Name:Espiritu, Diego Angelo G.	**Date Performed: 11/16/2023**
Course/Section: CPE232/CPE31S6	**Date Submitted: 11/16/2023**
Instructor: Dr. Jonathan Vidal 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.

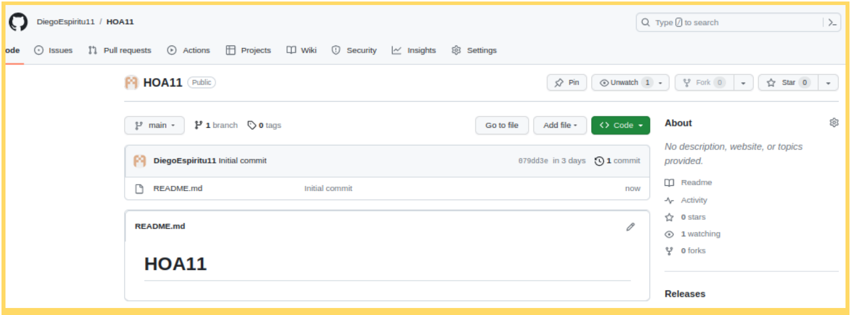
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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)

Step 1: Create a repository in github.

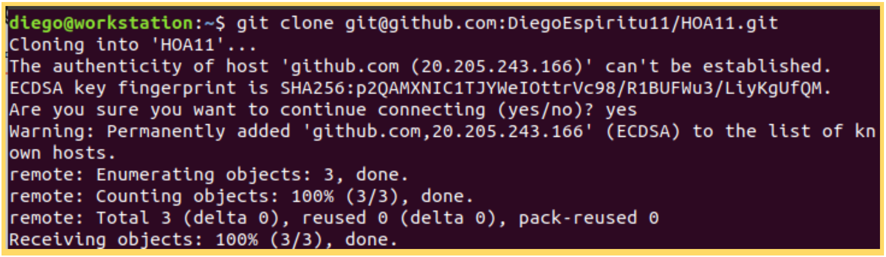


DiegoEspiritu11 Initial commit 079dd3e in 3 days 1 commit

README.md Initial commit now

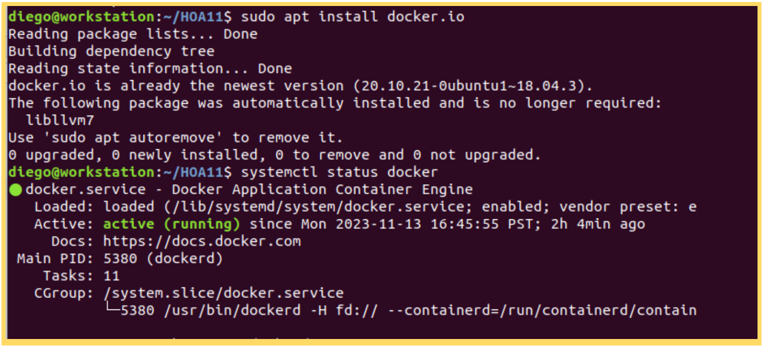
README.md

HOA11

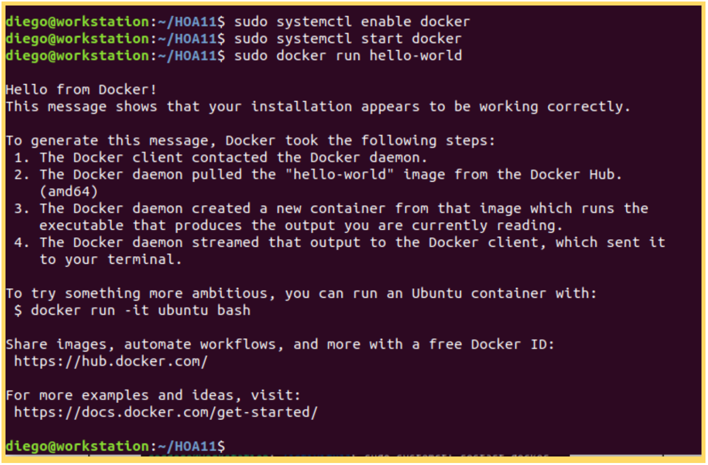


```
diego@workstation:~$ git clone git@github.com:DiegoEspiritu11/HOA11.git
Cloning into 'HOA11'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of kn
own hosts.
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.
```

Step 2: Install Docker and enable the docker socket.

