Name: Recto, Jon Jeous J.	**Date Performed: Apr 16, 2024**
Course/Section: CPE31S1	**Date Submitted: Apr 16, 2024**
Instructor: Dr. Taylar	**Semester and SY: 2nd Sem 23-24**

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.

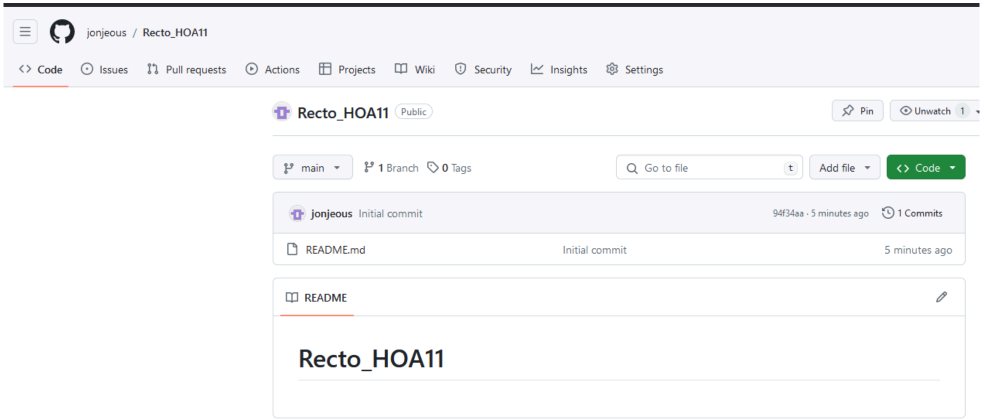
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

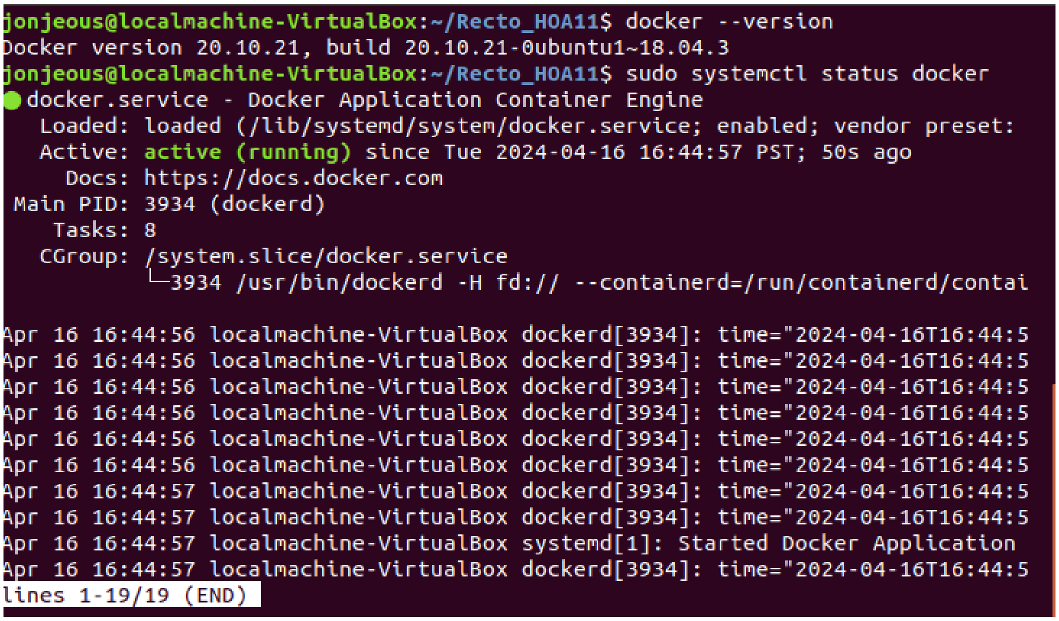
1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. Output (screenshots and explanations)

