Name: Ruud Van G. Apostol	**Date Performed:** 10/23/25
Course/Section: CPE212 / CPE31S4	**Date Submitted:** 10/23/25
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

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.

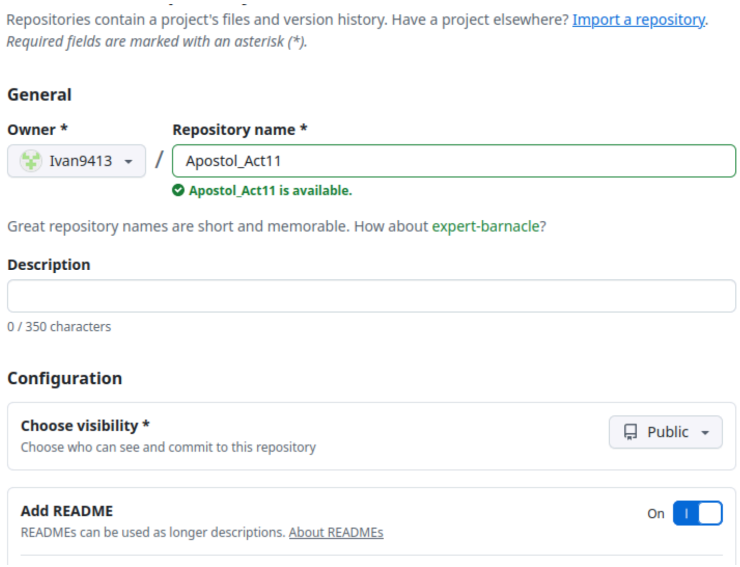
Source: https://docs.docker.com/get-started/overview/

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.



Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).

1 General

Owner *: Ivan9413 / Repository name *: Apostol_Act11

Apostol_Act11 is available.

Great repository names are short and memorable. How about expert-barnacle?