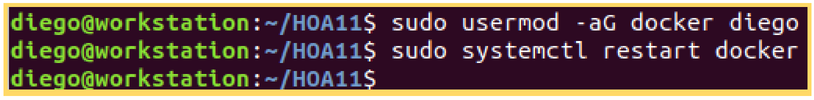


```
diego@workstation:~/HOA11$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.21-0ubuntu1~18.04.3).
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
diego@workstation:~/HOA11$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Mon 2023-11-13 16:45:55 PST; 2h 4min ago
     Docs: https://docs.docker.com
 Main PID: 5380 (dockerd)
    Tasks: 11
   CGroup: /system.slice/docker.service
           └─5380 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
```



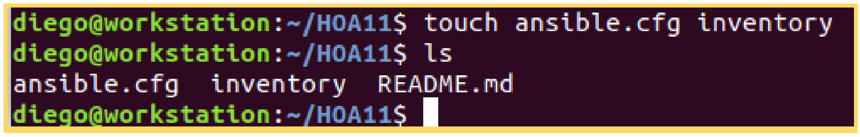
```
diego@workstation:~/HOA11$ sudo systemctl enable docker
diego@workstation:~/HOA11$ sudo systemctl start docker
diego@workstation:~/HOA11$ sudo docker run hello-world

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/

diego@workstation:~/HOA11$
```

Step 3: Add to Docker group to your current user.

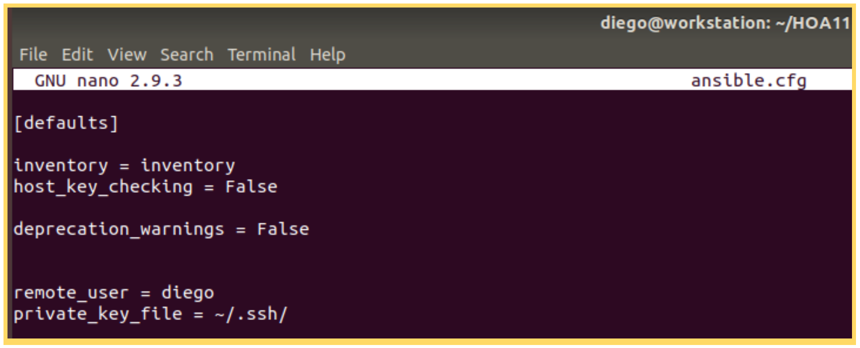


```
diego@workstation:~/HOA11$ sudo usermod -aG docker diego
diego@workstation:~/HOA11$ sudo systemctl restart docker
diego@workstation:~/HOA11$
```

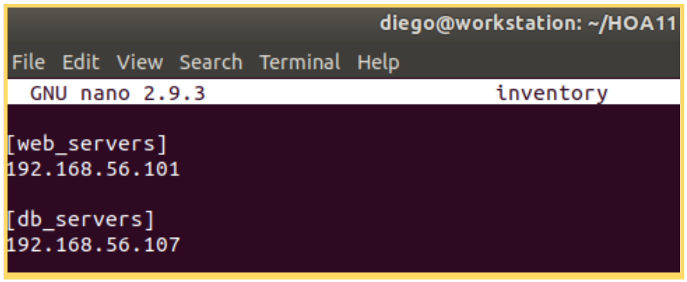
Step 4: Create a Dockerfile to install web and DB server.



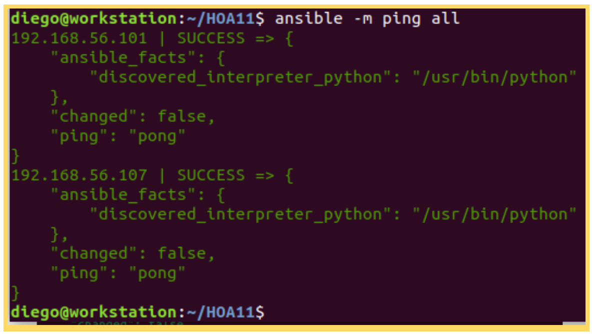
```
diego@workstation:~/HOA11$ touch ansible.cfg inventory
diego@workstation:~/HOA11$ ls
ansible.cfg  inventory  README.md
diego@workstation:~/HOA11$
```


