Name: Bacong, El Cid A.	**Date Performed:** 24 October 2025
Course/Section: CPE212 - CPE31S4	**Date Submitted:** 24 October 2025
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

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)

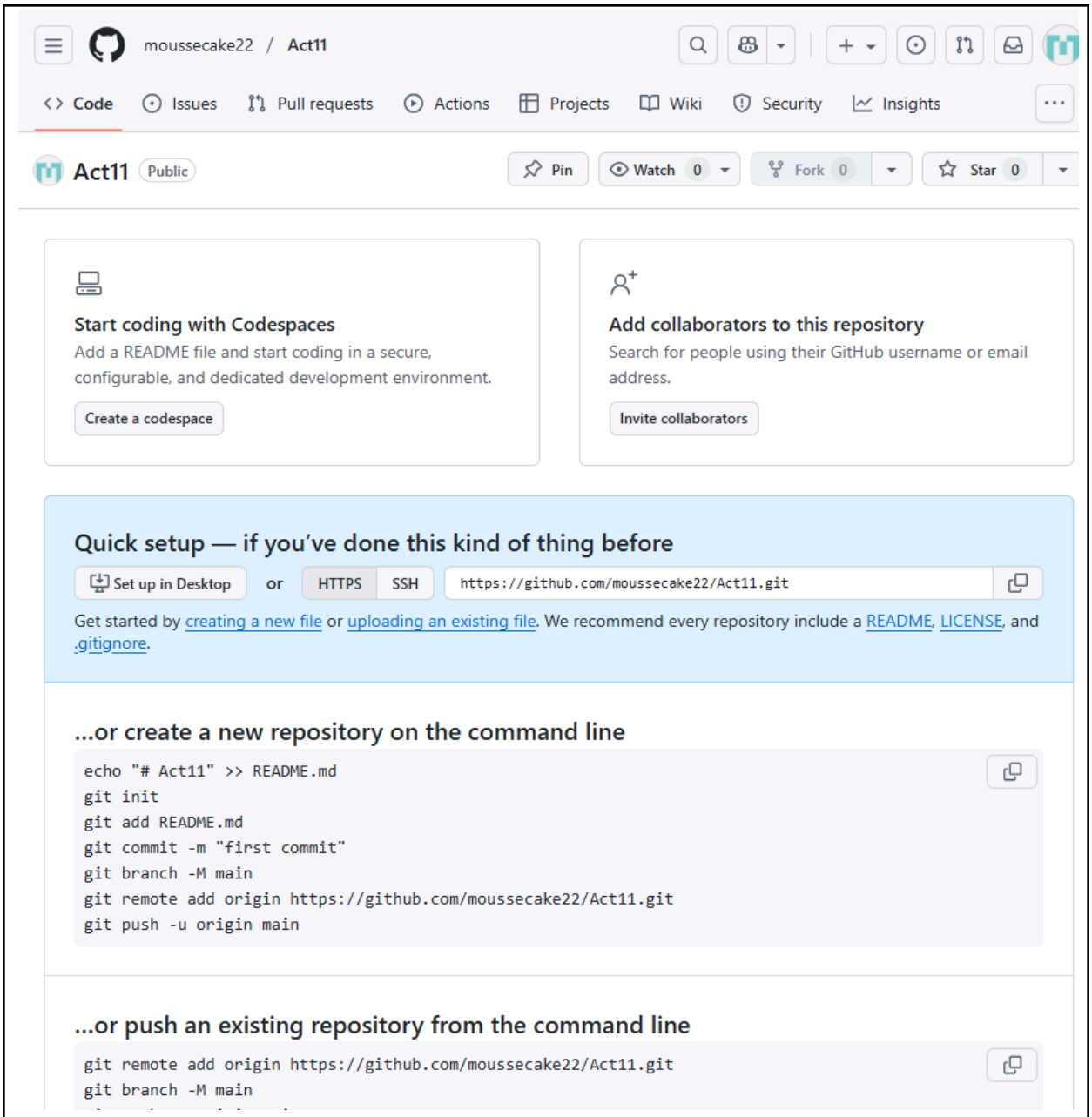

Figure 1: Creating a new repository

```
changed: [192.168.56.104]

TASK [Add Docker's official GPG key] ***************************************
changed: [192.168.56.104]

TASK [Add Docker repository] ***********************************************
changed: [192.168.56.104]

TASK [Install Docker Engine] ***********************************************

changed: [192.168.56.104]

TASK [Install python3-docker via apt] **************************************
changed: [192.168.56.104]

TASK [Ensure Docker service is started and enabled on boot] ****************
ok: [192.168.56.104]

TASK [Add user to docker group] ********************************************
ok: [192.168.56.104]

PLAY RECAP *********************************************************************
192.168.56.104             : ok=9    changed=6    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