Description

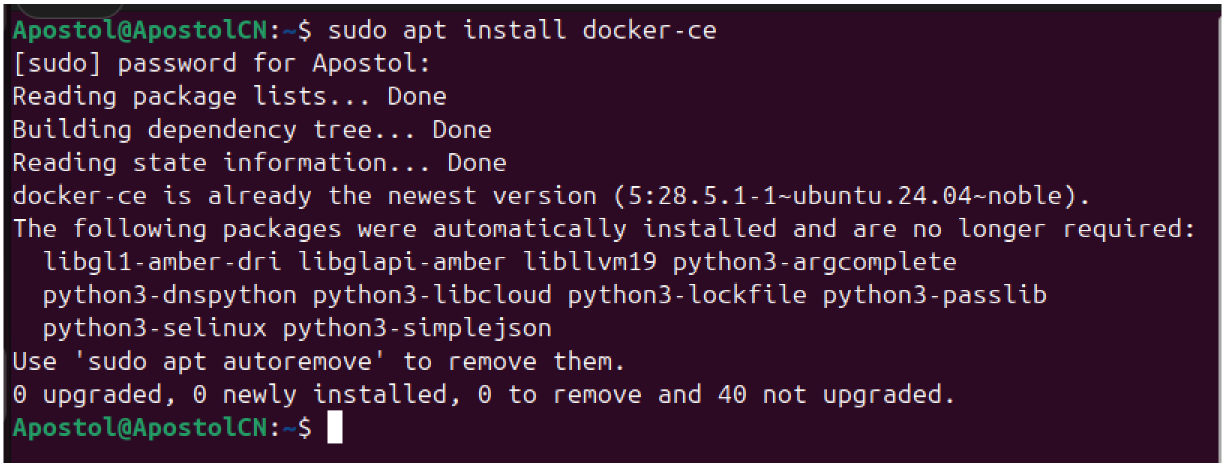
0 / 350 characters

2 Configuration

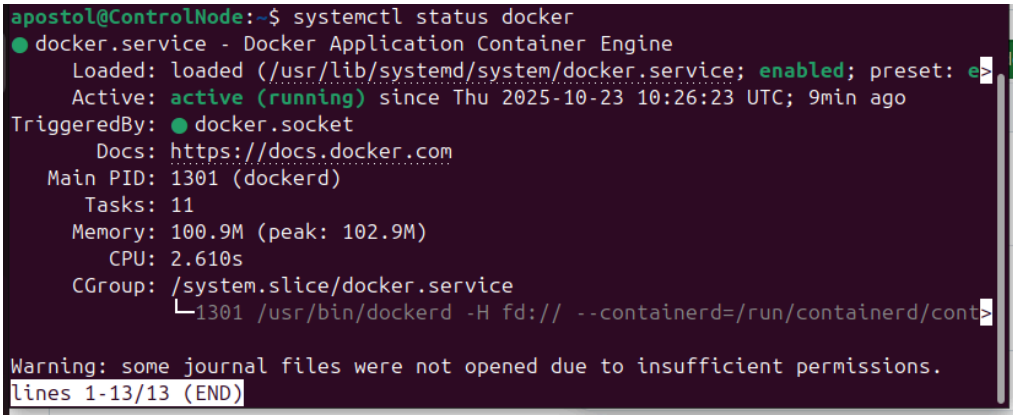
Choose visibility *: Public
Choose who can see and commit to this repository

Add README: On
READMEs can be used as longer descriptions. About READMEs

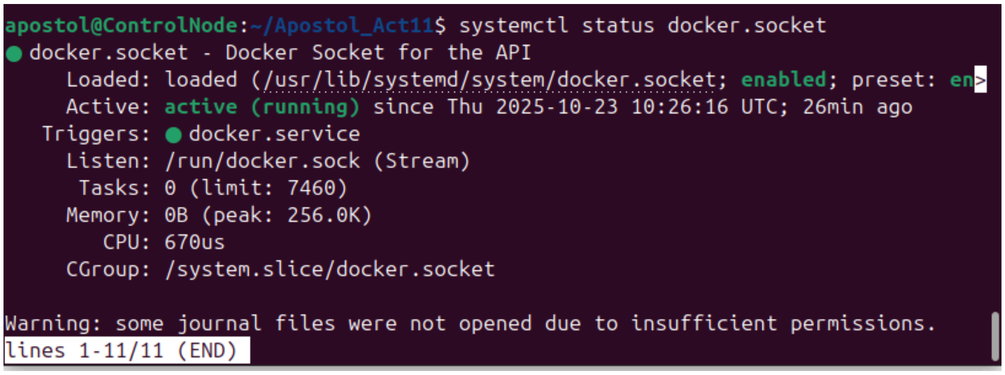
2. Install Docker and enable the docker socket.



```
Apostol@ApostolCN:~$ sudo apt install docker-ce
[sudo] password for Apostol:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-ce is already the newest version (5:28.5.1-1~ubuntu.24.04~noble).
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-amber libllvm19 python3-argcomplete
  python3-dnspython python3-libcloud python3-lockfile python3-passlib
  python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
Apostol@ApostolCN:~$
```