diego@workstation: ~/HOA11
File Edit View Search Terminal Help

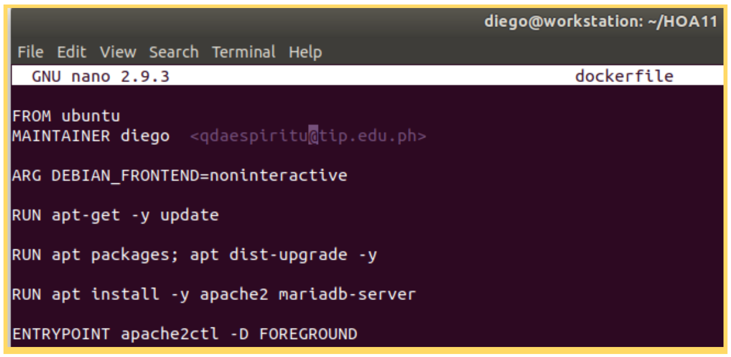
```
  GNU nano 2.9.3                                            ansible.cfg

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False


remote_user = diego
private_key_file = ~/.ssh/
```


diego@workstation: ~/HOA11
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                          inventory

[web_servers]
192.168.56.101

[db_servers]
192.168.56.107
```

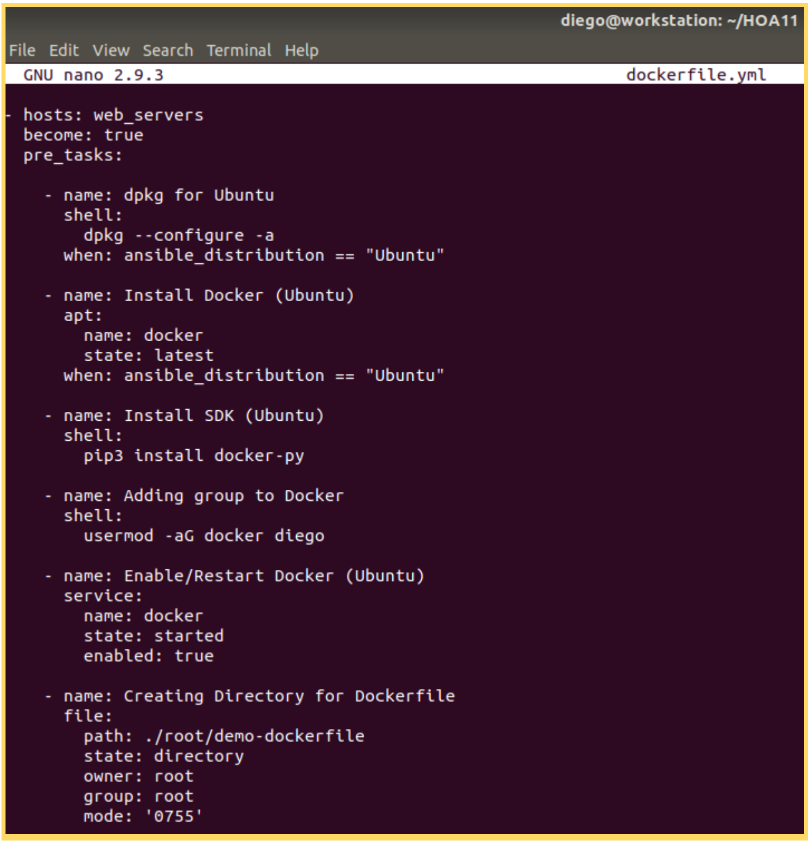


```
diego@workstation:~/HOA11$ ansible -m ping all
192.168.56.101 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.107 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
