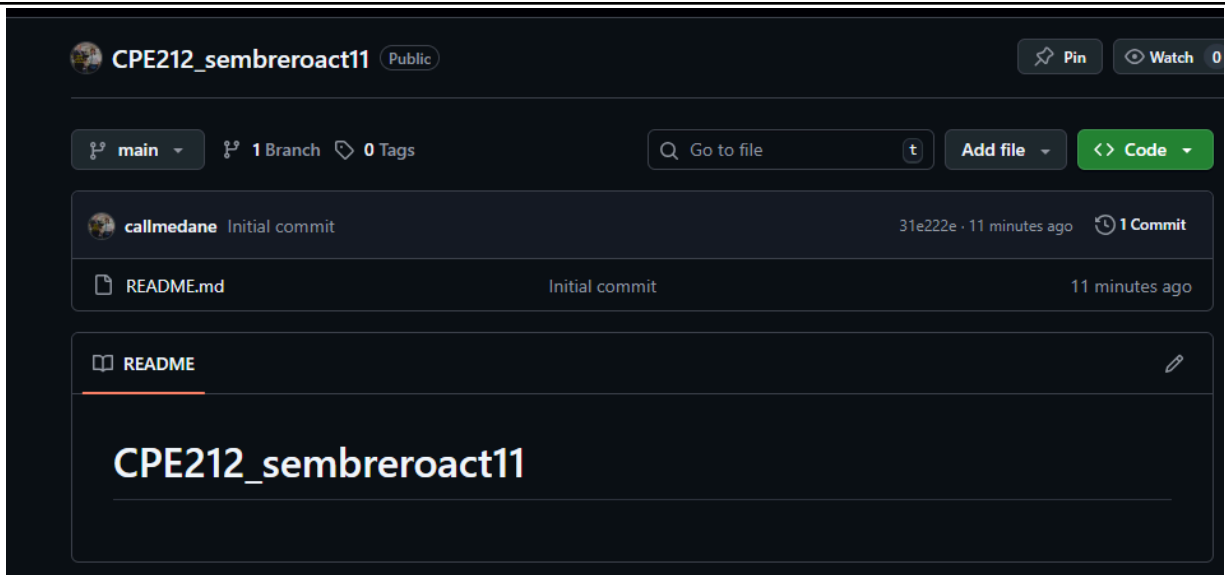


Name: Sembrero, Christian Dane D.	Date Performed: 10/24/25
Course/Section: CPE212-CPE31S2	Date Submitted: 10/24/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem 2025-2026
Activity 11: Containerization	
1. Objectives	
Create a Dockerfile and form a workflow using Ansible as Infrastructure 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)	



1. Created a new repo

```
vboxuser@Workstation:~/CPE212_sembbreroact11$ sudo apt install -y docker.io
Unpacking ubuntu-fan (0.12.10) ...
Setting up runc (1.1.4-0ubuntu1~18.04.2) ...
Setting up containerd (1.6.12-0ubuntu1~18.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up pigz (2.4-1) ...
Setting up docker.io (20.10.21-0ubuntu1~18.04.3) ...
Adding group `docker' (GID 127) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for systemd (237-3ubuntu10.57) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
vboxuser@Workstation:~/CPE212_sembbreroact11$
```

```

vboxuser@Workstation:~/CPE212_sembbreroact11$ sudo systemctl enable --now d
ocker
Docker version 20.10.21, build 20.10.21-0ubuntu1~18.04.3
○ vboxuser@Workstation:~/CPE212_sembbreroact11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor pre
   Active: active (running) since Fri 2025-10-24 17:40:47 +08; 2min 24s ag
   Docs: https://docs.docker.com
   Main PID: 10181 (dockerd)
   Tasks: 8
   CGroup: /system.slice/docker.service
           └─10181 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/

Oct 24 17:40:45 Workstation dockerd[10181]: time="2025-10-24T17:40:45.7552
Oct 24 17:40:45 Workstation dockerd[10181]: time="2025-10-24T17:40:45.7552
Oct 24 17:40:45 Workstation dockerd[10181]: time="2025-10-24T17:40:45.7553
Oct 24 17:40:45 Workstation dockerd[10181]: time="2025-10-24T17:40:45.7554
Oct 24 17:40:45 Workstation dockerd[10181]: time="2025-10-24T17:40:45.9109
Oct 24 17:40:46 Workstation dockerd[10181]: time="2025-10-24T17:40:46.8855
Oct 24 17:40:47 Workstation dockerd[10181]: time="2025-10-24T17:40:47.2243
Oct 24 17:40:47 Workstation dockerd[10181]: time="2025-10-24T17:40:47.2246
Oct 24 17:40:47 Workstation systemd[1]: Started Docker Application Contain
Oct 24 17:40:47 Workstation dockerd[10181]: time="2025-10-24T17:40:47.4119
lines 1-19/19 (END)