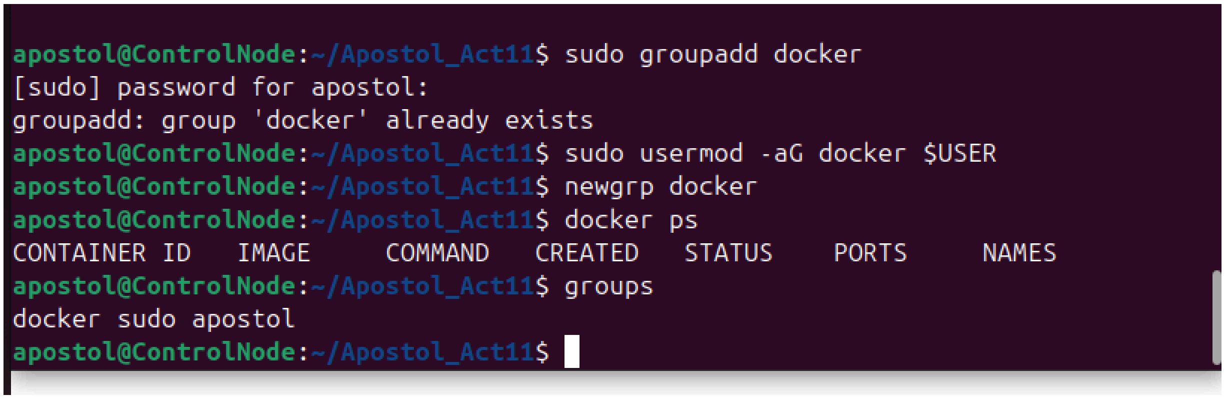


```
apostol@ControlNode:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: e>
     Active: active (running) since Thu 2025-10-23 10:26:23 UTC; 9min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1301 (dockerd)
      Tasks: 11
     Memory: 100.9M (peak: 102.9M)
        CPU: 2.610s
     CGroup: /system.slice/docker.service
             └─1301 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Warning: some journal files were not opened due to insufficient permissions.
lines 1-13/13 (END)
```



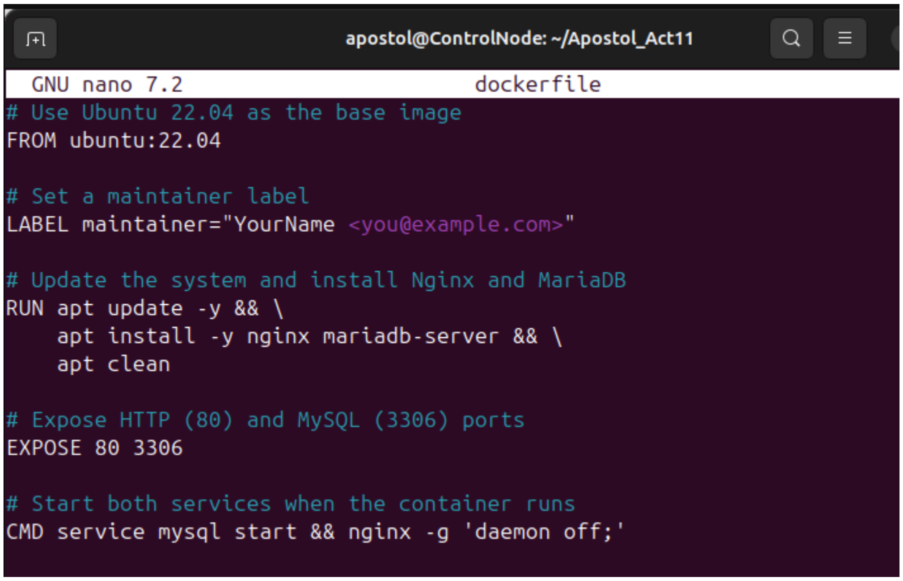
```
apostol@ControlNode:~/Apostol_Act11$ systemctl status docker.socket
● docker.socket - Docker Socket for the API
     Loaded: loaded (/usr/lib/systemd/system/docker.socket; enabled; preset: en>
     Active: active (running) since Thu 2025-10-23 10:26:16 UTC; 26min ago
   Triggers: ● docker.service
     Listen: /run/docker.sock (Stream)
      Tasks: 0 (limit: 7460)
     Memory: 0B (peak: 256.0K)
        CPU: 670us
     CGroup: /system.slice/docker.socket

Warning: some journal files were not opened due to insufficient permissions.
lines 1-11/11 (END)
```

