

Name: Solis, Paul Vincent M.	Date Performed: 10/24/25
Course/Section: CPE212 - CPE31S2	Date Submitted: 10/24/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st sem yr 25-26

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.

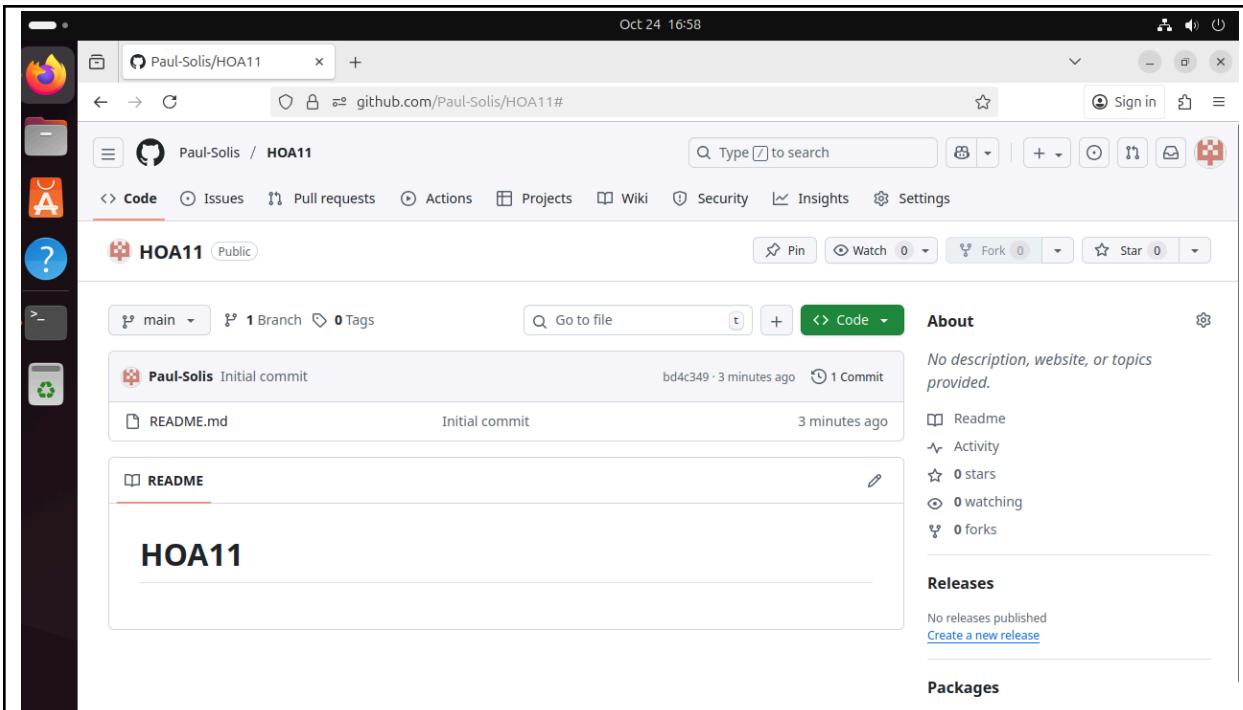
Source: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm>

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.



In this number 1. The task is to create a repository for the docker file so i created one for the docker file.

2.

```
paul@Workstation:~/HOA11$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-amber libllvm17t64 libllvm19
  linux-headers-6.14.0-27-generic linux-hwe-6.14-headers-6.14.0-27
  linux-hwe-6.14-tools-6.14.0-27 linux-image-6.14.0-27-generic
  linux-modules-6.14.0-27-generic linux-modules-extra-6.14.0-27-generic
  linux-tools-6.14.0-27-generic python3-netifaces
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 btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-buildx docker-compose-v2 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 2 not upgraded.
Need to get 75.6 MB of archives.
After this operation, 287 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1
```

```
paul@Workstation:~/HOA11$ sudo systemctl enable docker
paul@Workstation:~/HOA11$ sudo systemctl start docker
paul@Workstation:~/HOA11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: en>
  Active: active (running) since Fri 2025-10-24 17:01:22 PST; 53s ago
    TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 5741 (dockerd)
     Tasks: 10
    Memory: 21.9M (peak: 22.5M)
      CPU: 888ms
     CGroup: /system.slice/docker.service
             └─5741 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Oct 24 17:01:21 Workstation dockerd[5741]: time="2025-10-24T17:01:21.782381467+>
Oct 24 17:01:22 Workstation dockerd[5741]: time="2025-10-24T17:01:22.371939484+>
Oct 24 17:01:22 Workstation dockerd[5741]: time="2025-10-24T17:01:22.420297673+>
Oct 24 17:01:22 Workstation dockerd[5741]: time="2025-10-24T17:01:22.421066584+>
Oct 24 17:01:22 Workstation dockerd[5741]: time="2025-10-24T17:01:22.433356405+>
```