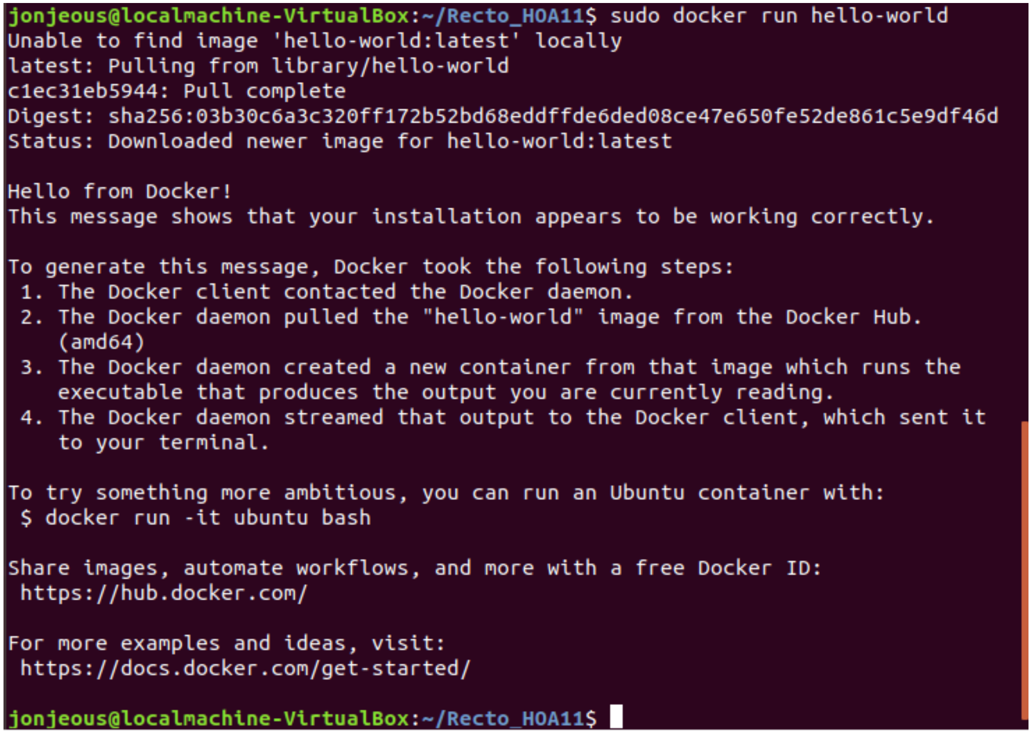
1. Created a new repository named Recto_HOA11



2. Install and run docker

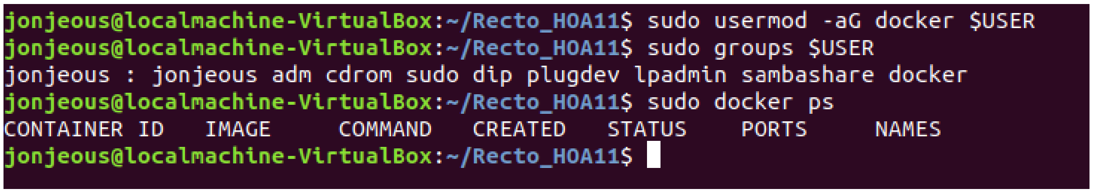


```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~18.04.3
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Tue 2024-04-16 16:44:57 PST; 50s ago
     Docs: https://docs.docker.com
 Main PID: 3934 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─3934 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:56 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:57 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:57 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
Apr 16 16:44:57 localmachine-VirtualBox systemd[1]: Started Docker Application
Apr 16 16:44:57 localmachine-VirtualBox dockerd[3934]: time="2024-04-16T16:44:5
lines 1-19/19 (END)
```