3. Add to Docker group to your current user.

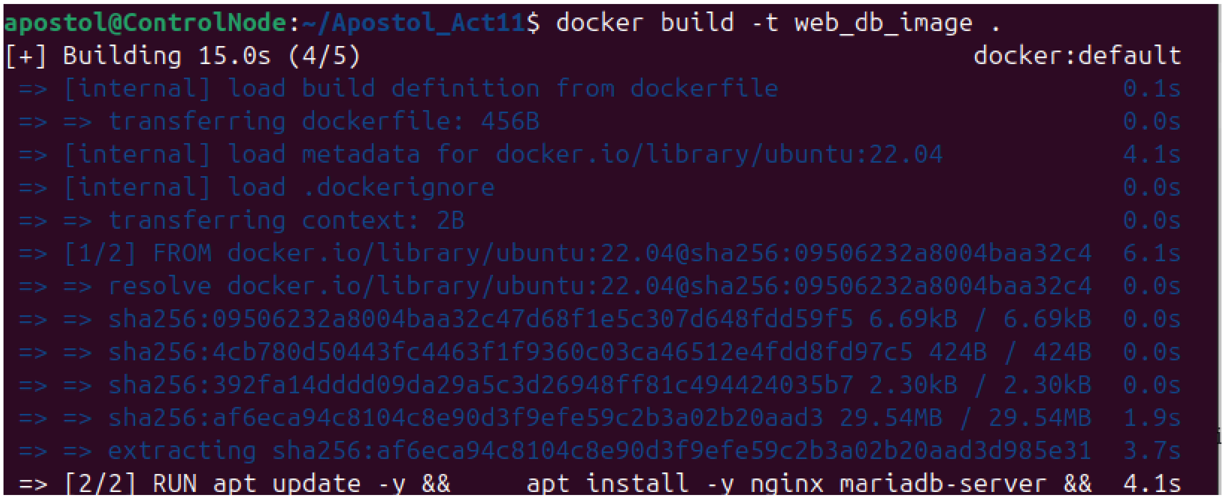


```
apostol@ControlNode:~/Apostol_Act11$ sudo groupadd docker
[sudo] password for apostol:
groupadd: group 'docker' already exists
apostol@ControlNode:~/Apostol_Act11$ sudo usermod -aG docker $USER
apostol@ControlNode:~/Apostol_Act11$ newgrp docker
apostol@ControlNode:~/Apostol_Act11$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
apostol@ControlNode:~/Apostol_Act11$ groups
docker sudo apostol
apostol@ControlNode:~/Apostol_Act11$
```

4. Create a Dockerfile to install web and DB server.

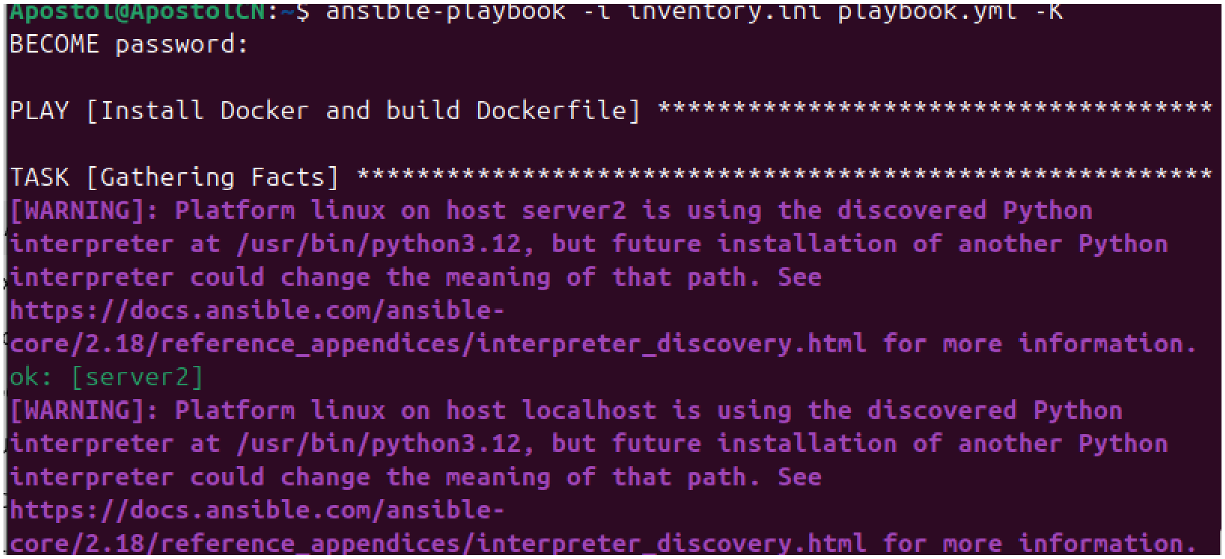


```
  GNU nano 7.2                         dockerfile
# Use Ubuntu 22.04 as the base image
FROM ubuntu:22.04

# Set a maintainer label
LABEL maintainer="YourName <you@example.com>"

# Update the system and install Nginx and MariaDB
RUN apt update -y && \
    apt install -y nginx mariadb-server && \
    apt clean

# Expose HTTP (80) and MySQL (3306) ports
EXPOSE 80 3306

# Start both services when the container runs
CMD service mysql start && nginx -g 'daemon off;'
```