```

2. What I did was a manual installation of docker, I was having frequent errors in the playbook in terms of Syntax errors. I was able to install and enable docker.

```
vboxuser@Workstation:~/CPE212_sembbreroact11$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:6dc565aa630927052111f823c303948cf83670a3903ffa3849f1488ab517f891
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

3. I was able to add the docker group to my current user using `sudo usermod -aG, this will allow me to run docker without using sudo. And I just ran docker hello-world to verify.`

```
vboxuser@Workstation:~/CPE212_sembreroact11$ docker build -t web_db_image .
Sending build context to Docker daemon 61.44kB
Step 1/5 : FROM ubuntu:18.04
18.04: Pulling from library/ubuntu
7c457f213c76: Pull complete
Digest: sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
Status: Downloaded newer image for ubuntu:18.04
---> f9a80a55f492
Step 2/5 : LABEL maintainer="vboxuser <qcddsebrero@tip.edu.ph"
---> Running in 3f36c015b76d
Removing intermediate container 3f36c015b76d
---> 9d9cbde95806
Step 3/5 : RUN apt update -y && apt install -y nginx mariadb-server && apt clean
---> Running in dd74103fd4b1

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [102 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [33
73 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [102 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [102 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
1
```

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...

Building dependency tree...

Reading state information...

The following additional packages will be installed:

fontconfig-config fonts-dejavu-core galera-3 gawk geoip-database iproute2
libaio1 libatm1 libbsd0 libcgi-fast-perl libcgi-pm-perl
libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libelf1
libencode-locale-perl libexpat1 libfcgi-perl libfontconfig1 libfreetype6
libgd3 libgdbm-compat4 libgdbm5 libgeoip1 libhtml-parser-perl
libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
libhttp-message-perl libicu60 libio-html-perl libjbig0 libjemalloc1
libjpeg-turbo8 libjpeg8 liblwp-mediatypes-perl libmnl0 libmpfr6
libmysqlclient20 libnginx-mod-http-geoip libnginx-mod-http-image-filter

Get:82 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnginx-mod-

stream amd64 1.14.0-0ubuntu1.11 [63.5 kB]

Get:84 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libterm-readkey-p
erl amd64 2.37-1build1 [24.4 kB]

Get:85 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 mariadb-s
erver all 1:10.1.48-0ubuntu0.18.04.1 [12.9 kB]

Get:86 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 nginx-core am
d64 1.14.0-0ubuntu1.11 [413 kB]

Get:87 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 nginx all 1.1
4.0-0ubuntu1.11 [3596 B]

debconf: delaying package configuration, since apt-utils is not installed

Fetched 50.4 MB in 33s (1518 kB/s)

Selecting previously unselected package libmpfr6:amd64.

(Reading database ... 4050 files and directories currently installed.)

Preparing to unpack .../libmpfr6_4.0.1-1_amd64.deb ...

Unpacking libmpfr6:amd64 (4.0.1-1) ...

Selecting previously unselected package readline-common.

Preparing to unpack .../readline-common_7.0-3_all.deb ...

Unpacking readline-common (7.0-3) ...

Selecting previously unselected package libreadline7:amd64.

Preparing to unpack .../libreadline7_7.0-3_amd64.deb ...

Unpacking libreadline7:amd64 (7.0-3) ...

Selecting previously unselected package libsigsegv2:amd64.

Preparing to unpack .../libsigsegv2_2.12-1_amd64.deb ...

Unpacking libsigsegv2:amd64 (2.12-1) ...

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