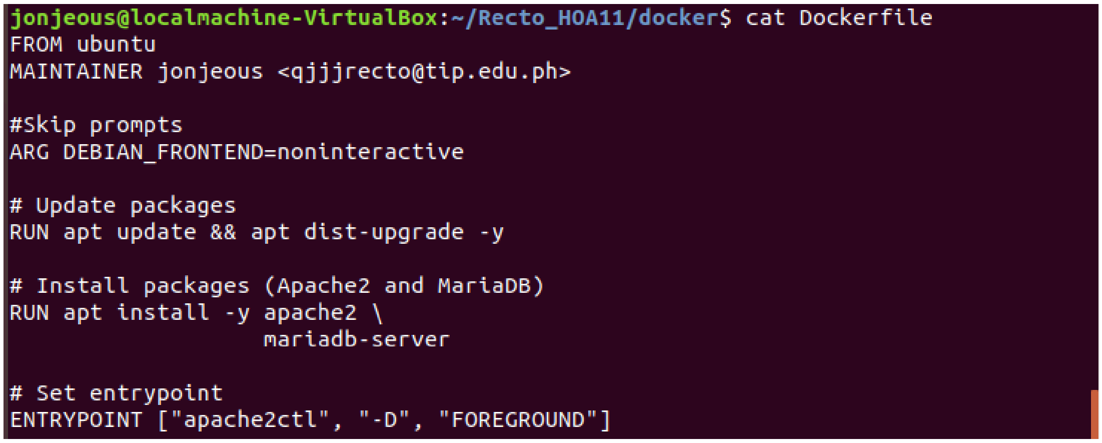
```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:03b30c6a3c320ff172b52bd68eddffde6ded08ce47e650fe52de861c5e9df46d
Status: Downloaded newer 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/

jonjeous@localmachine-VirtualBox:~/Recto_HOA11$
```

3. Add docker group

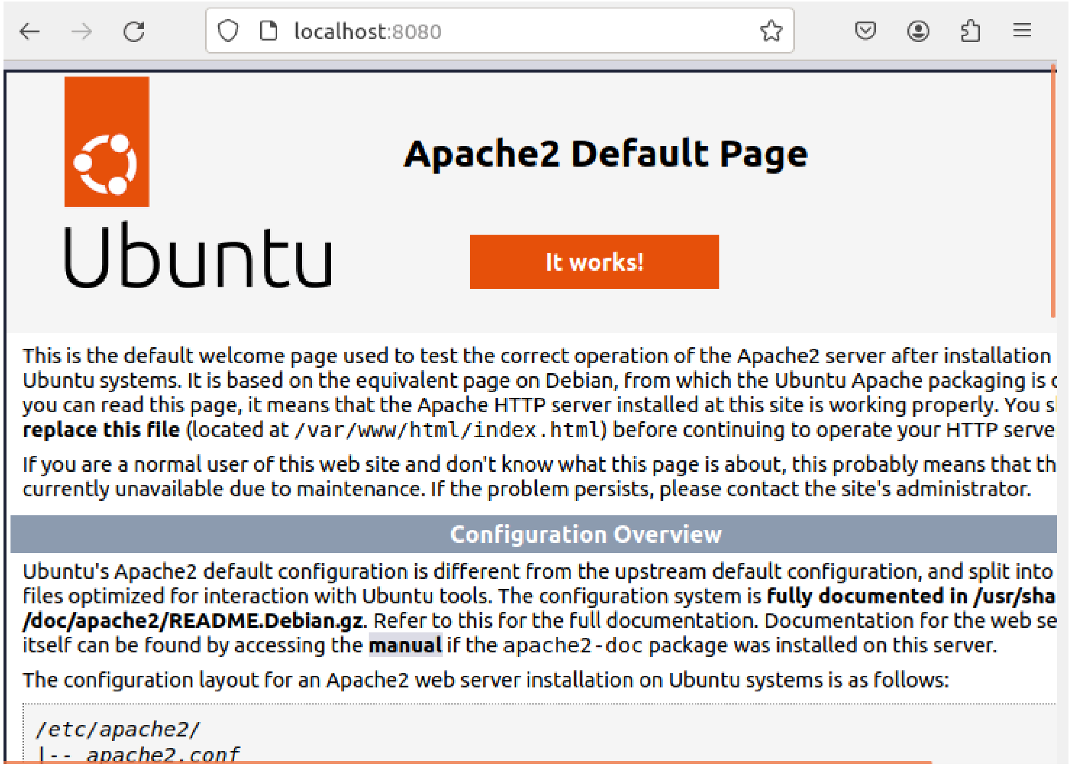


```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo usermod -aG docker $USER
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo groups $USER
jonjeous : jonjeous adm cdrom sudo dip plugdev lpadmin sambashare docker
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$
```