diego@workstation:~/HOA11$
```

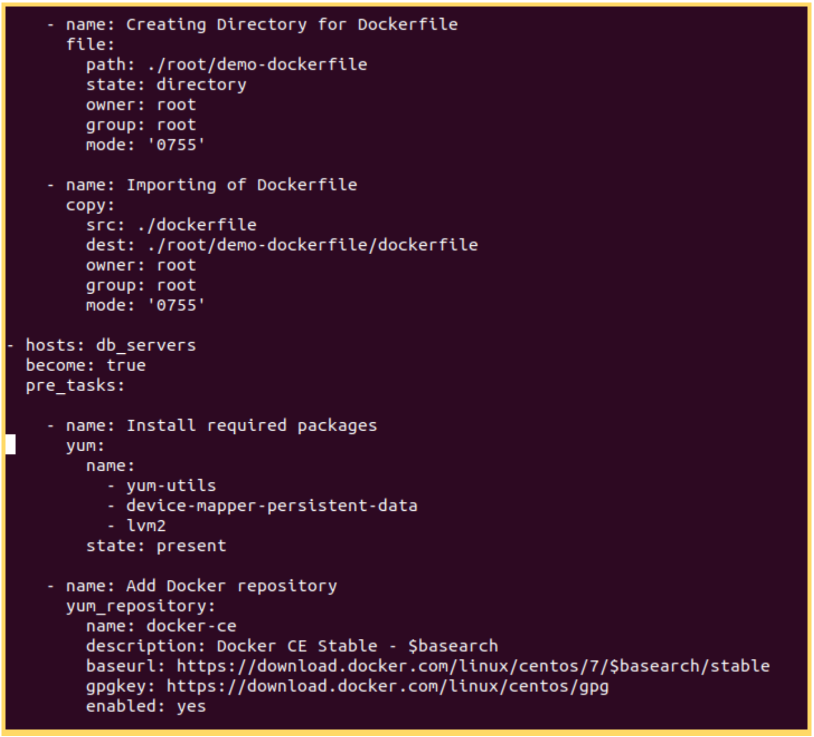

diego@workstation: ~/HOA11
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                                     dockerfile

FROM ubuntu
MAINTAINER diego  <qdaespiritu@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get -y update

RUN apt packages; apt dist-upgrade -y

RUN apt install -y apache2 mariadb-server

ENTRYPOINT apache2ctl -D FOREGROUND
```

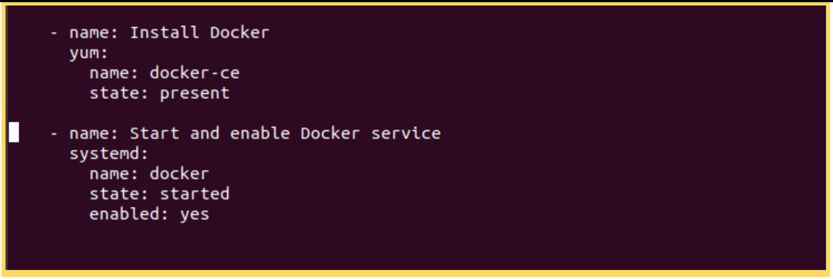
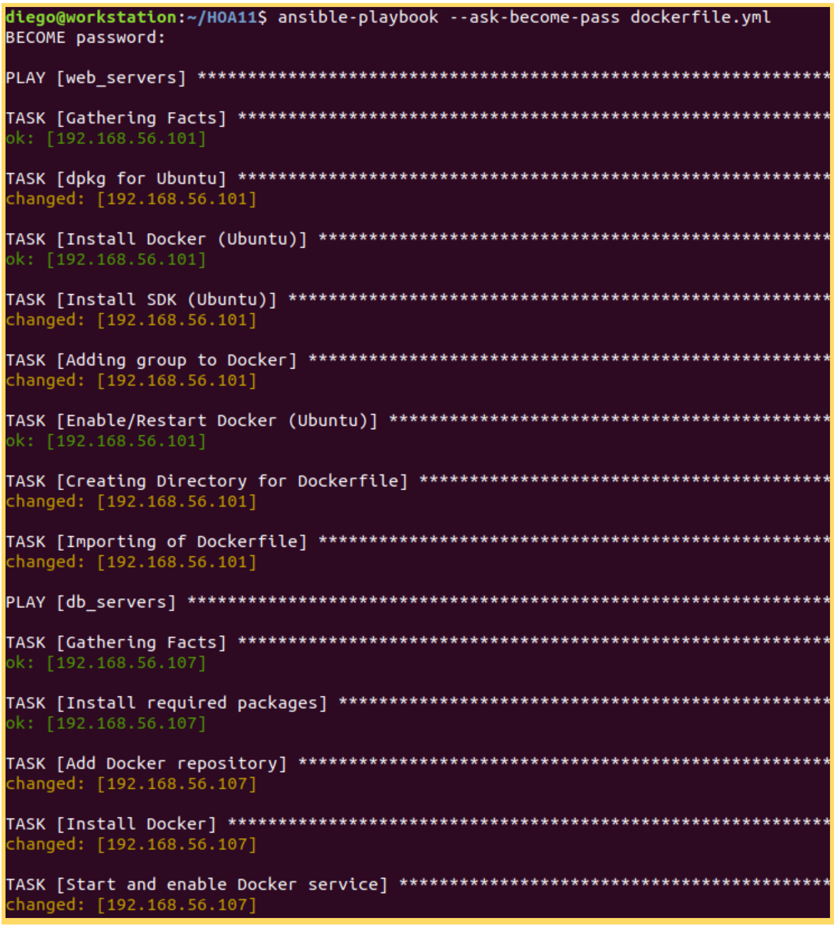
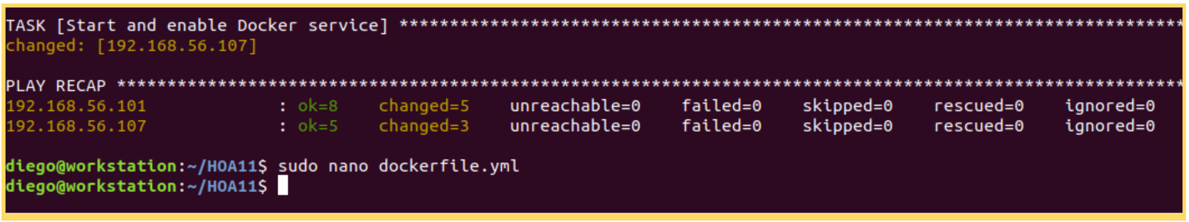
Step 5: Install and build the Dockerfile using Ansible.

