

<b>Name:</b> Hazel Aillson T. Manuel	<b>Date Performed:</b> 10/24/2025
<b>Course/Section:</b> CPE212 CPE31S4	<b>Date Submitted:</b> 10/24/2025
<b>Instructor:</b> Engr. Robin Valenzuela	<b>Semester and SY:</b> 1st Sem '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)

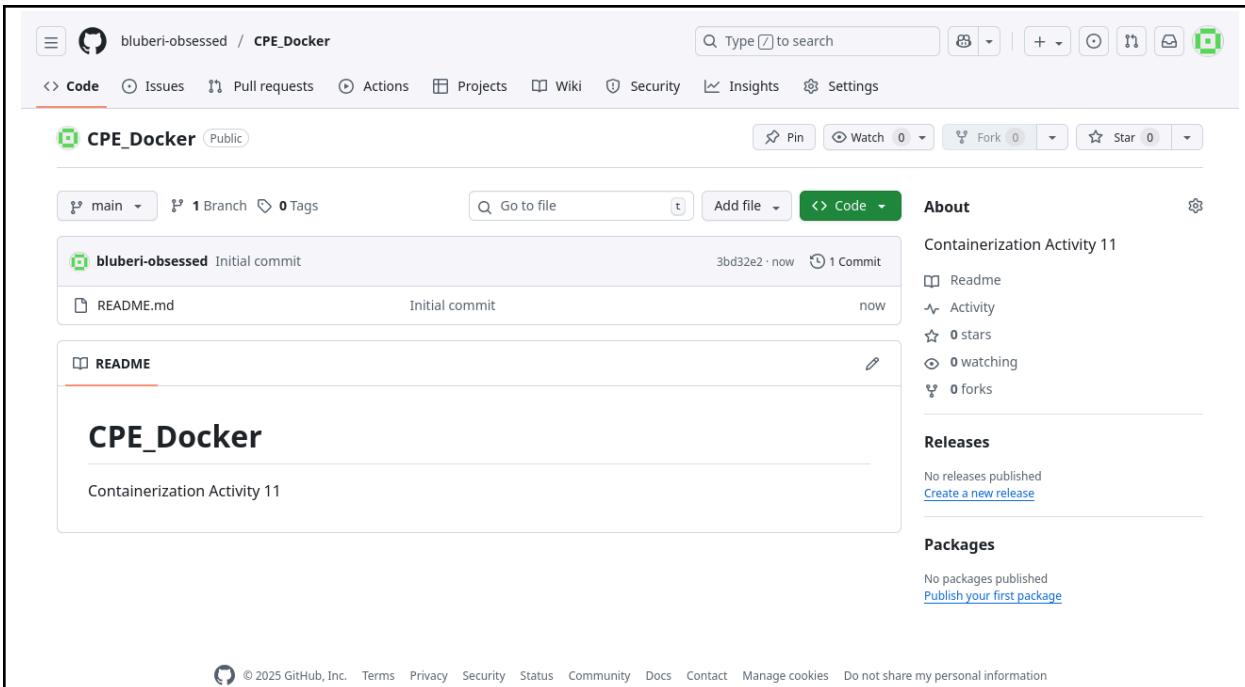


Figure 1: Creating a New Repository in Github

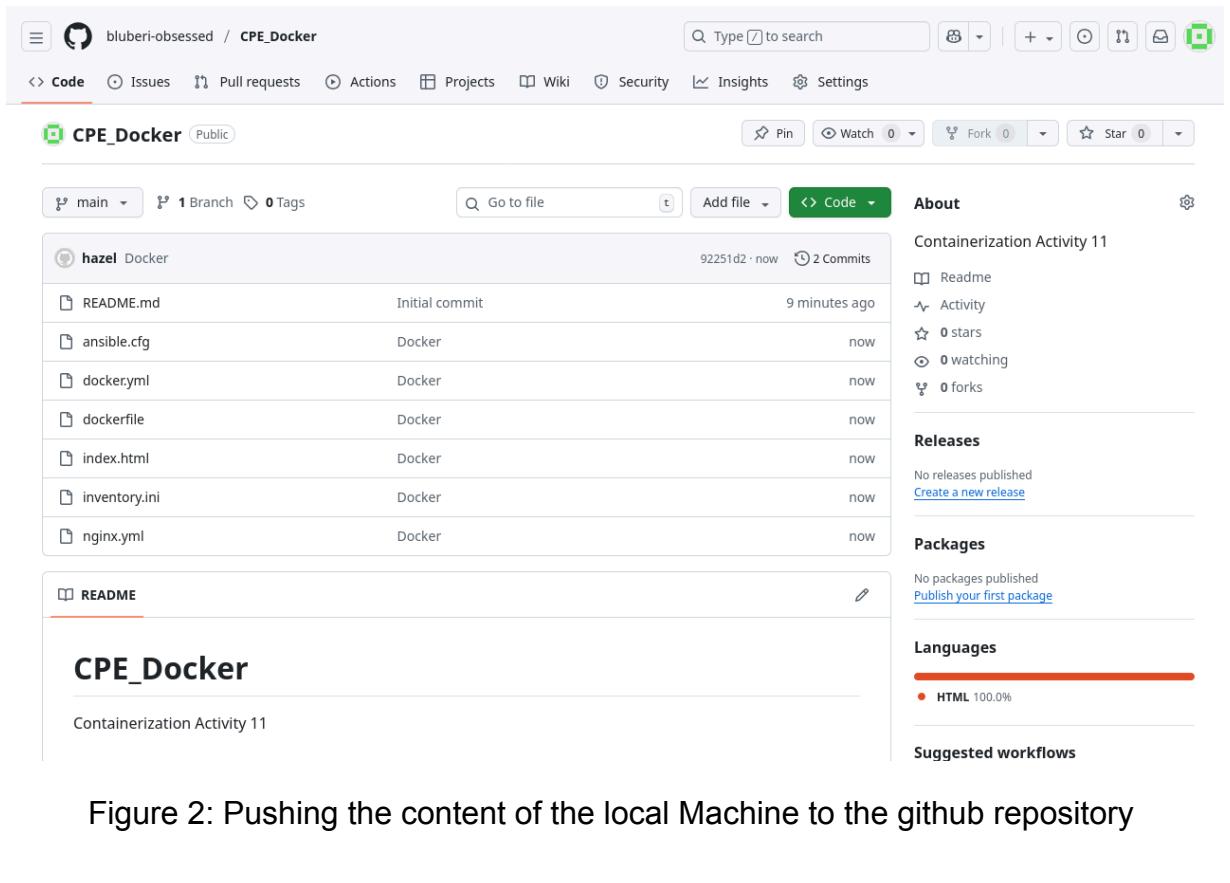


Figure 2: Pushing the content of the local Machine to the github repository

```
Shell
---
- name: Install Docker on all nodes
  hosts: all
  become: yes

  tasks:
    - name: Update APT package cache
      apt:
        update_cache: yes

    - name: Install prerequisite packages
      apt:
        name:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
        state: present

    - name: Add Docker's official GPG key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker repository
      apt_repository:
        repo: "deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ ansible_distribution_release }} stable"
        state: present

    - name: Install Docker Engine
      apt:
        name: docker-ce
        state: present
        update_cache: yes

    - name: Install python3-docker via apt
      apt:
        name: python3-docker
        state: present

    - name: Ensure Docker service is started and enabled on boot
      service:
        name: docker
        state: started
        enabled: yes
```

Figure 3: Installing and Setting up Docker

```
See 'snap info <snapname>' for additional versions.  
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ docker --version  
Docker version 28.5.1, build e180ab8
```

Figure 3: Verifying if Docker is installed in the local machine

```
Docker version 28.5.1, build e180ab8  
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ groups $USER  
hazel : hazel sudo docker
```

Figure 4: Adding the current user to the group called docker

Shell

```
# Allows it to use the official Ubuntu base image  
FROM ubuntu:20.04  
  
# This allows the system to update the packages in the system and also  
ensures that Apache2 and MySQL is installed  
RUN apt-get update && \  
    DEBIAN_FRONTEND=noninteractive apt-get install -y \ # Prevents prompts  
    apache2 \  
    mysql-server && \  
    apt-get clean  
  
# 80 for http and 3306 for MySQL  
EXPOSE 80 3306  
  
# Starts the services of Apache and MySQL  
CMD service apache2 start && service mysql start && bash
```