4. Dockerfile



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11/docker$ cat Dockerfile
FROM ubuntu
MAINTAINER jonjeous <qjjjrecto@tip.edu.ph>

#Skip prompts
ARG DEBIAN_FRONTEND=noninteractive

# Update packages
RUN apt update && apt dist-upgrade -y

# Install packages (Apache2 and MariaDB)
RUN apt install -y apache2 \
                   mariadb-server

# Set entrypoint
ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
```



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation
Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is d
you can read this page, it means that the Apache HTTP server installed at this site is working properly. You s
replace this file (located at `/var/www/html/index.html`) before continuing to operate your HTTP serve

If you are a normal user of this web site and don't know what this page is about, this probably means that th
currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

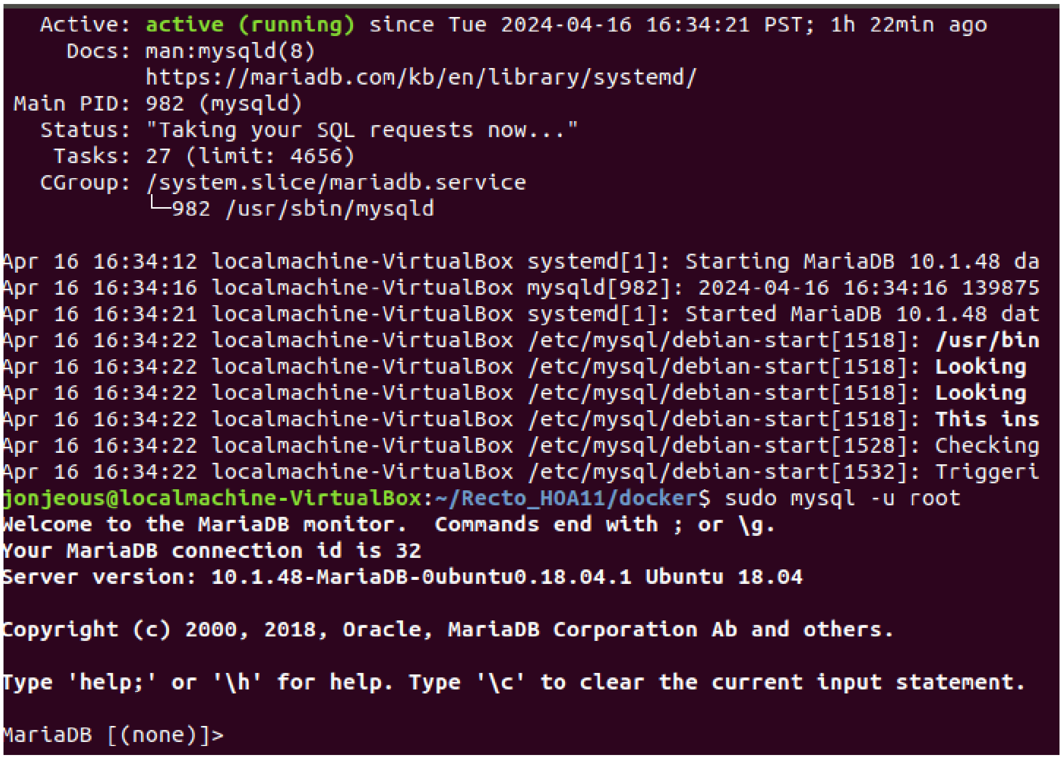
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into
files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/sha
/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web se
itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
```