```
Removing intermediate container dd74103fd4b1
---> ed5b55c60432
Step 4/5 : EXPOSE 80 3306
---> Running in 603c2cdb9c80
Removing intermediate container 603c2cdb9c80
---> 332c960f5f50
Step 5/5 : CMD service mysql start && nginx -g 'daemon off;'
---> Running in 119fb27b3bdb
Removing intermediate container 119fb27b3bdb
---> 83dc0a0daa1a
Successfully built 83dc0a0daa1a
Successfully tagged web_db_image:latest
vboxuser@Workstation:~/CPE212_sembbreroact11$
```

4. I was able to build and run Docker images successfully. And the exposed ports let me access both web and database services outside the container.

```
vboxuser@Workstation:~/CPE212_sembbreroact11$ ansible-playbook build_docker_image.yaml -K

PLAY [Install Docker and build Docker image] *****

TASK [Gathering Facts] *****
ok: [192.168.56.110]
[WARNING]: Module invocation had junk after the JSON data:
AttributeError("module 'platform' has no attribute 'dist'")
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [Install prerequisite packages] *****
fatal: [192.168.56.112]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: 'apt-get'", "rc": 2}
ok: [192.168.56.110]
ok: [192.168.56.109]

TASK [Add Docker official GPG key] *****
ok: [192.168.56.110]
ok: [192.168.56.109]
```

```

TASK [Add Docker repository] *****
ok: [192.168.56.110]
ok: [192.168.56.109]

TASK [Install Docker engine] *****
ok: [192.168.56.110]
ok: [192.168.56.109]

TASK [Ensure Docker service is running] *****
ok: [192.168.56.110]
ok: [192.168.56.109]

TASK [Copy Dockerfile to remote host] *****
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: /home/vboxuser/CPE212_sembroact11/./Dockerfile
fatal: [192.168.56.109]: FAILED! => {"changed": false, "msg": "Could not find or access './Dockerfile'\nSearched in:\n\t/home/vboxuser/CPE212_sembroact11/files/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/files/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/./Dockerfile"}
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: /home/vboxuser/CPE212_sembroact11/./Dockerfile
fatal: [192.168.56.110]: FAILED! => {"changed": false, "msg": "Could not find or access './Dockerfile'\nSearched in:\n\t/home/vboxuser/CPE212_sembroact11/files/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/files/./Dockerfile\n\t/home/vboxuser/CPE212_sembroact11/./Dockerfile"}
to retry, use: --limit @/home/vboxuser/CPE212_sembroact11/build_docker_image.retry

PLAY RECAP *****
192.168.56.109      : ok=6    changed=0    unreachable=0    failed=1
192.168.56.110      : ok=6    changed=0    unreachable=0    failed=1
192.168.56.112      : ok=1    changed=0    unreachable=0    failed=1

```

5. I was able to install Docker on both remote hosts(Ubuntu) except the CentOS. And I copy the Dockerfile because Ansible can't find the dockerfile to my workstation. Unfortunately, I wasn't able to run some of tasks and I also deleted a task called build docker image due to syntax error.

Github Repo Link:

https://github.com/callmedane/CPE212_sembroact11.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization allows you to package your application with everything it needs to run in a "box" so it will run anywhere, on your laptop, a server, or in the cloud. It is fast, keeps things clean, is easy to move around, and makes scaling easy.

Conclusions:

I learned how Docker and Ansible function for containerization. Despite a few syntax errors, I was successful with running a Docker container and grasped how containers make deployment easier and quicker to manage.