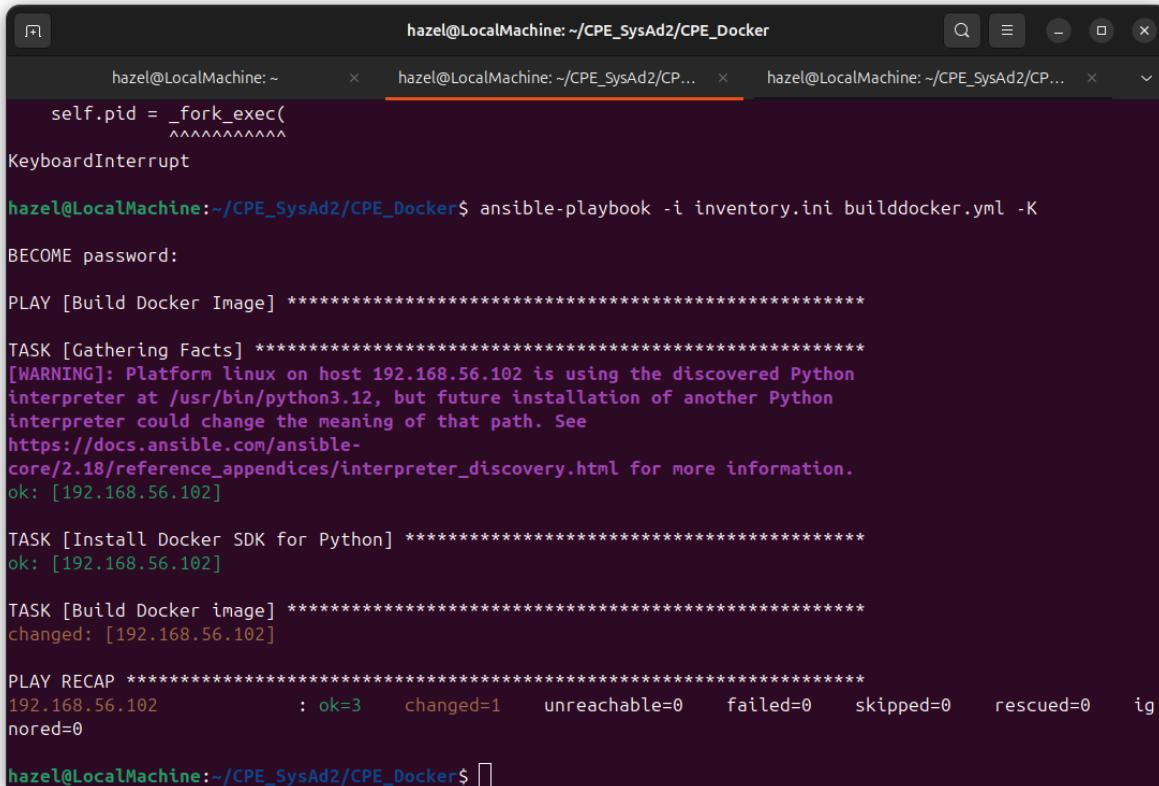
Figure 5: Creating a Docker File to install web and database server

Shell

```
- name: Build Docker Image  
  hosts: localhost  
  become: true  
  tasks:  
  
    - name: Install Docker SDK for Python  
      apt:  
        name: python3-docker  
        state: present  
  
    - name: Build Docker image  
      community.docker.docker_image:  
        name: web_db_image  
        source: build
```

```
build:  
  path: /home/hazel/CPE_SysAd2/CPE_Docker
```

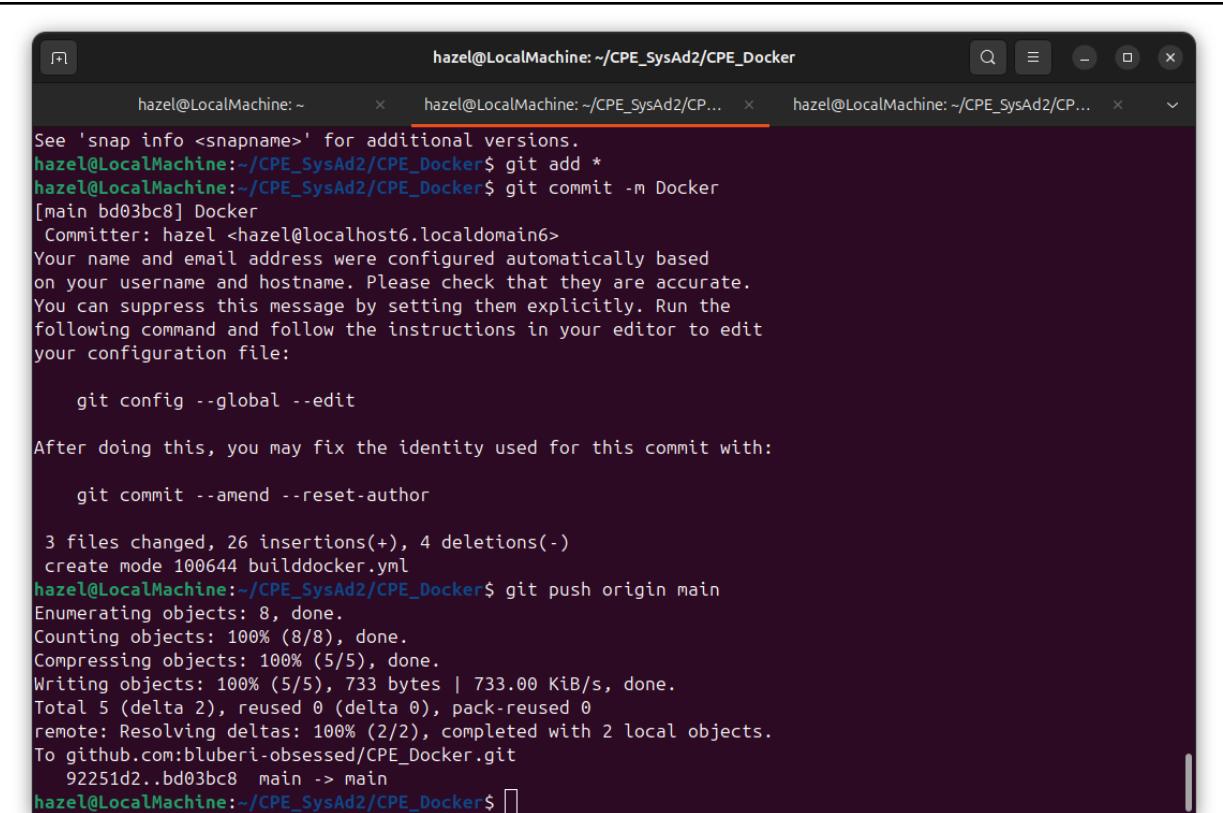
Figure 5: Playbook that installs the Docker SDK and Building the Docker Image