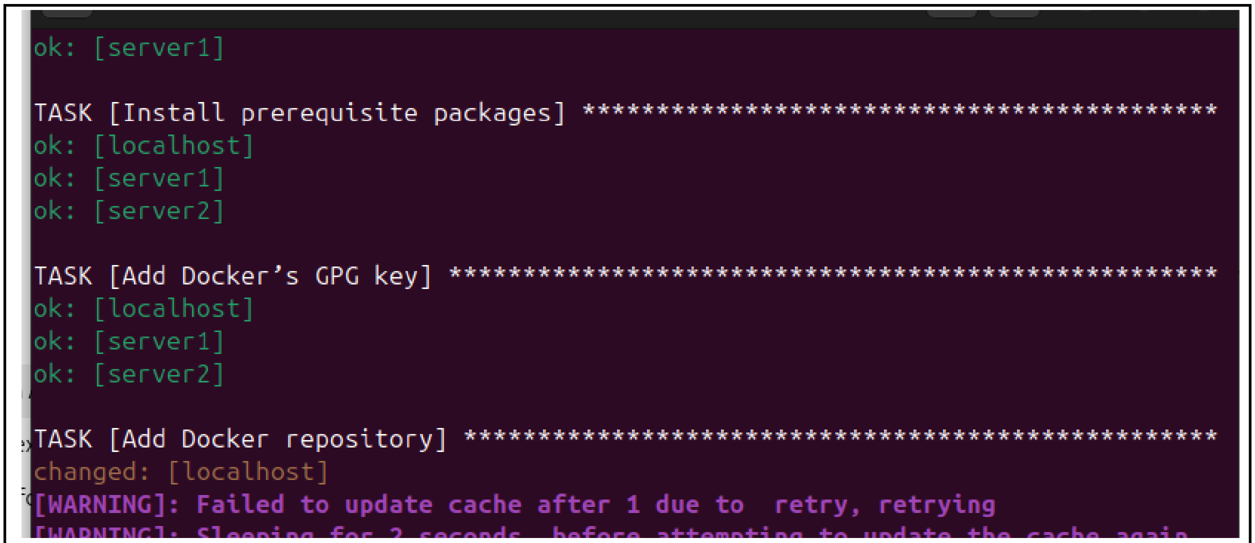


```
apostol@ControlNode:~/Apostol_Act11$ docker build -t web_db_image .
[+] Building 15.0s (4/5)                                          docker:default
 => [internal] load build definition from dockerfile                        0.1s
 => => transferring dockerfile: 456B                                        0.0s
 => [internal] load metadata for docker.io/library/ubuntu:22.04             4.1s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                             0.0s
 => [1/2] FROM docker.io/library/ubuntu:22.04@sha256:09506232a8004baa32c4   6.1s
 => => resolve docker.io/library/ubuntu:22.04@sha256:09506232a8004baa32c4   0.0s
 => => sha256:09506232a8004baa32c47d68f1e5c307d648fdd59f5 6.69kB / 6.69kB   0.0s
 => => sha256:4cb780d50443fc4463f1f9360c03ca46512e4fdd8fd97c5 424B / 424B   0.0s
 => => sha256:392fa14dddd09da29a5c3d26948ff81c494424035b7 2.30kB / 2.30kB   0.0s
 => => sha256:af6eca94c8104c8e90d3f9efe59c2b3a02b20aad3 29.54MB / 29.54MB   1.9s
 => => extracting sha256:af6eca94c8104c8e90d3f9efe59c2b3a02b20aad3d985e31   3.7s
 => [2/2] RUN apt update -y &&     apt install -y nginx mariadb-server &&   4.1s
```