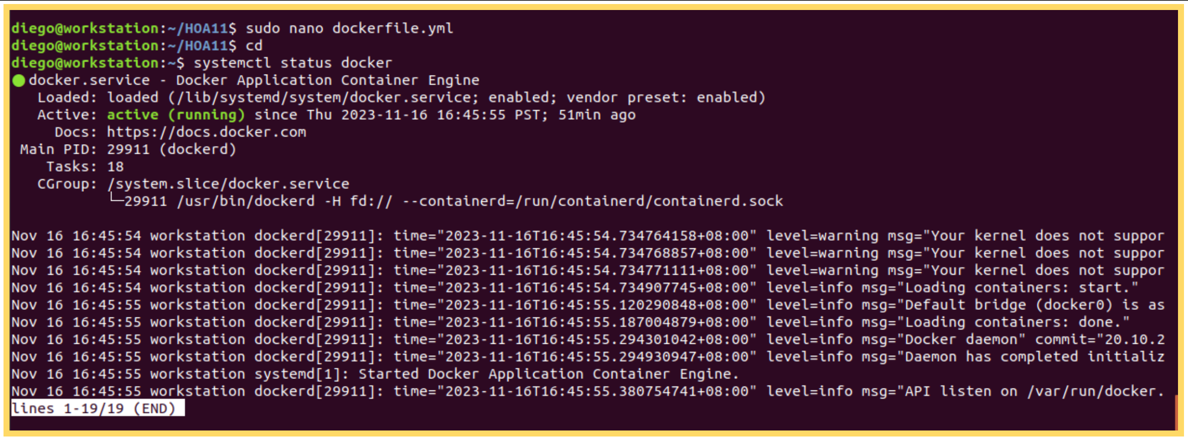

diego@workstation: ~/HOA11
File Edit View Search Terminal Help
GNU nano 2.9.3
dockerfile.yml

```
- hosts: web_servers
  become: true
  pre_tasks:

    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: Install Docker (Ubuntu)
      apt:
        name: docker
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install SDK (Ubuntu)
      shell:
        pip3 install docker-py

    - name: Adding group to Docker
      shell:
        usermod -aG docker diego

    - name: Enable/Restart Docker (Ubuntu)
      service:
        name: docker
        state: started
        enabled: true

    - name: Creating Directory for Dockerfile
      file:
        path: ./root/demo-dockerfile
        state: directory
        owner: root
        group: root
        mode: '0755'
```