```
cidee@workstation:~/Act11$ docker --version
Docker version 24.0.2, build cb74dfc
```

Figure 2: Installing Docker and Adding user to docker group

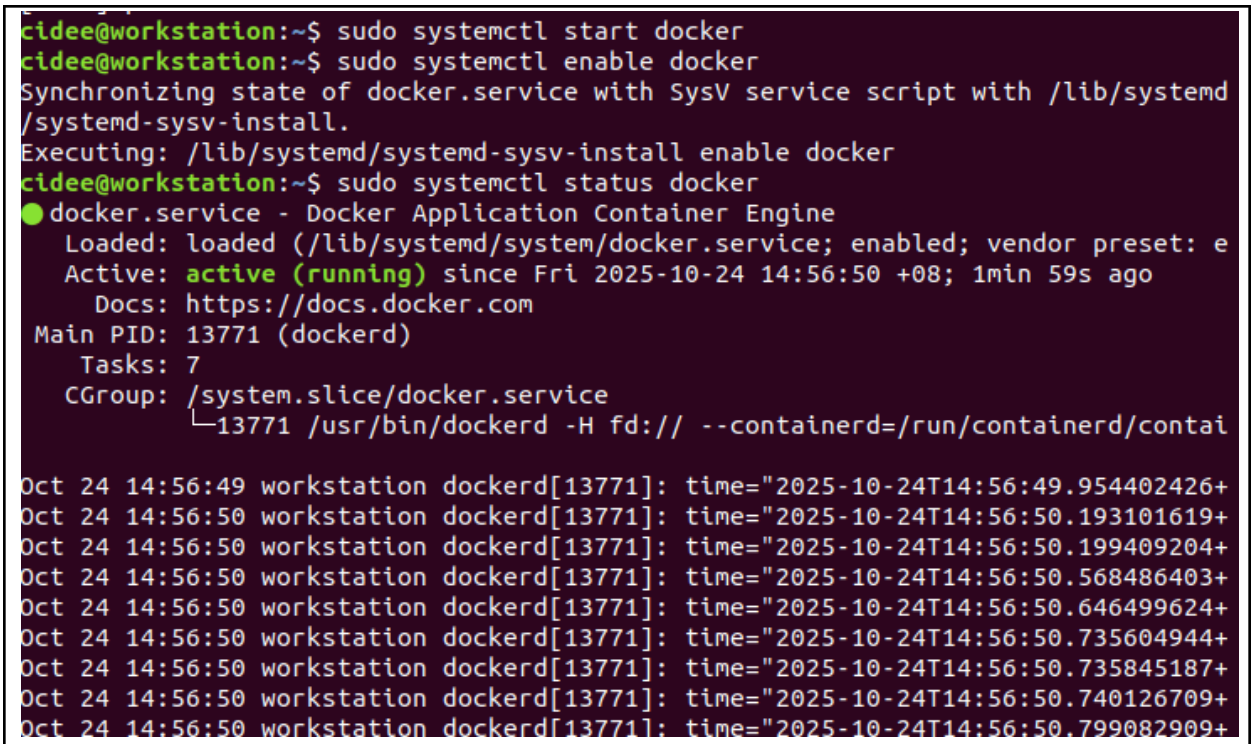

Figure 3: Enabling Docker Socket

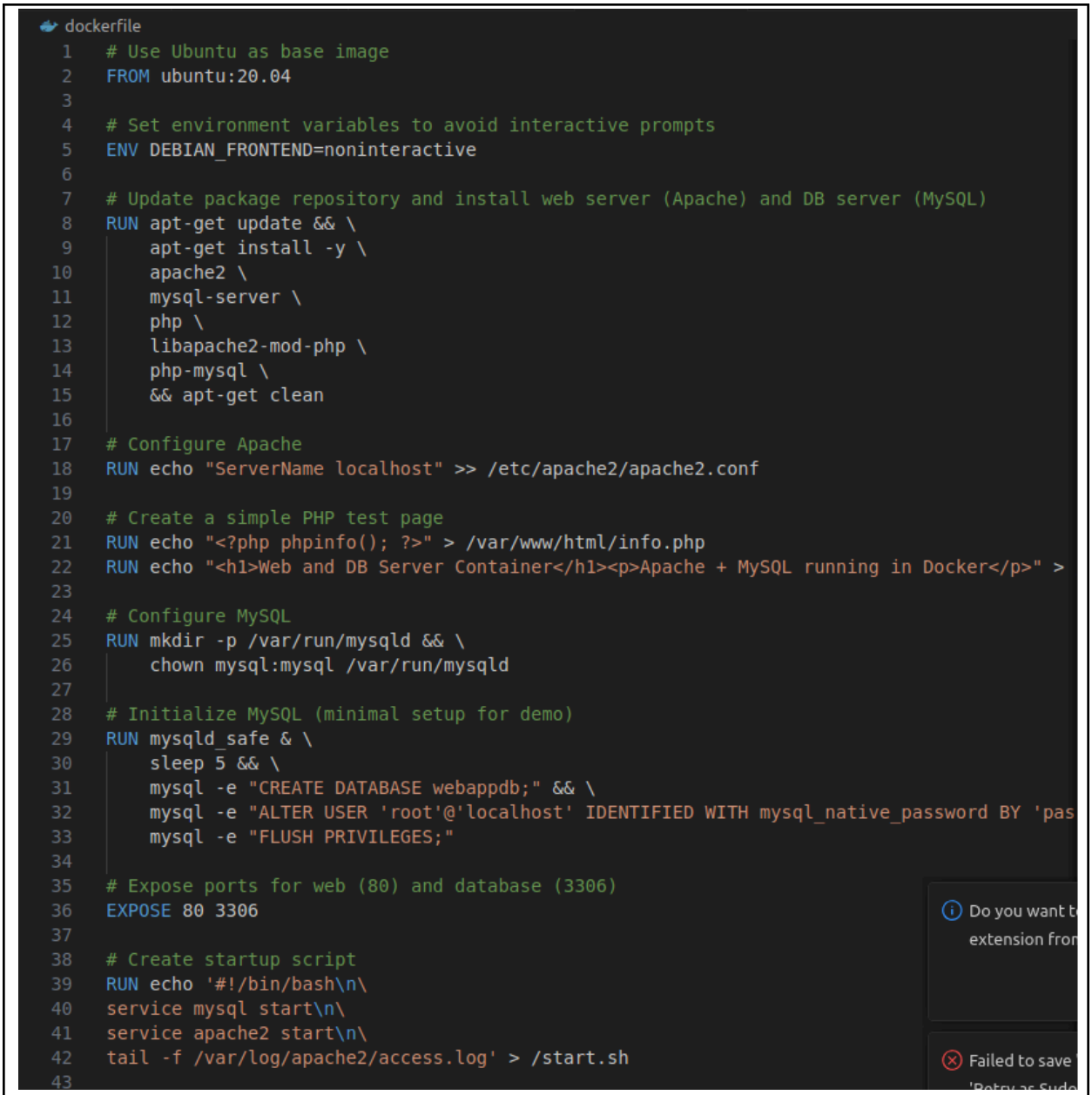

Figure 4: Creating Dockerfile for Web and DB Server

build-docker-image.yml

