
qcacbuduan@Workstation:~/CPE-212-Activity11$ cat Dockerfile
FROM ubuntu:latest

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && \
    apt-get install -y apache2 mariadb-server && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*

RUN a2enmod rewrite

EXPOSE 80

CMD service mysql start && service apache2 start && tail -f /dev/null
```

Create a Dockerfile to install web and DB server

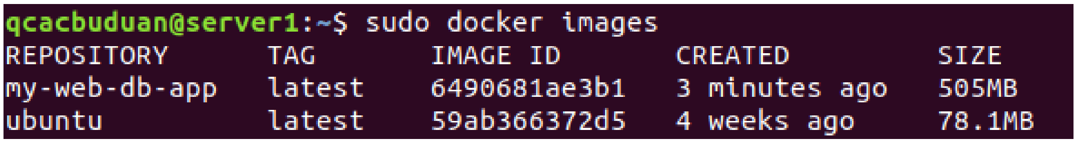


```
 - name: Copy Dockerfile
   copy:
     src: Dockerfile
     dest: /tmp/Dockerfile

 - name: Build Docker Image
   docker_image:
     path: /tmp
     name: my-web-db-app
     state: present
   register: docker_image
```

Install and build the Dockerfile in an ubuntu server using Ansible by Edit the main.yml file and add the following above.

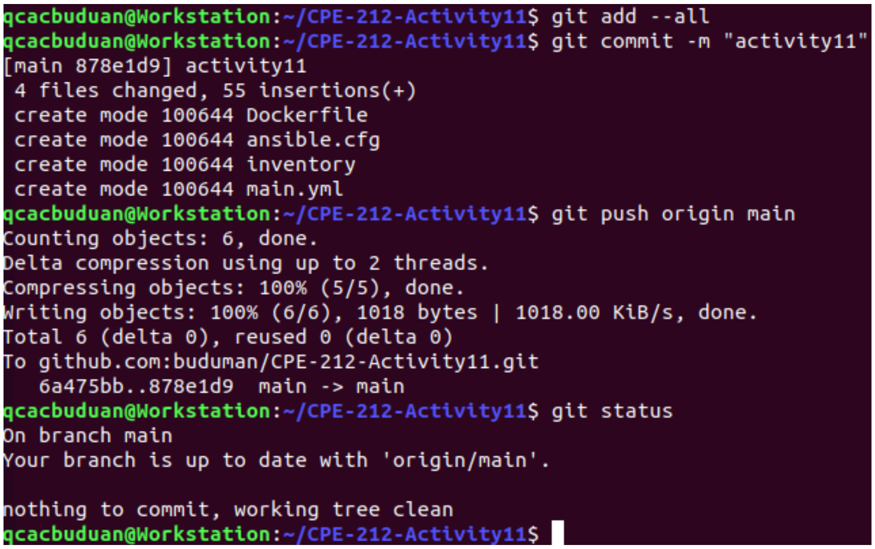


```
TASK [Copy Dockerfile] ******************************************************************
**
ok: [server1]

TASK [Build Docker Image] ***************************************************************
**
[WARNING]: Please specify build.path instead of path. The path option has been
renamed and will be removed in Ansible 2.12.
[WARNING]: The value of the "source" option was determined to be "build". Pleas
e
set the "source" option explicitly. Autodetection will be removed in Ansible
2.12.
changed: [server1]

PLAY RECAP ******************************************************************************
**
server1                    : ok=6    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```



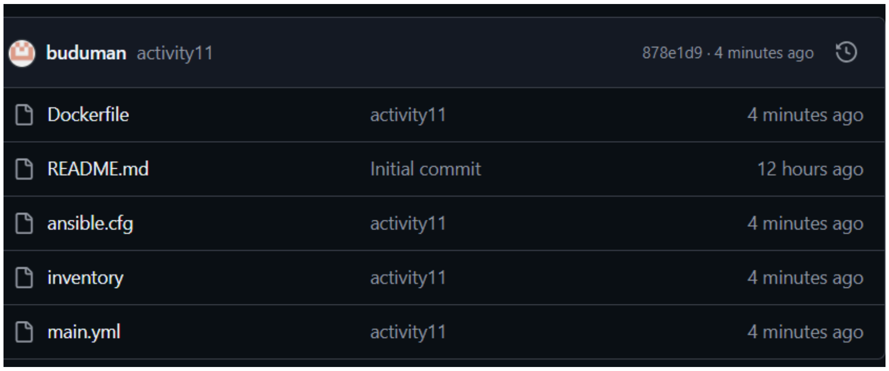
```
qcacbuduan@server1:~$ sudo docker images
REPOSITORY      TAG       IMAGE ID       CREATED         SIZE
my-web-db-app   latest    6490681ae3b1   3 minutes ago   505MB
ubuntu          latest    59ab366372d5   4 weeks ago     78.1MB
```

see if building the image was successful by using the command *docker images*



```
qcacbuduan@Workstation:~/CPE-212-Activity11$ git add --all
qcacbuduan@Workstation:~/CPE-212-Activity11$ git commit -m "activity11"
[main 878e1d9] activity11
 4 files changed, 55 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 main.yml
qcacbuduan@Workstation:~/CPE-212-Activity11$ git push origin main
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1018 bytes | 1018.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:buduman/CPE-212-Activity11.git
   6a475bb..878e1d9  main -> main
qcacbuduan@Workstation:~/CPE-212-Activity11$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
qcacbuduan@Workstation:~/CPE-212-Activity11$
```

Add, commit, and push the files into my github repository



buduman activity11 878e1d9 · 4 minutes ago

Dockerfile	activity11	4 minutes ago
README.md	Initial commit	12 hours ago
ansible.cfg	activity11	4 minutes ago
inventory	activity11	4 minutes ago
main.yml	activity11	4 minutes ago

Github repository: https://github.com/buduman/CPE-212-Activity11.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   Some of the benefits of implementing containerization is that it doesn’t affect or rely on the system’s configurations, making it portable and versatile. It means that it would perform the same even if it is on other devices and on other operating systems. Another benefit of implementing containerization is that it is lightweight and efficient, which means that it doesn’t slow down the performance of your device by a huge margin.

Conclusions:

- After doing this activity, I was able to learn more about the functions of docker, how containerization works, and learn the basics of docker such as building a docker image. This activity strengthened the foundation of what I have learned so far in using docker and implementing containerization while still applying it into ansible-playbook.