Name: Jasmin Raiza Tendencia	**Date Performed:** 11/13/2023
Course/Section: CPE232-CPE31S4	**Date Submitted:** 11/13/2023
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st sem/2023-2024

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.

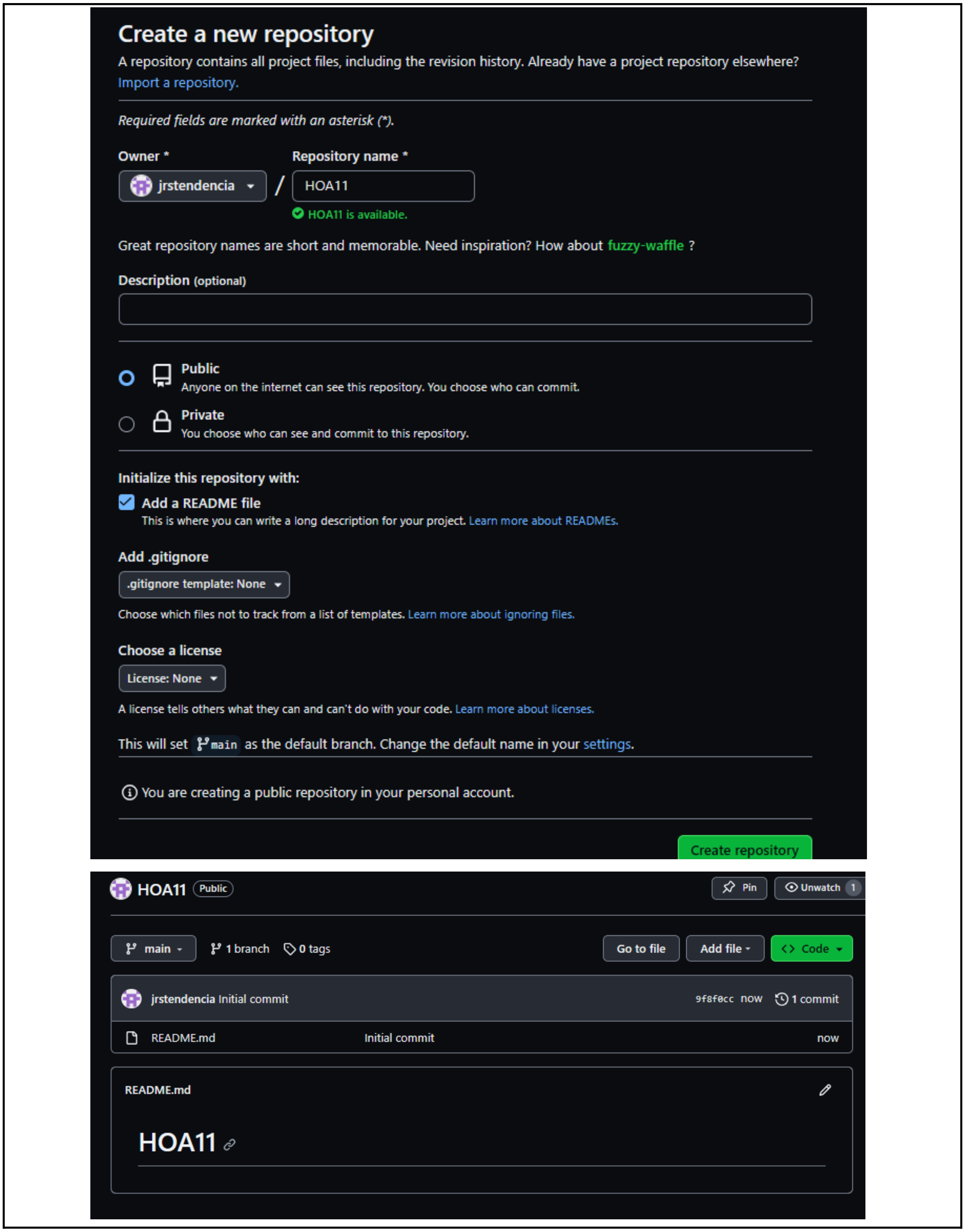
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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. Create a new repository for this activity.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: jrstendencia / Repository name *: HOA11

✓ HOA11 is available.

Great repository names are short and memorable. Need inspiration? How about fuzzy-waffle ?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☑ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

ⓘ You are creating a public repository in your personal account.