```yaml
---
- name: Build custom Docker image with Web and DB server
  hosts: all
  become: yes

  tasks:
    - name: Copy Dockerfile to remote host
      copy:
        src: Dockerfile
        dest: /tmp/Dockerfile

    - name: Build Docker image from Dockerfile
      command: docker build -t web-db-server /tmp/
      args:
        chdir: /tmp

    - name: Run the web-db container
      command: |
        docker run -d \
          --name web-db-container \
          -p 8080:80 \
          -p 3306:3306 \
          web-db-server

    - name: Verify container is running
      command: docker ps
      register: container_list

    - name: Display running containers
      debug:
        var: container_list.stdout

```

Figure 5: Install and Build Dockerfile using Ansible

```
TASK [Update APT package cache] ************************************************
changed: [192.168.56.104]

TASK [Install prerequisite packages] *******************************************
ok: [192.168.56.104]

TASK [Add Docker's official GPG key] *******************************************
ok: [192.168.56.104]

TASK [Add Docker repository] ***************************************************
ok: [192.168.56.104]

TASK [Install Docker Engine] ***************************************************
ok: [192.168.56.104]

TASK [Ensure Docker service is started and enabled] ****************************
ok: [192.168.56.104]

TASK [Add user to docker group] ************************************************
ok: [192.168.56.104]
```

Figure 6: Execute Workflow

```
TASK [Check Docker containers] *************************************************
changed: [192.168.56.104]

TASK [Display container status] ************************************************
ok: [192.168.56.104] => {
    "containers.stdout": "CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS
 PORTS     NAMES"
}

TASK [Check Docker images] *****************************************************
changed: [192.168.56.104]

TASK [Display available images] ************************************************
ok: [192.168.56.104] => {
    "images.stdout": "REPOSITORY   TAG       IMAGE ID   CREATED   SIZE"
}
```

Figure 7: Execute Deployment

```
cidee@workstation:~/Docker_Act11$ git add *
cidee@workstation:~/Docker_Act11$ git commit -m Docker
[main fc266f9] Docker
 9 files changed, 302 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 build-docker-image.yml
 create mode 100644 complete-workflow.yml
 create mode 100644 docker-install.yml
 create mode 100644 dockerfile
 create mode 100644 index.html
 create mode 100644 inventory.ini
 create mode 100644 nginx-deploy.yml
 create mode 100644 verify-deployment.yml
cidee@workstation:~/Docker_Act11$ git push origin main
Counting objects: 11, done.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 3.40 KiB | 1.70 MiB/s, done.
Total 11 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:moussecake22/Docker_Act11.git
   4fe30ff..fc266f9  main -> main
```

Figure 8: Saving my work on the New Repository I created

GitHub link: https://github.com/moussecake22/Docker_Act11

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   Containerization makes it easier for developers to run apps anywhere without problems. It keeps everything the app needs in one package, so it works the same on any computer or server. Containers are lightweight and fast, which makes testing and deployment quicker. They also keep apps separated for better security and fewer errors. Overall, containerization helps developers save time, avoid setup issues, and makes the whole process of building and updating apps smoother.

Conclusions:

In this activity, I learned how Docker makes app development faster and easier by packaging everything an app needs to run anywhere. Using Docker and Ansible helped me understand how automation and containerization support continuous delivery and make deploying updates more efficient.