In this number 2 the task is to install docker and enable and start is so i did the installing of docker and enable it and start it

3.

```
paul@Workstation:~/HOA11$ newgrp docker
paul@Workstation:~/HOA11$ docker run hello-world
Unable to find image 'hello-world:latest' locally
Latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:6dc565aa63092705211f823c303948cf83670a3903ffa3849f1488ab517f891
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:
In this task, I was able to add the docker into my current user.
4.
```

```

paul@Workstation:~/HOA11$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
web_db_server   latest   dbcfc995a5ee  23 minutes ago  564MB
<none>          <none>   b4212f22dd84  38 minutes ago  564MB
ubuntu          latest   97bed23a3497  3 weeks ago   78.1MB
hello-world     latest   1b44b5a3e06a  2 months ago   10.1kB
paul@Workstation:~/HOA11$ sudo docker tag b4212f22dd84 webdb-server:latest
paul@Workstation:~/HOA11$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
web_db_server   latest   dbcfc995a5ee  24 minutes ago  564MB
webdb-server    latest   b4212f22dd84  39 minutes ago  564MB
ubuntu          latest   97bed23a3497  3 weeks ago   78.1MB
hello-world     latest   1b44b5a3e06a  2 months ago   10.1kB
paul@Workstation:~/HOA11$ sudo docker run -it webdb-server
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.5. Set the 'ServerName' directive globally to suppress this message
 *
 * Starting MariaDB database server mariadb [ OK ]
root@8cc3b83713a1:#

```

In this i was able to create for web server and db server i use the command docker run to run it

5.

```

paul@Workstation:~/HOA11$ sudo nano dockerbuild
paul@Workstation:~/HOA11$ ansible-playbook dockerbuild -i inventory11.ini -K
BECOME password:

PLAY [Install Docker and build Docker image] ****
TASK [Gathering Facts] ****
ok: [192.168.56.123]
ok: [192.168.56.122]

TASK [Install required packages for Docker] ****
ok: [192.168.56.122]
ok: [192.168.56.123]

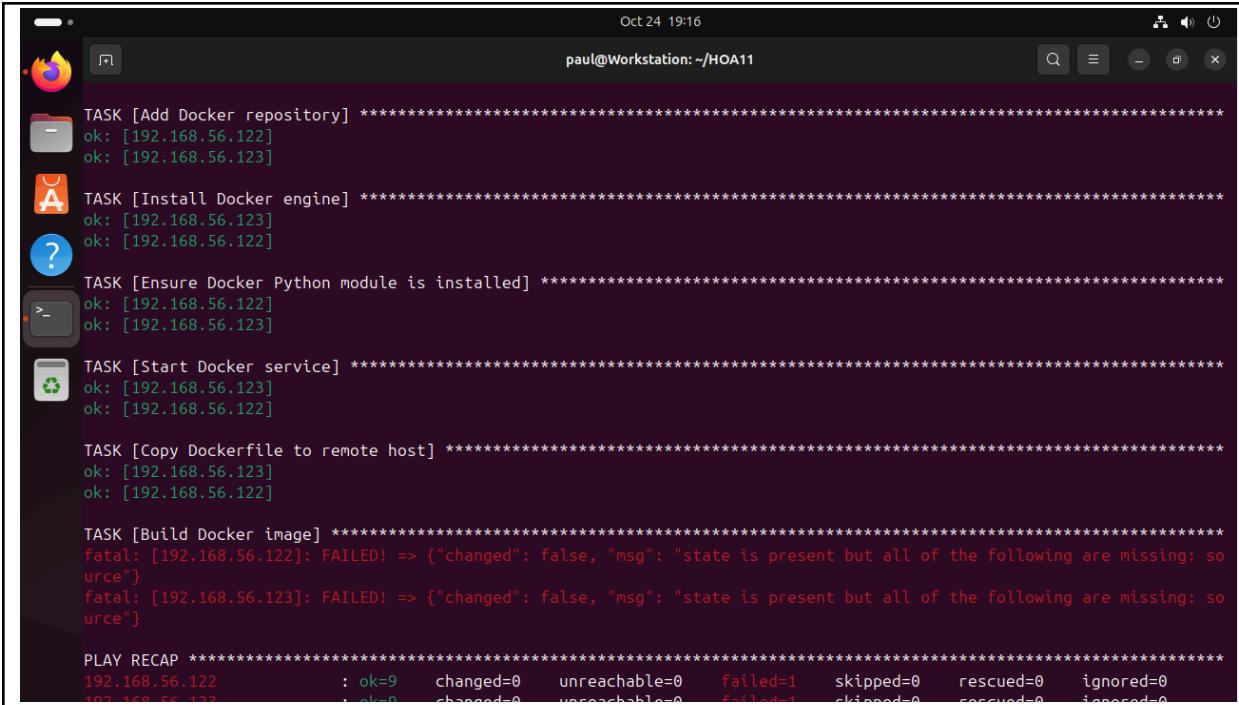
TASK [Remove PEP 668 restriction file] ****
ok: [192.168.56.122]
ok: [192.168.56.123]

TASK [Add Docker GPG key] ****
ok: [192.168.56.122]
ok: [192.168.56.123]

TASK [Add Docker repository] ****
ok: [192.168.56.122]
ok: [192.168.56.123]

TASK [Install Docker engine] ****
ok: [192.168.56.123]
ok: [192.168.56.122]