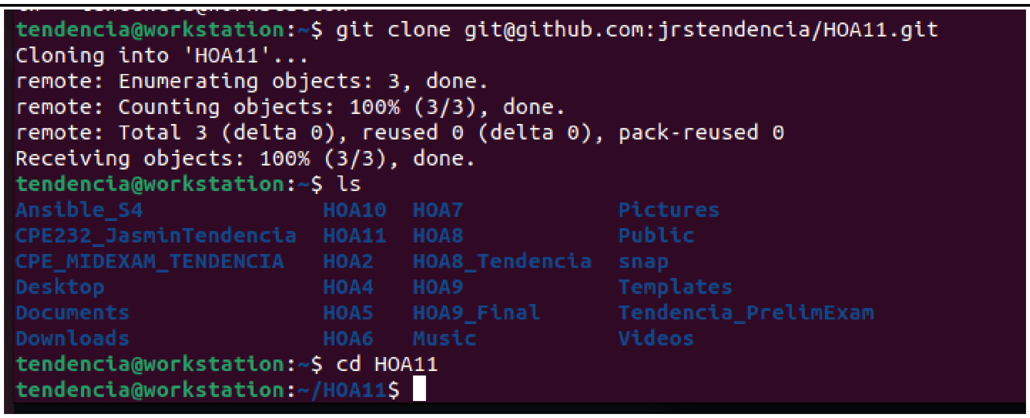
HOA11 Public

jrstendencia Initial commit — 9f8f0cc now — 1 commit

README.md — Initial commit — now

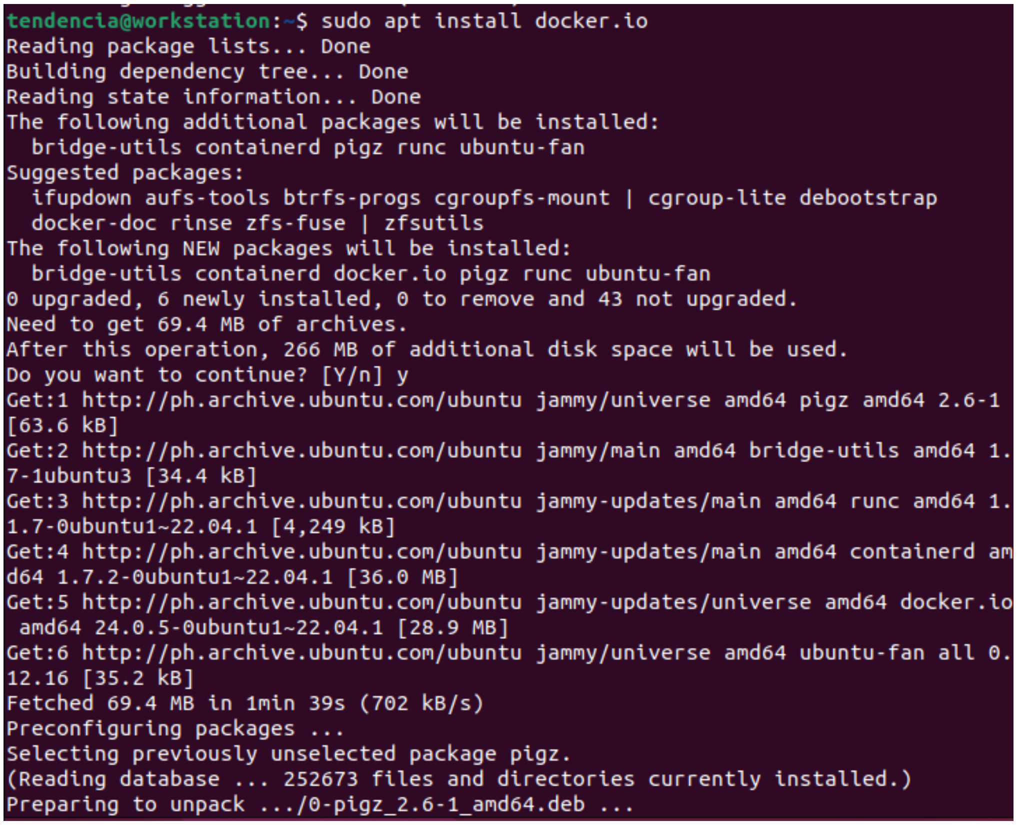
README.md

HOA11

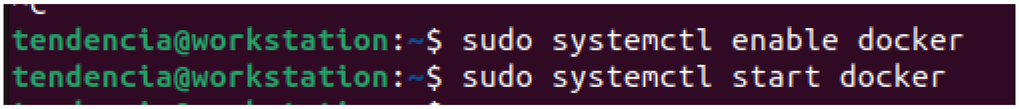


```
tendencia@workstation:~$ git clone git@github.com:jrstendencia/HOA11.git
Cloning into 'HOA11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
tendencia@workstation:~$ ls
Ansible_S4               HOA10  HOA7            Pictures
CPE232_JasminTendencia   HOA11  HOA8            Public
CPE_MIDEXAM_TENDENCIA    HOA2   HOA8_Tendencia  snap
Desktop                  HOA4   HOA9            Templates
Documents                HOA5   HOA9_Final      Tendencia_PrelimExam
Downloads                HOA6   Music           Videos
tendencia@workstation:~$ cd HOA11
tendencia@workstation:~/HOA11$ 
```

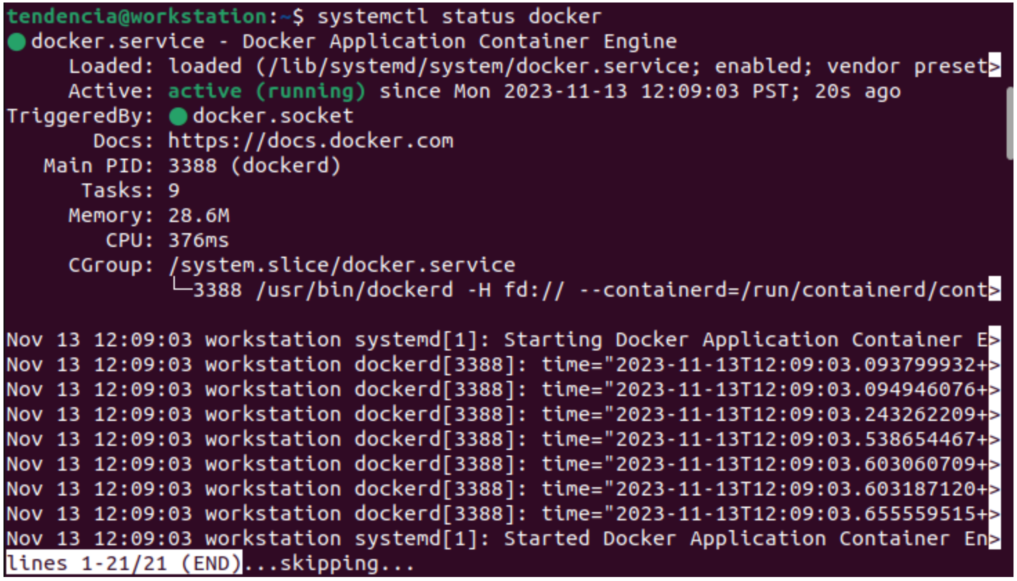
2. Install Docker and enable the docker socket.



```
tendencia@workstation:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
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-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 43 not upgraded.
Need to get 69.4 MB of archives.
After this operation, 266 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1
[63.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.
7-1ubuntu3 [34.4 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.
1.7-0ubuntu1~22.04.1 [4,249 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd am
d64 1.7.2-0ubuntu1~22.04.1 [36.0 MB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io
 amd64 24.0.5-0ubuntu1~22.04.1 [28.9 MB]
Get:6 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.
12.16 [35.2 kB]
Fetched 69.4 MB in 1min 39s (702 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 252673 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
```