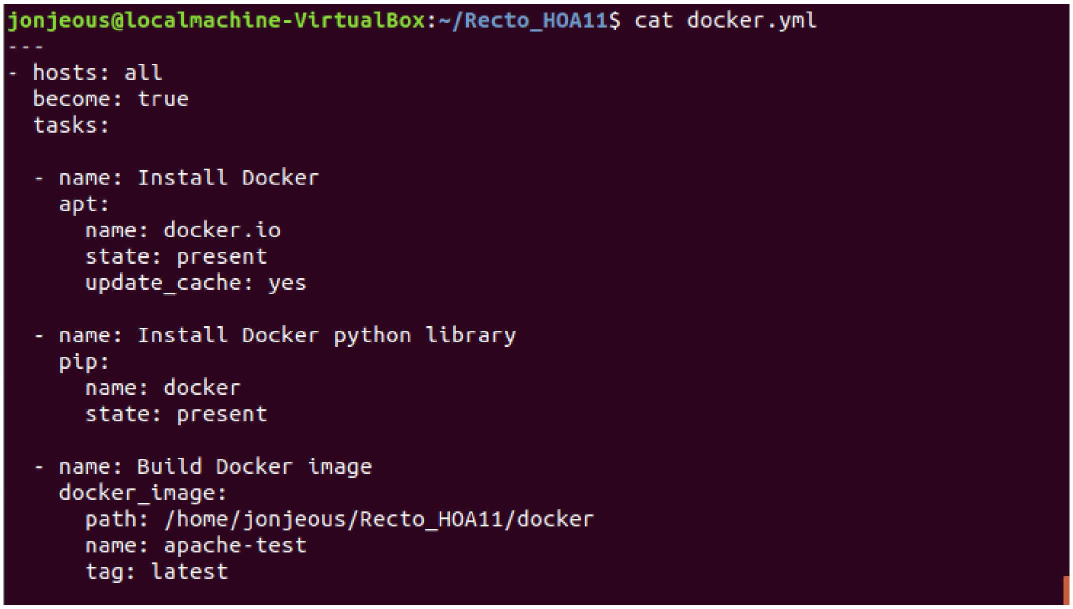
apache2



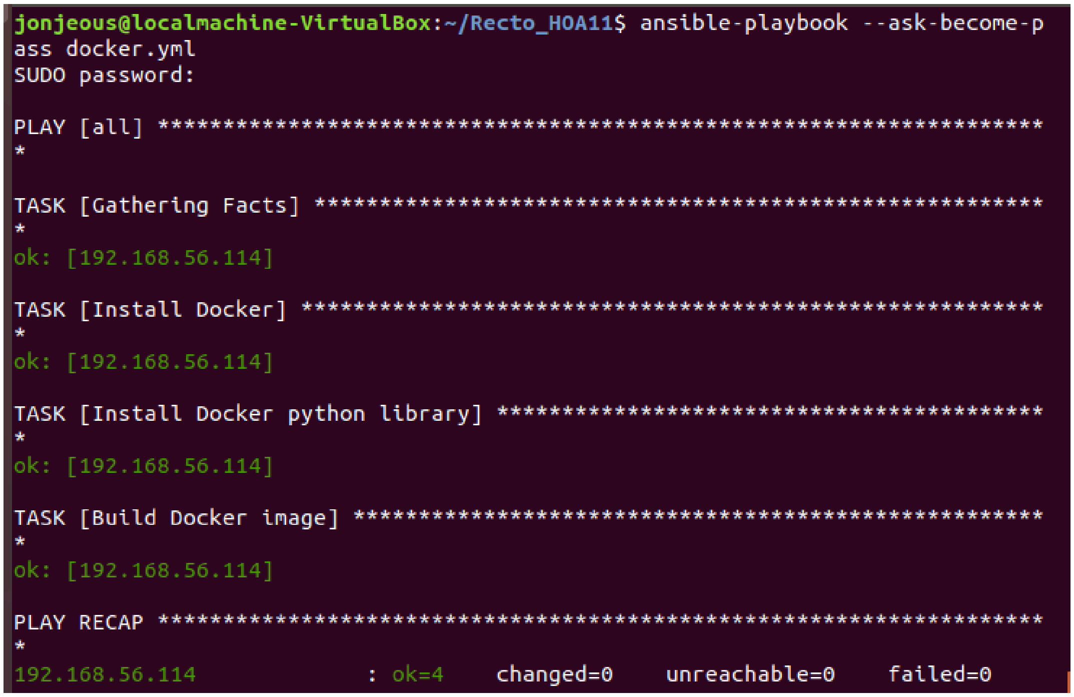
```
   Active: active (running) since Tue 2024-04-16 16:34:21 PST; 1h 22min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 982 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4656)
   CGroup: /system.slice/mariadb.service
           └─982 /usr/sbin/mysqld

Apr 16 16:34:12 localmachine-VirtualBox systemd[1]: Starting MariaDB 10.1.48 da
Apr 16 16:34:16 localmachine-VirtualBox mysqld[982]: 2024-04-16 16:34:16 139875
Apr 16 16:34:21 localmachine-VirtualBox systemd[1]: Started MariaDB 10.1.48 dat
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: /usr/bin
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: Looking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: Looking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: This ins