The screenshot shows a terminal window with three tabs open, all titled "hazel@LocalMachine: ~/CPE\_SysAd2/CPE\_Docker". The central tab is active and displays the output of an Ansible playbook run. The output shows:

```
hazel@LocalMachine: ~      x  hazel@LocalMachine: ~/CPE_SysAd2/CPE_Docker  x  hazel@LocalMachine: ~/CPE_SysAd2/CPE_Docker  x    
self.pid = _fork_exec(  
    ^^^^^^^^^^  
KeyboardInterrupt  
  
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ ansible-playbook -i inventory.ini builddocker.yml -K  
BECOME password:  
  
PLAY [Build Docker Image] *****  
  
TASK [Gathering Facts] *****  
[WARNING]: Platform linux on host 192.168.56.102 is using the discovered Python  
interpreter at /usr/bin/python3.12, but future installation of another Python  
interpreter could change the meaning of that path. See  
https://docs.ansible.com/ansible-  
core/2.18/reference_appendices/interpreter_discovery.html for more information.  
ok: [192.168.56.102]  
  
TASK [Install Docker SDK for Python] *****  
ok: [192.168.56.102]  
  
TASK [Build Docker image] *****  
changed: [192.168.56.102]  
  
PLAY RECAP *****  
192.168.56.102 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ig  
nored=0  
  
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ 
```

Figure 7: Running the playbook that installs the Docker SDK and Build its image



The screenshot shows a terminal window with three tabs open, all titled "hazel@LocalMachine: ~/CPE\_SysAd2/CPE\_Docker". The current tab displays the following command-line session:

```
See 'snap info <snapname>' for additional versions.
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ git add *
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ git commit -m Docker
[main bd03bc8] Docker
  Committer: hazel <hazel@localhost6.localdomain6>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

3 files changed, 26 insertions(+), 4 deletions(-)
create mode 100644 builddocker.yml
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 733 bytes | 733.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:bluberi-obsessed/CPE_Docker.git
  92251d2..bd03bc8  main -> main
hazel@LocalMachine:~/CPE_SysAd2/CPE_Docker$
```

Figure 7: Syncing what I have done to the local machine's directory to my github repository

GitHub Repository: [https://github.com/bluberi-obsessed/CPE\\_Docker.git](https://github.com/bluberi-obsessed/CPE_Docker.git)

### Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization is the process of bundling an application, along with its dependencies. Having this feature actually allows us to run these containers across different environments which makes it consistent whenever the environment you are in. This also makes it so that it is in an isolated environment which improves its security as it limits the access between services.

### Conclusions:

In this activity, docker was introduced to all of us. We were taught how to install the docker and configure it so that it can run properly or run as intended in our machine. Furthermore, we also installed its SDK and built its image. What we actually did is to

set up an environment, maybe manually or automatically. Of course, there have been issues such as unable to run the playbook as I intended so I had to do a lot of debugging to the playbook and rechecking my dockerfile to ensure that it is correct. Overall, this has been a good foundation for us to further learn about Docker