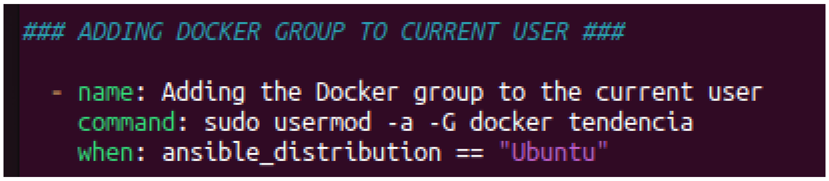


```
tendencia@workstation:~$ sudo systemctl enable docker
tendencia@workstation:~$ sudo systemctl start docker
```



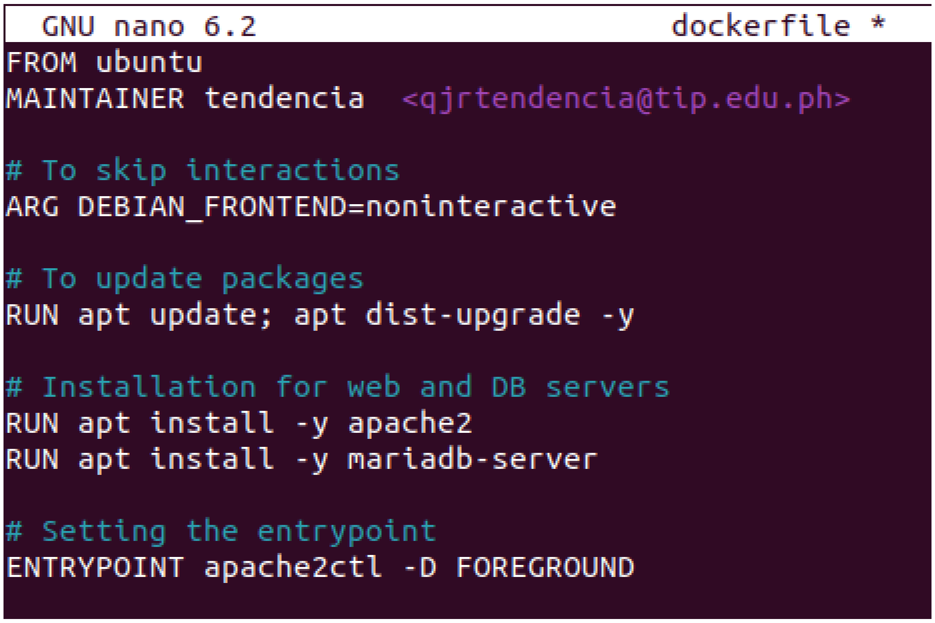
```
tendencia@workstation:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Mon 2023-11-13 12:09:03 PST; 20s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3388 (dockerd)
      Tasks: 9
     Memory: 28.6M
        CPU: 376ms
     CGroup: /system.slice/docker.service
             └─3388 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Nov 13 12:09:03 workstation systemd[1]: Starting Docker Application Container E>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.093799932+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.094946076+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.243262209+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.538654467+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.603060709+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.603187120+>
Nov 13 12:09:03 workstation dockerd[3388]: time="2023-11-13T12:09:03.655559515+>
Nov 13 12:09:03 workstation systemd[1]: Started Docker Application Container En>
lines 1-21/21 (END)...skipping...
```

3. Add to Docker group to your current user.



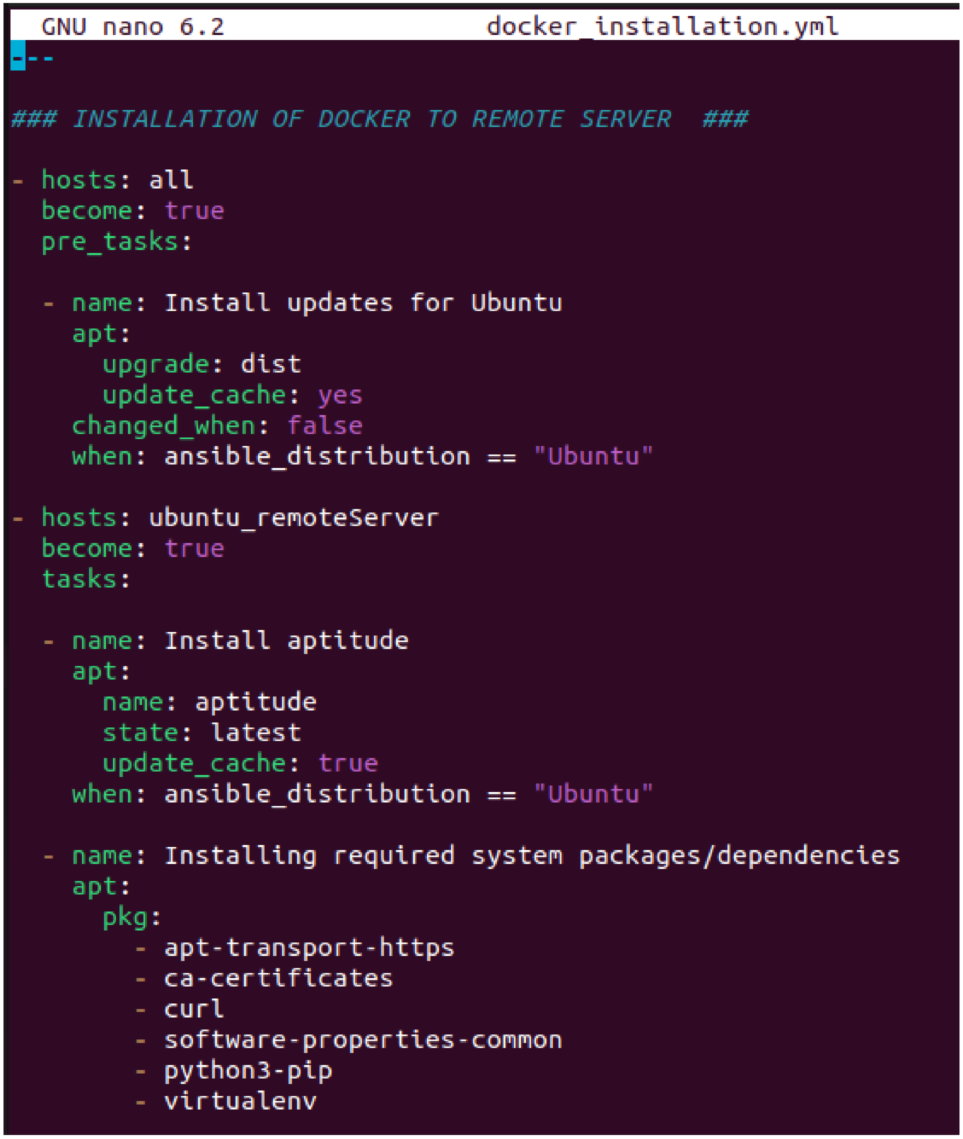
```
### ADDING DOCKER GROUP TO CURRENT USER ###

  - name: Adding the Docker group to the current user
    command: sudo usermod -a -G docker tendencia
    when: ansible_distribution == "Ubuntu"
```