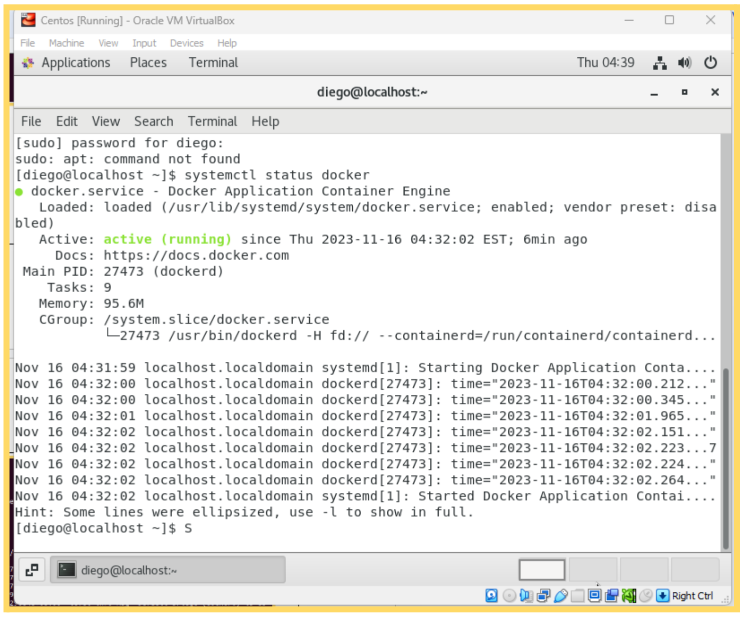


```
    - name: Creating Directory for Dockerfile
      file:
        path: ./root/demo-dockerfile
        state: directory
        owner: root
        group: root
        mode: '0755'

    - name: Importing of Dockerfile
      copy:
        src: ./dockerfile
        dest: ./root/demo-dockerfile/dockerfile
        owner: root
        group: root
        mode: '0755'

- hosts: db_servers
  become: true
  pre_tasks:

    - name: Install required packages
      yum:
        name:
          - yum-utils
          - device-mapper-persistent-data
          - lvm2
        state: present

    - name: Add Docker repository
      yum_repository:
        name: docker-ce
        description: Docker CE Stable - $basearch
        baseurl: https://download.docker.com/linux/centos/7/$basearch/stable
        gpgkey: https://download.docker.com/linux/centos/gpg
        enabled: yes
```

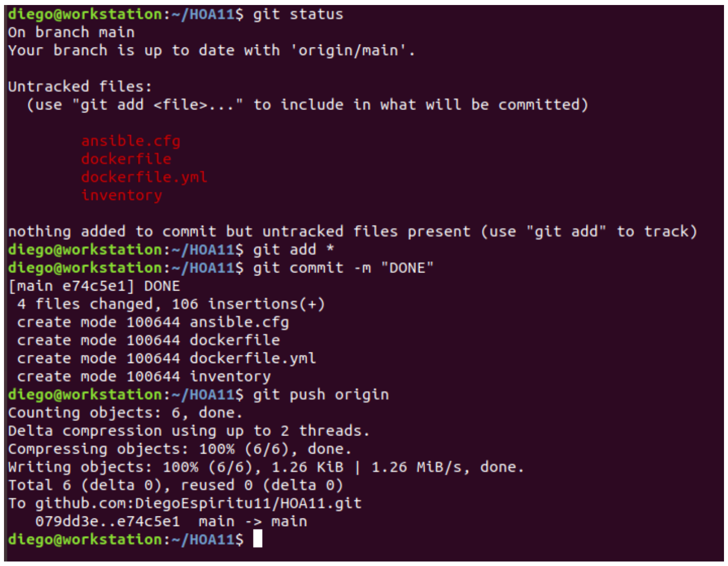

```
    - name: Install Docker
      yum:
        name: docker-ce
        state: present

    - name: Start and enable Docker service
      systemd:
        name: docker
        state: started
        enabled: yes
```

```
diego@workstation:~/HOA11$ ansible-playbook --ask-become-pass dockerfile.yml
BECOME password:

PLAY [web_servers] ***********************************************************

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.101]

TASK [dpkg for Ubuntu] *******************************************************
changed: [192.168.56.101]

TASK [Install Docker (Ubuntu)] ***********************************************
ok: [192.168.56.101]

TASK [Install SDK (Ubuntu)] **************************************************
changed: [192.168.56.101]

TASK [Adding group to Docker] ************************************************
changed: [192.168.56.101]

TASK [Enable/Restart Docker (Ubuntu)] ****************************************
ok: [192.168.56.101]

TASK [Creating Directory for Dockerfile] *************************************
changed: [192.168.56.101]

TASK [Importing of Dockerfile] ***********************************************
changed: [192.168.56.101]

PLAY [db_servers] ************************************************************

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.107]

TASK [Install required packages] *********************************************
ok: [192.168.56.107]

TASK [Add Docker repository] *************************************************
changed: [192.168.56.107]

TASK [Install Docker] ********************************************************
changed: [192.168.56.107]

TASK [Start and enable Docker service] ***************************************
changed: [192.168.56.107]
```

```
TASK [Start and enable Docker service] ************************************************************
changed: [192.168.56.107]

PLAY RECAP ****************************************************************************************
192.168.56.101             : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.107             : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

diego@workstation:~/HOA11$ sudo nano dockerfile.yml
diego@workstation:~/HOA11$ 
```