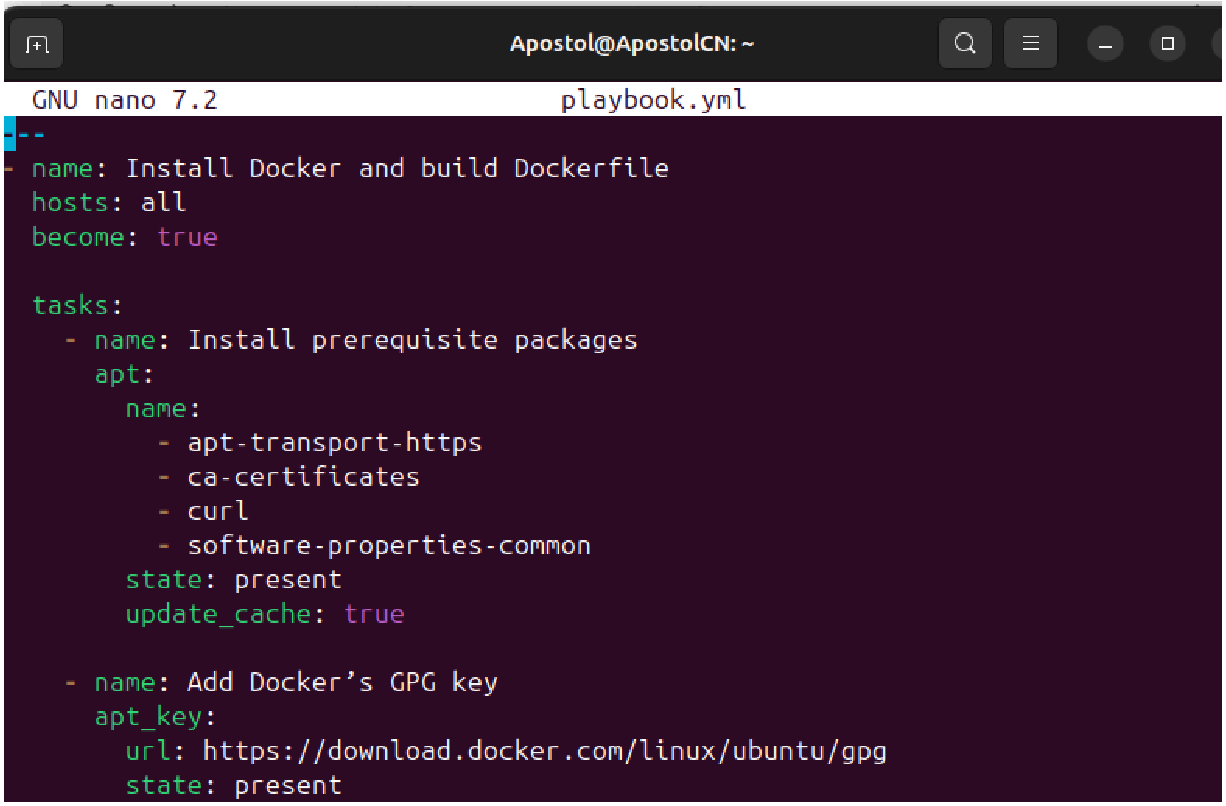
5. Install and build the Dockerfile using Ansible.