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

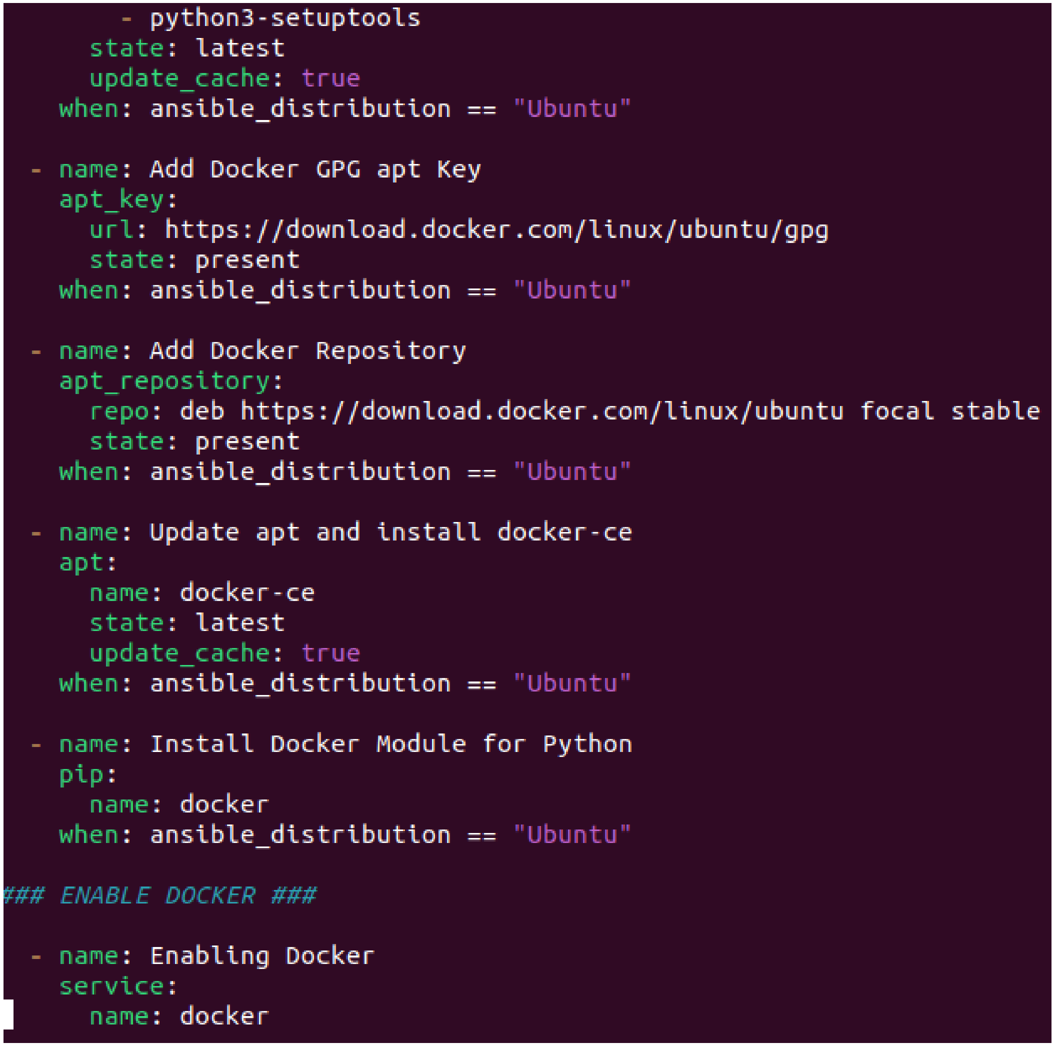


```
  GNU nano 6.2                          dockerfile *
FROM ubuntu
MAINTAINER tendencia  <qjrtendencia@tip.edu.ph>

# To skip interactions
ARG DEBIAN_FRONTEND=noninteractive

# To update packages
RUN apt update; apt dist-upgrade -y

# Installation for web and DB servers
RUN apt install -y apache2
RUN apt install -y mariadb-server

# Setting the entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
```