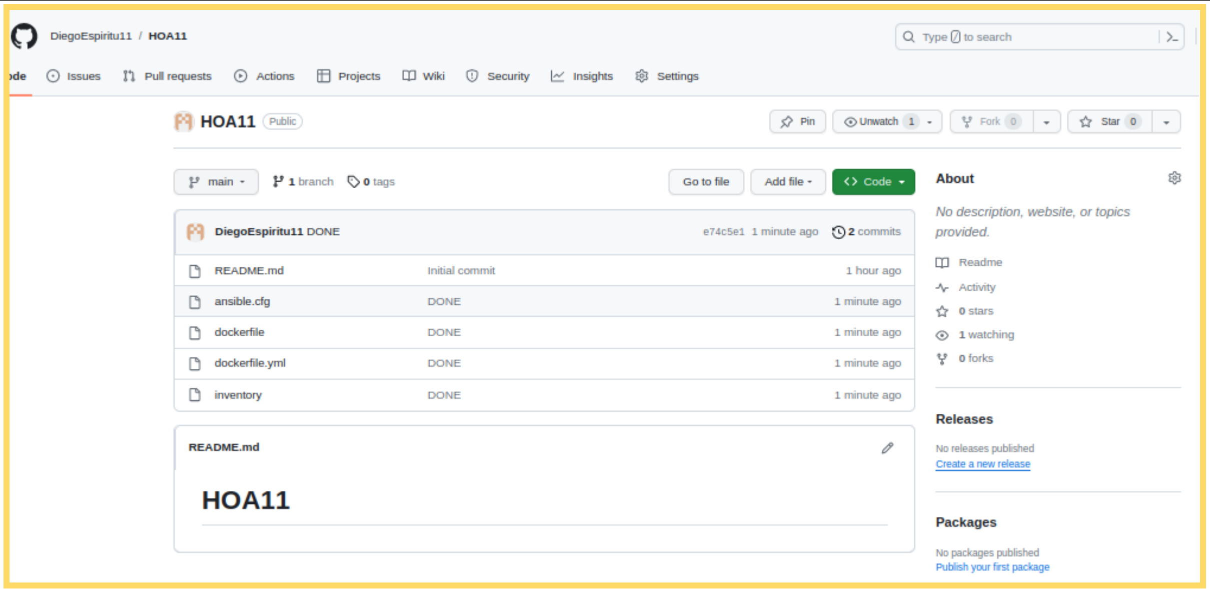
```
diego@workstation:~/HOA11$ sudo nano dockerfile.yml
diego@workstation:~/HOA11$ cd
diego@workstation:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-16 16:45:55 PST; 51min ago
     Docs: https://docs.docker.com
 Main PID: 29911 (dockerd)
    Tasks: 18
   CGroup: /system.slice/docker.service
           └─29911 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 16 16:45:54 workstation dockerd[29911]: time="2023-11-16T16:45:54.734764158+08:00" level=warning msg="Your kernel does not suppor
Nov 16 16:45:54 workstation dockerd[29911]: time="2023-11-16T16:45:54.734768857+08:00" level=warning msg="Your kernel does not suppor
Nov 16 16:45:54 workstation dockerd[29911]: time="2023-11-16T16:45:54.734771111+08:00" level=warning msg="Your kernel does not suppor
Nov 16 16:45:54 workstation dockerd[29911]: time="2023-11-16T16:45:54.734907745+08:00" level=info msg="Loading containers: start."
Nov 16 16:45:55 workstation dockerd[29911]: time="2023-11-16T16:45:55.120290848+08:00" level=info msg="Default bridge (docker0) is as
Nov 16 16:45:55 workstation dockerd[29911]: time="2023-11-16T16:45:55.187004879+08:00" level=info msg="Loading containers: done."
Nov 16 16:45:55 workstation dockerd[29911]: time="2023-11-16T16:45:55.294301042+08:00" level=info msg="Docker daemon" commit="20.10.2
Nov 16 16:45:55 workstation dockerd[29911]: time="2023-11-16T16:45:55.294930947+08:00" level=info msg="Daemon has completed initializ
Nov 16 16:45:55 workstation systemd[1]: Started Docker Application Container Engine.
Nov 16 16:45:55 workstation dockerd[29911]: time="2023-11-16T16:45:55.380754741+08:00" level=info msg="API listen on /var/run/docker.
lines 1-19/19 (END)
```