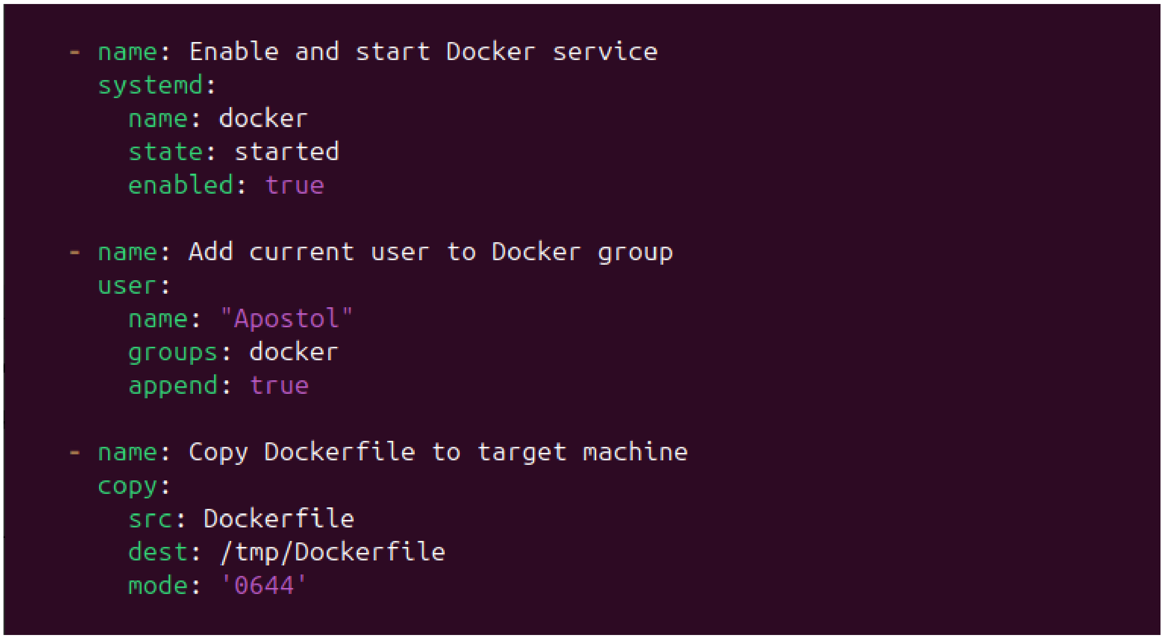
```
Apostol@ApostolCN:~$ ansible-playbook -i inventory.ini playbook.yml -K
BECOME password:

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

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host server2 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: [server2]
[WARNING]: Platform linux on host localhost 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: [server1]

TASK [Install prerequisite packages] *******************************************
ok: [localhost]
ok: [server1]
ok: [server2]

TASK [Add Docker’s GPG key] ****************************************************
ok: [localhost]
ok: [server1]
ok: [server2]

TASK [Add Docker repository] ***************************************************
changed: [localhost]
[WARNING]: Failed to update cache after 1 due to  retry, retrying
[WARNING]: Sleeping for 2 seconds, before attempting to update the cache again
```



```
  GNU nano 7.2                          playbook.yml
---
- name: Install Docker and build Dockerfile
  hosts: all
  become: true

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

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



```
    - name: Enable and start Docker service
      systemd:
        name: docker
        state: started
        enabled: true

    - name: Add current user to Docker group
      user:
        name: "Apostol"
        groups: docker
        append: true

    - name: Copy Dockerfile to target machine
      copy:
        src: Dockerfile
        dest: /tmp/Dockerfile
        mode: '0644'
```

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

4. Output

Reflections:

Answer the following:

1. **What are the benefits of implementing containerizations?**
   - Implementing containerization provides benefits such as portability, scalability, and efficiency, which lead to faster deployments, better resource utilization, and easier management across different environments like laptops, on-premises servers, or the cloud.

Conclusions:

- I learned how to setup and install docker in ubuntu and how to use ansible with docker. I also install nginx.