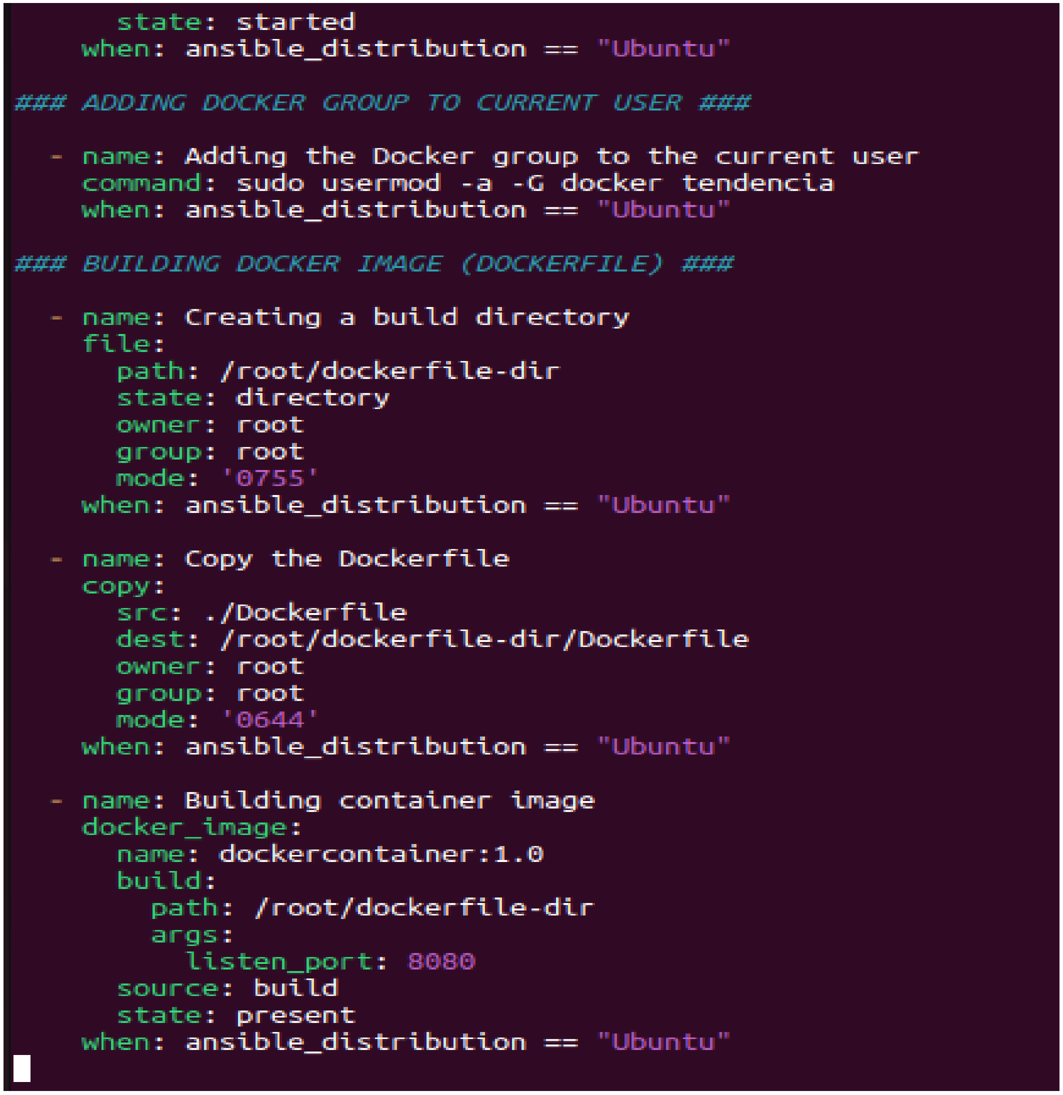
5. Install and build the Dockerfile using Ansible.



```
  GNU nano 6.2                 docker_installation.yml
---

### INSTALLATION OF DOCKER TO REMOTE SERVER  ###

- hosts: all
  become: true
  pre_tasks:

  - name: Install updates for Ubuntu
    apt:
      upgrade: dist
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: ubuntu_remoteServer
  become: true
  tasks:

  - name: Install aptitude
    apt:
      name: aptitude
      state: latest
      update_cache: true
    when: ansible_distribution == "Ubuntu"

  - name: Installing required system packages/dependencies
    apt:
      pkg:
        - apt-transport-https
        - ca-certificates
        - curl
        - software-properties-common
        - python3-pip
        - virtualenv
```

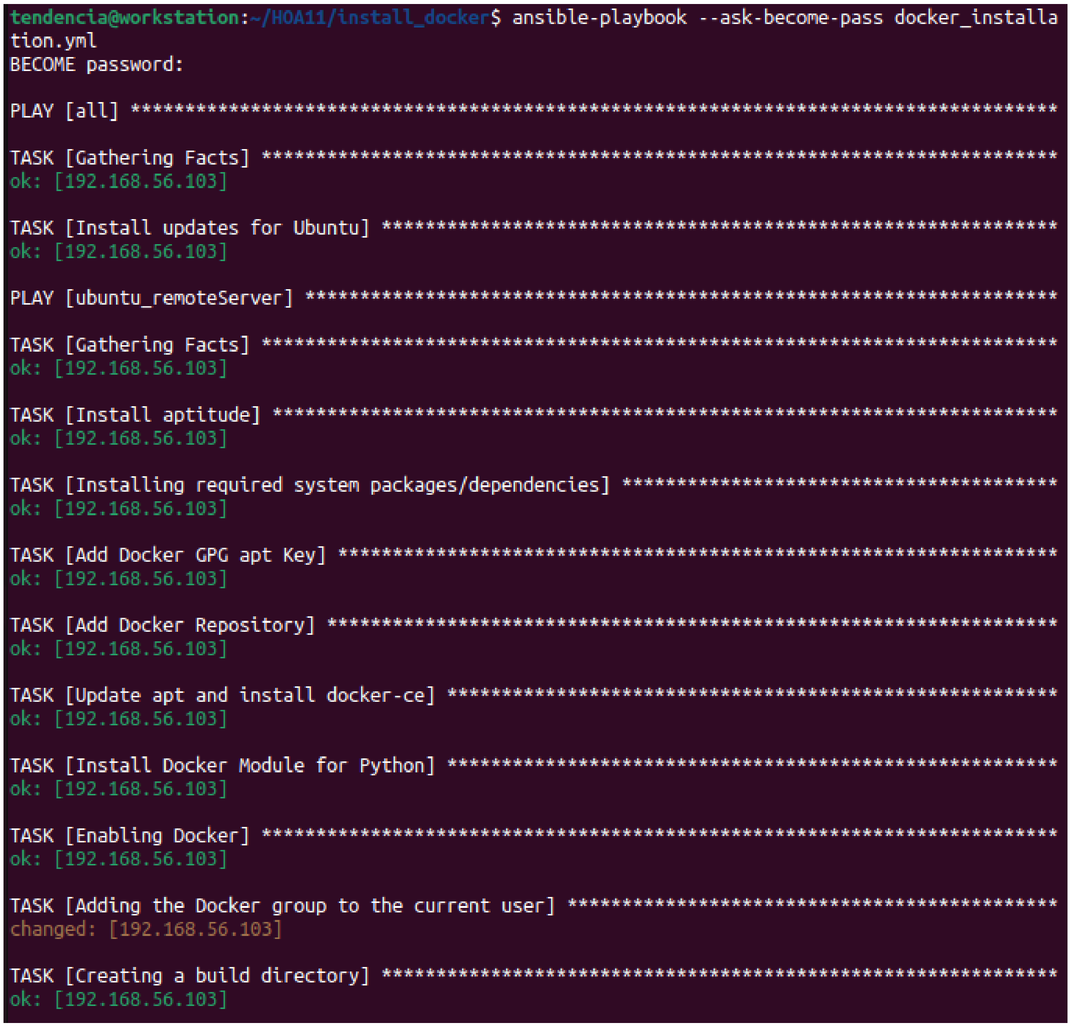


```
        - python3-setuptools
      state: latest
      update_cache: true
    when: ansible_distribution == "Ubuntu"

  - name: Add Docker GPG apt Key
    apt_key:
      url: https://download.docker.com/linux/ubuntu/gpg
      state: present
    when: ansible_distribution == "Ubuntu"

  - name: Add Docker Repository
    apt_repository:
      repo: deb https://download.docker.com/linux/ubuntu focal stable
      state: present
    when: ansible_distribution == "Ubuntu"

  - name: Update apt and install docker-ce
    apt:
      name: docker-ce
      state: latest
      update_cache: true
    when: ansible_distribution == "Ubuntu"

  - name: Install Docker Module for Python
    pip:
      name: docker
    when: ansible_distribution == "Ubuntu"

### ENABLE DOCKER ###

  - name: Enabling Docker
    service:
      name: docker
```