Centos [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 04:39
diego@localhost:~
File Edit View Search Terminal Help
diego@localhost:~
Right Ctrl

```
[sudo] password for diego:
sudo: apt: command not found
[diego@localhost ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disa
bled)
   Active: active (running) since Thu 2023-11-16 04:32:02 EST; 6min ago
     Docs: https://docs.docker.com
 Main PID: 27473 (dockerd)
    Tasks: 9
   Memory: 95.6M
   CGroup: /system.slice/docker.service
           └─27473 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd...

Nov 16 04:31:59 localhost.localdomain systemd[1]: Starting Docker Application Conta....
Nov 16 04:32:00 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:00.212..."
Nov 16 04:32:00 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:00.345..."
Nov 16 04:32:01 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:01.965..."
Nov 16 04:32:02 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:02.151..."
Nov 16 04:32:02 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:02.223...7
Nov 16 04:32:02 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:02.224..."
Nov 16 04:32:02 localhost.localdomain dockerd[27473]: time="2023-11-16T04:32:02.264..."
Nov 16 04:32:02 localhost.localdomain systemd[1]: Started Docker Application Contai....
Hint: Some lines were ellipsized, use -l to show in full.
[diego@localhost ~]$ S
```



```
diego@workstation:~/HOA11$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ansible.cfg
        dockerfile
        dockerfile.yml
        inventory

nothing added to commit but untracked files present (use "git add" to track)
diego@workstation:~/HOA11$ git add *
diego@workstation:~/HOA11$ git commit -m "DONE"
[main e74c5e1] DONE
 4 files changed, 106 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile
 create mode 100644 dockerfile.yml
 create mode 100644 inventory
diego@workstation:~/HOA11$ git push origin
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.26 KiB | 1.26 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:DiegoEspiritu11/HOA11.git
   079dd3e..e74c5e1  main -> main
diego@workstation:~/HOA11$
```



DiegoEspiritu11 DONE e74c5e1 1 minute ago 2 commits

README.md	Initial commit	1 hour ago
ansible.cfg	DONE	1 minute ago
dockerfile	DONE	1 minute ago
dockerfile.yml	DONE	1 minute ago
inventory	DONE	1 minute ago

README.md

HOA11

No description, website, or topics provided.

No releases published

No packages published

github.com/DiegoEspiritu11/HOA11

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   -Implementing containerization offers several advantages, including reducing overhead, enhancing portability, ensuring consistent operation, and increasing efficiency. Containerization optimizes the utilization of resources and minimizes overhead by utilizing all available resources on a host. Unlike virtual machines and other traditional application architectures, containers are more resource-efficient as they can share an operating system.

Conclusions:

In conclusion containers are highly beneficial and contribute to convenience in work processes. They reduce overhead, enhance portability, ensure more consistent operation, and improve overall efficiency. Throughout the process, we gained knowledge on creating a Dockerfile and establishing a workflow using Ansible as infrastructure-as-code, enabling a continuous delivery process.