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1528]: Checking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1532]: Triggeri
jonjeous@localmachine-VirtualBox:~/Recto_HOA11/docker$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.1.48-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

MariaDB

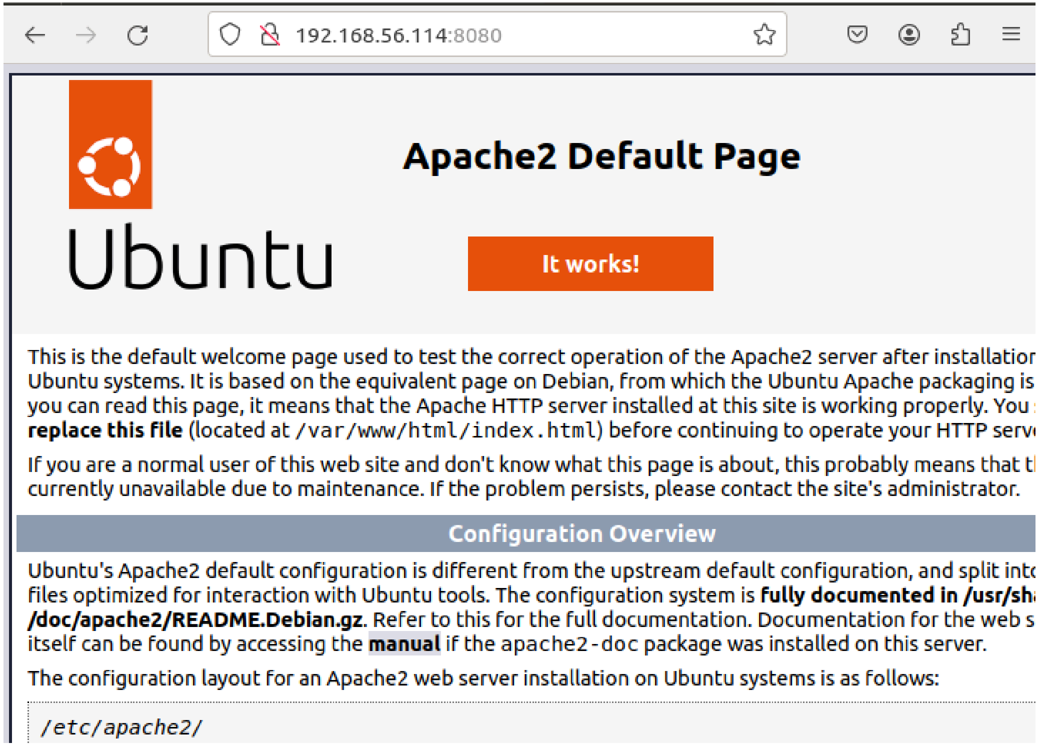
5. Dockerfile using ansible



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ cat docker.yml
---
- hosts: all
  become: true
  tasks:

  - name: Install Docker
    apt:
      name: docker.io
      state: present
      update_cache: yes

  - name: Install Docker python library
    pip:
      name: docker
      state: present

  - name: Build Docker image
    docker_image:
      path: /home/jonjeous/Recto_HOA11/docker
      name: apache-test
      tag: latest
```