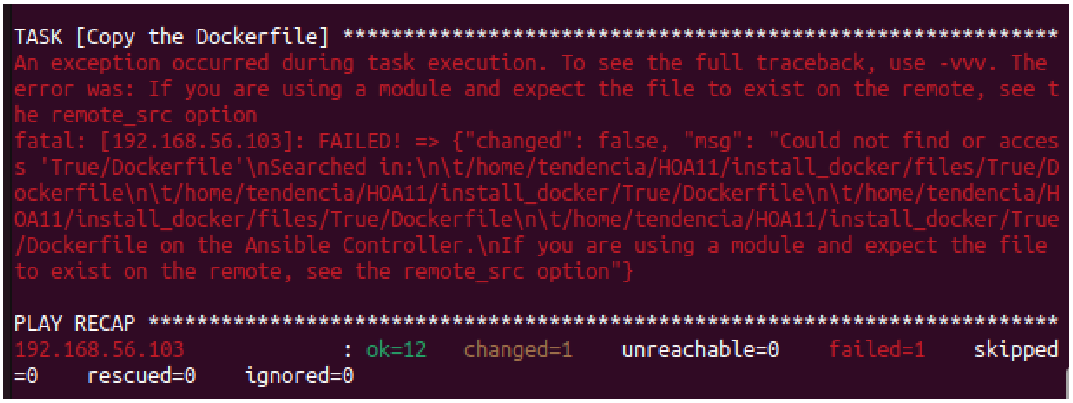


```
      state: started
    when: ansible_distribution == "Ubuntu"

### ADDING DOCKER GROUP TO CURRENT USER ###

  - name: Adding the Docker group to the current user
    command: sudo usermod -a -G docker tendencia
    when: ansible_distribution == "Ubuntu"

### BUILDING DOCKER IMAGE (DOCKERFILE) ###

  - name: Creating a build directory
    file:
      path: /root/dockerfile-dir
      state: directory
      owner: root
      group: root
      mode: '0755'
    when: ansible_distribution == "Ubuntu"

  - name: Copy the Dockerfile
    copy:
      src: ./Dockerfile
      dest: /root/dockerfile-dir/Dockerfile
      owner: root
      group: root
      mode: '0644'
    when: ansible_distribution == "Ubuntu"

  - name: Building container image
    docker_image:
      name: dockercontainer:1.0
      build:
        path: /root/dockerfile-dir
        args:
          listen_port: 8080
      source: build
      state: present
    when: ansible_distribution == "Ubuntu"
```

Result after run:

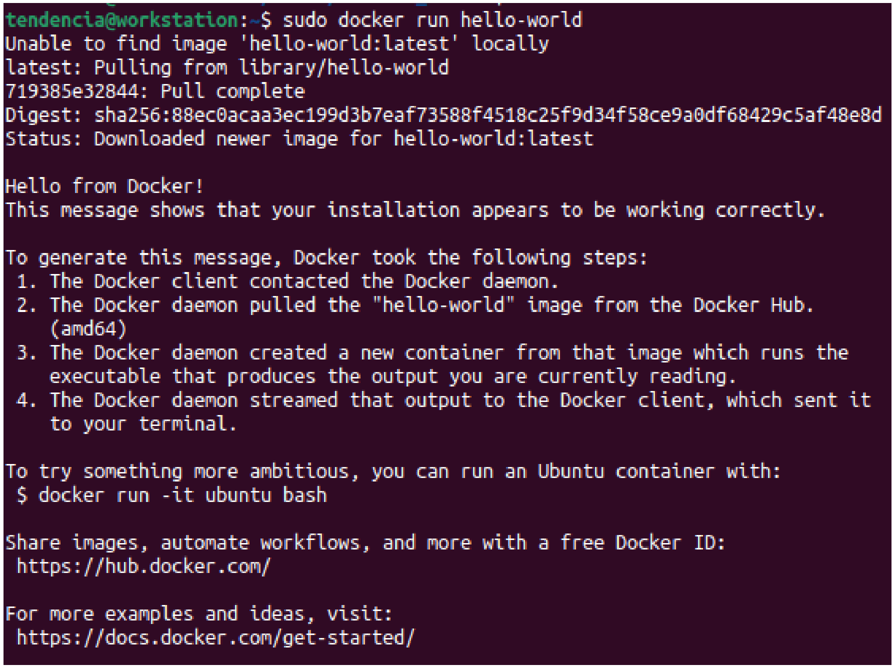


```
tendencia@workstation:~/HOA11/install_docker$ ansible-playbook --ask-become-pass docker_installa
tion.yml
BECOME password:

PLAY [all] *************************************************************************************

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.103]

TASK [Install updates for Ubuntu] **************************************************************
ok: [192.168.56.103]

PLAY [ubuntu_remoteServer] *********************************************************************

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.103]

TASK [Install aptitude] ************************************************************************
ok: [192.168.56.103]

TASK [Installing required system packages/dependencies] ****************************************
ok: [192.168.56.103]

TASK [Add Docker GPG apt Key] ******************************************************************
ok: [192.168.56.103]

TASK [Add Docker Repository] *******************************************************************
ok: [192.168.56.103]

TASK [Update apt and install docker-ce] ********************************************************
ok: [192.168.56.103]

TASK [Install Docker Module for Python] ********************************************************
ok: [192.168.56.103]

TASK [Enabling Docker] *************************************************************************
ok: [192.168.56.103]

TASK [Adding the Docker group to the current user] *********************************************
changed: [192.168.56.103]

TASK [Creating a build directory] **************************************************************
ok: [192.168.56.103]
```