```



Oct 24 19:16
paul@Workstation: ~/HOA11

```
TASK [Add Docker repository] *****
ok: [192.168.56.122]
ok: [192.168.56.123]

TASK [Install Docker engine] *****
ok: [192.168.56.123]
ok: [192.168.56.122]

TASK [Ensure Docker Python module is installed] *****
ok: [192.168.56.122]
ok: [192.168.56.123]

TASK [Start Docker service] *****
ok: [192.168.56.123]
ok: [192.168.56.122]

TASK [Copy Dockerfile to remote host] *****
ok: [192.168.56.123]
ok: [192.168.56.122]

TASK [Build Docker image] *****
fatal: [192.168.56.122]: FAILED! => {"changed": false, "msg": "state is present but all of the following are missing: source"}
fatal: [192.168.56.123]: FAILED! => {"changed": false, "msg": "state is present but all of the following are missing: source"}

PLAY RECAP *****
192.168.56.122 : ok=9    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.123 : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

in here i was able to download and build the docker but the problem s that the build docker image is error and i dont know what was the problem

6.

```
paul@Workstation:~/HOA11$ git add .
paul@Workstation:~/HOA11$ git commit -m "HOA11"
[main 0f4b5b5] HOA11
 5 files changed, 91 insertions(+)
  create mode 100644 Dockerfile
  create mode 100644 ansible.cfg
  create mode 100644 dockerbuild
  create mode 100644 inventory11.ini
  create mode 100644 pip3
paul@Workstation:~/HOA11$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.49 KiB | 509.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Paul-Solis/HOA11.git
 bd4c349..0f4b5b5  main -> main
```

The screenshot shows a GitHub repository page for 'HOA11'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a header bar with the repository name 'HOA11' (Public), a 'Pin' button, a 'Watch' button (0), a 'Fork' button (0), and a 'Star' button (0). The main content area shows a list of files: Dockerfile, README.md, ansible.cfg, dockerbuild, inventory11.ini, and pip3. The README file is currently selected. On the right side, there's an 'About' section with a note: 'No description, website, or topics provided.' It also lists Readme, Activity, 0 stars, 0 watching, and 0 forks. Below that is a 'Releases' section with a note: 'No releases published' and a link to 'Create a new release'. There's also a 'Packages' section.

i uploaded it into my github

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization ensures applications run consistently across different environments by isolating them in lightweight, portable containers.

Conclusions:

In this laboratory activity, we explored the process of containerizing applications using Docker, focusing on setting up both a web server and a database server within a single container. By doing so, we demonstrated the benefits of containerization, such as consistency, portability, and efficient resource usage. Ultimately, this exercise highlighted how containerization streamlines development, testing, and deployment workflows.