

Name: Jose Mari Tan Dela Peña	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.

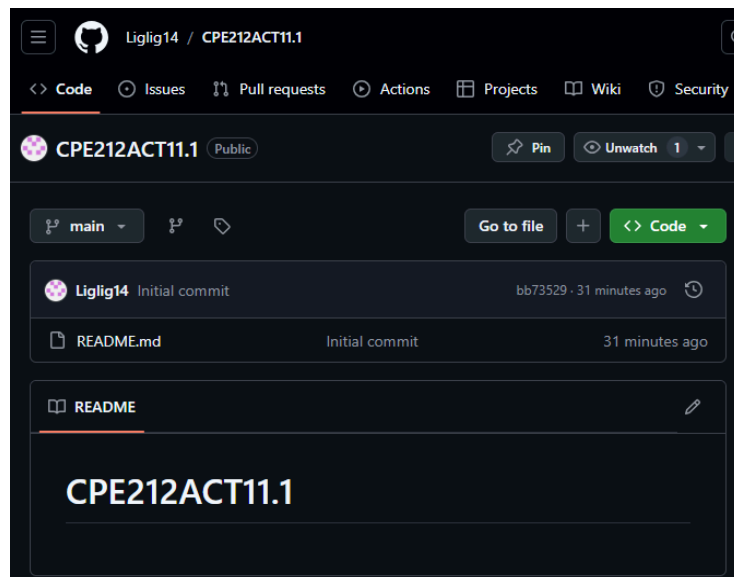
Source: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm>

3. Tasks

1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add a Docker group to your current user.
4. Create a Dockerfile to install web and DB servers.
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.



2.) Install Docker and enable the docker socket.

```
jose@workstation:~/CPE212ACT11.1$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-11-13 08:28:32 PST; 6min ago
 TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 16024 (dockerd)
      Tasks: 9
     Memory: 22.8M (peak: 23.8M)
        CPU: 601ms
     CGroup: /system.slice/docker.service
            └─16024 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

3.) Add a Docker group to your current user.

```
jose@workstation:~/CPE212ACT11.1$ ansible-playbook --ask-become-pass install.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.100]
ok: [192.168.56.101]

TASK [update repository index (Ubuntu)] *****
ok: [192.168.56.100]
ok: [192.168.56.101]

PLAY [UbuntuServers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.100]
ok: [192.168.56.101]

TASK [DockerGroup : Ensure docker group exists] *****
ok: [192.168.56.100]
ok: [192.168.56.101]

TASK [DockerGroup : Add current user to docker group] *****
changed: [192.168.56.100]
changed: [192.168.56.101]

PLAY RECAP *****
192.168.56.100      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.101      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4.) Create a Dockerfile to install web and DB servers.

```
jose@workstation:~/CPE212ACT11.1$ docker build -t my-apache-mariadb .
[+] Building 218.7s (11/11) FINISHED          docker:default
=> [internal] load .dockerignore               0.6s
=> => transferring context: 2B                 0.0s
=> [internal] load build definition from Dockerfile 0.6s
=> => transferring dockerfile: 849B            0.1s
=> [internal] load metadata for docker.io/library/ubuntu:20.04 4.4s
=> [1/6] FROM docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead0 16.6s
=> => resolve docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead07 0.1s
=> => sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f1 27.51MB / 27.51MB 12.9s
=> => sha256:8e5c4f0285ecbb4ead070431d29b576a530d3166df7 6.69kB / 6.69kB 0.0s
=> => sha256:e5a6aeef391a8a9bdaee3de6b28f393837c479d8217324a 424B / 424B 0.0s
=> => sha256:6013ae1a63c2ee58a8949f03c6366a3ef6a2f386a7d 2.30kB / 2.30kB 0.0s
=> => extracting sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f116e11cd9 2.6s
=> [internal] load build context               0.2s
=> => transferring context: 2.97kB             0.1s
=> [2/6] RUN apt-get update && apt-get install -y apt-utils 168.7s
=> [3/6] COPY ansible.cfg /etc/ansible/ansible.cfg 0.4s
=> [4/6] COPY inventory /etc/ansible/inventory 0.2s
=> [5/6] COPY install.yml /etc/ansible/install.yml 0.2s
=> [6/6] COPY roles /etc/ansible/roles 0.2s
=> exporting to image                          26.8s
=> => exporting layers                          26.7s
=> => writing image sha256:7240cf0ddd41ee7ab5dbc380c8af1d6bd18f2889bbfb8 0.1s
=> => naming to docker.io/library/my-apache-mariadb 0.0s
jose@workstation:~/CPE212ACT11.1$ sudo docker images
```

5.) Install and build the Dockerfile using Ansible. (Success)

Control Node:

```
jose@workstation:~/CPE212ACT11.1$ sudo docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
my-apache-mariadb   latest      7240cf0ddda1     2 minutes ago   363MB
```

Manage Node:

```
jose@server1:~$ sudo docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
jose0714/my-apache-mariadb   latest      7240cf0ddda1     15 minutes ago   363MB
```

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

```
jose@workstation:~/CPE212ACT11.1$ git add Dockerfile
jose@workstation:~/CPE212ACT11.1$ git add ansible.cfg
jose@workstation:~/CPE212ACT11.1$ git add install.yml
jose@workstation:~/CPE212ACT11.1$ git add inventory
jose@workstation:~/CPE212ACT11.1$ git add roles
jose@workstation:~/CPE212ACT11.1$ git commit -m "11/13/2024"
^[[3~[main b3fc90d] 11/13/2024
 6 files changed, 146 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 ansible.cfg
 create mode 100644 install.yml
 create mode 100644 inventory
 create mode 100644 roles/DockerGroup/tasks/main.yml
 create mode 100644 roles/UbuntuDocker/tasks/main.yml
jose@workstation:~/CPE212ACT11.1$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (13/13), 2.05 KiB | 174.00 KiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Liglig14/CPE212ACT11.1.git
   bb73529..b3fc90d  main -> main
jose@workstation:~/CPE212ACT11.1$
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- Containerization works by packaging an application, software packages in an isolated "container." This container runs independently over the host system's OS, but it doesn't affect or rely on the system's configurations, making it highly portable. Docker packages an application with everything it needs to run, so it performs the same even if it's on a laptop, a desktop or live servers. This consistency reduces the chance of issues or errors when moving an app between different operating systems.

Conclusions:

- Our instructor said that Docker is very versatile, lightweight, and efficient, and I never realize that it was until I had a problem earlier in which I built the Docker Image from the Docker File on my control node instead of manage node, and I had to learn how to utilize the push and pull function of Docker which alleviates the file transfer through container which is also the objective of this activity (Continuous Delivery process) which was done through the use of Ansible as an Infrastructure as Code.

Github Repository Link: <https://github.com/Liglig14/CPE212ACT11.1>

The screenshot shows the GitHub interface for the repository 'CPE212ACT11.1' by user 'Liglig14'. The repository is public and has 1 watcher. The main branch is selected. The file list shows the following files and their commit history:

File	Commit	Time
roles	11/13/2024	now
Dockerfile	11/13/2024	now
README.md	Initial commit	2 hours ago
ansible.cfg	11/13/2024	now
install.yml	11/13/2024	now
inventory	11/13/2024	now

The README section is visible, showing the repository name 'CPE212ACT11.1'.