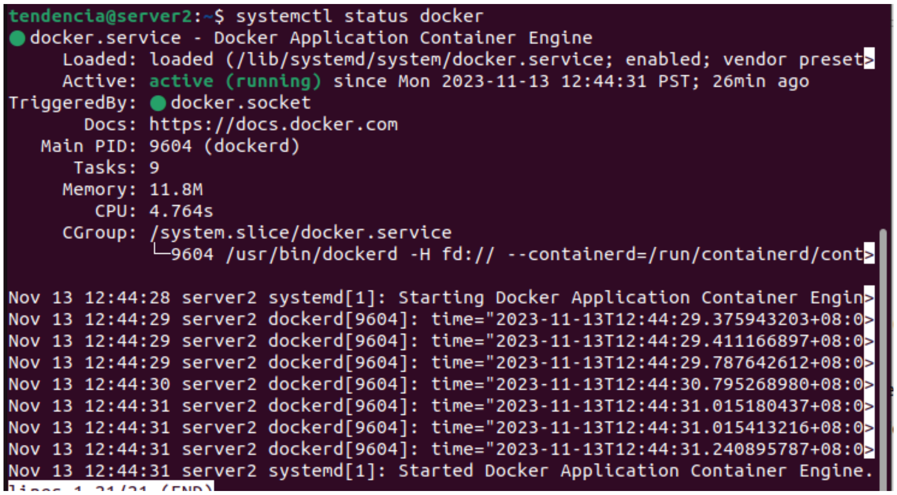


```
TASK [Copy the Dockerfile] *********************************************************************
An exception occurred during task execution. To see the full traceback, use -vvv. The
error was: If you are using a module and expect the file to exist on the remote, see t
he remote_src option
fatal: [192.168.56.103]: FAILED! => {"changed": false, "msg": "Could not find or acces
s 'True/Dockerfile'\nSearched in:\n\t/home/tendencia/HOA11/install_docker/files/True/D
ockerfile\n\t/home/tendencia/HOA11/install_docker/True/Dockerfile\n\t/home/tendencia/H
OA11/install_docker/files/True/Dockerfile\n\t/home/tendencia/HOA11/install_docker/True
/Dockerfile on the Ansible Controller.\nIf you are using a module and expect the file
to exist on the remote, see the remote_src option"}

PLAY RECAP *************************************************************************************
192.168.56.103             : ok=12   changed=1    unreachable=0    failed=1    skipped
=0    rescued=0    ignored=0
```

Verify installation of docker:

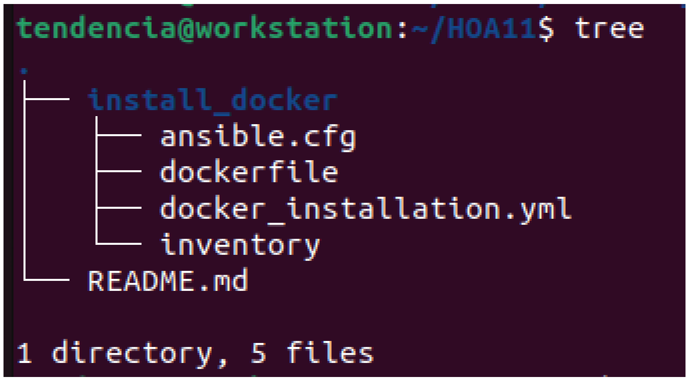


```
tendencia@workstation:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

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/
```



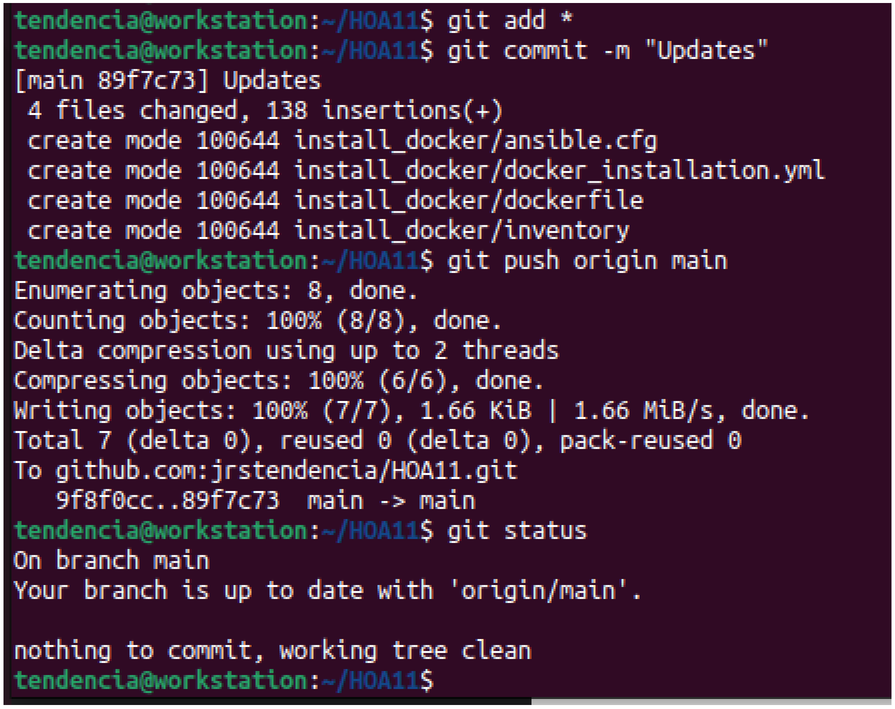
```
tendencia@server2:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Mon 2023-11-13 12:44:31 PST; 26min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 9604 (dockerd)
      Tasks: 9
     Memory: 11.8M
        CPU: 4.764s
     CGroup: /system.slice/docker.service
             └─9604 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Nov 13 12:44:28 server2 systemd[1]: Starting Docker Application Container Engin>
Nov 13 12:44:29 server2 dockerd[9604]: time="2023-11-13T12:44:29.375943203+08:0>
Nov 13 12:44:29 server2 dockerd[9604]: time="2023-11-13T12:44:29.411166897+08:0>
Nov 13 12:44:29 server2 dockerd[9604]: time="2023-11-13T12:44:29.787642612+08:0>
Nov 13 12:44:30 server2 dockerd[9604]: time="2023-11-13T12:44:30.795268980+08:0>
Nov 13 12:44:31 server2 dockerd[9604]: time="2023-11-13T12:44:31.015180437+08:0>
Nov 13 12:44:31 server2 dockerd[9604]: time="2023-11-13T12:44:31.015413216+08:0>
Nov 13 12:44:31 server2 dockerd[9604]: time="2023-11-13T12:44:31.240895787+08:0>
Nov 13 12:44:31 server2 systemd[1]: Started Docker Application Container Engine.
lines 1-21/21 (END)
```

tree:

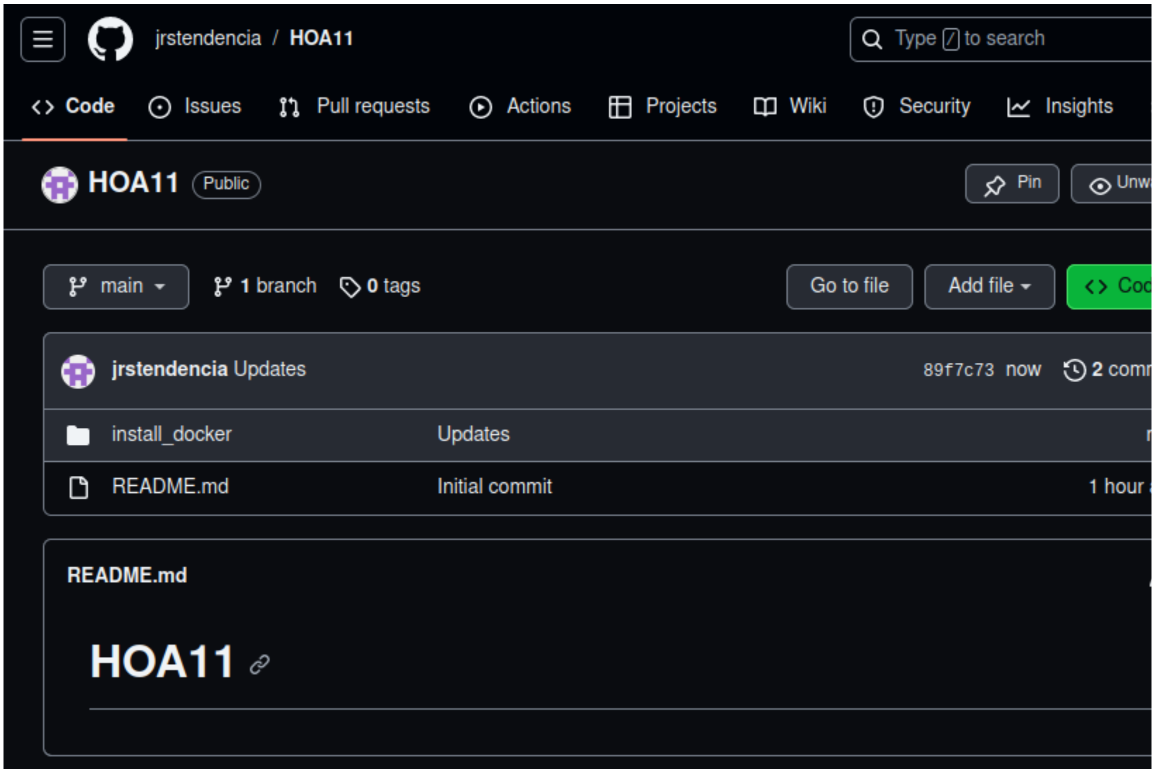


```
tendencia@workstation:~/HOA11$ tree
.
├── install_docker
│   ├── ansible.cfg
│   ├── dockerfile
│   ├── docker_installation.yml
│   └── inventory
└── README.md

1 directory, 5 files
```

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



```
tendencia@workstation:~/HOA11$ git add *
tendencia@workstation:~/HOA11$ git commit -m "Updates"
[main 89f7c73] Updates
 4 files changed, 138 insertions(+)
 create mode 100644 install_docker/ansible.cfg
 create mode 100644 install_docker/docker_installation.yml
 create mode 100644 install_docker/dockerfile
 create mode 100644 install_docker/inventory
tendencia@workstation:~/HOA11$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.66 KiB | 1.66 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jrstendencia/HOA11.git
   9f8f0cc..89f7c73  main -> main
tendencia@workstation:~/HOA11$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
tendencia@workstation:~/HOA11$
```



jrstendencia Updates 89f7c73 now 2 comm

install_docker Updates

README.md Initial commit 1 hour

README.md

HOA11

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   Implementing containerization, such as using Docker, offers several key benefits in

modern software development. First, containers enable consistent deployment across different environments, ensuring that an application runs reliably from development to production. Second, enhance scalability and resource utilization, allowing for efficient use of computing resources and easier management of large-scale applications. Third, containerization promotes isolation, enabling developers to encapsulate an application and its dependencies, reducing conflicts between different software components. This isolation enhances security by limiting the potential impact of vulnerabilities. Additionally, containers facilitate rapid development and deployment cycles, supporting continuous integration and delivery practices. Overall, containerization streamlines the development process, improves resource efficiency, and enhances the reliability and security of applications in diverse computing environments.

Conclusions:

In conclusion, the implementation of containerization brings significant advantages to software development, including consistent deployment, enhanced scalability, improved security through isolation, and streamlined development cycles. To apply these principles practically, a recommended approach involves creating a new repository for the project, installing and enabling Docker with the docker socket, adding the user to the Docker group for permissions, crafting a Dockerfile to install web and database servers, and employing Ansible to install and build the Dockerfile. This process, when completed, can be committed and pushed to the repository, ensuring a version-controlled and replicable environment for the application's deployment and management.