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ ansible-playbook --ask-become-p
ass docker.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.114]

TASK [Install Docker] **********************************************************
*
ok: [192.168.56.114]

TASK [Install Docker python library] *******************************************
*
ok: [192.168.56.114]

TASK [Build Docker image] ******************************************************
*
ok: [192.168.56.114]

PLAY RECAP *********************************************************************
*
192.168.56.114             : ok=4    changed=0    unreachable=0    failed=0
```



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installatior Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is you can read this page, it means that the Apache HTTP server installed at this site is working properly. You **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP serv

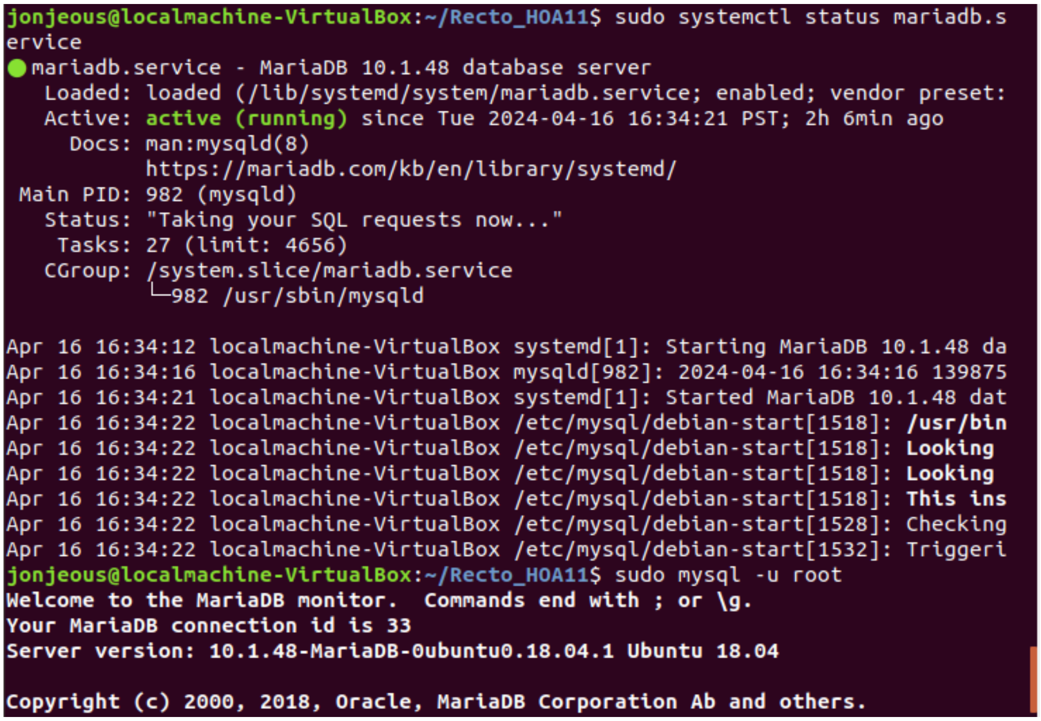
If you are a normal user of this web site and don't know what this page is about, this probably means that t currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split intc files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/sh /doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web s itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

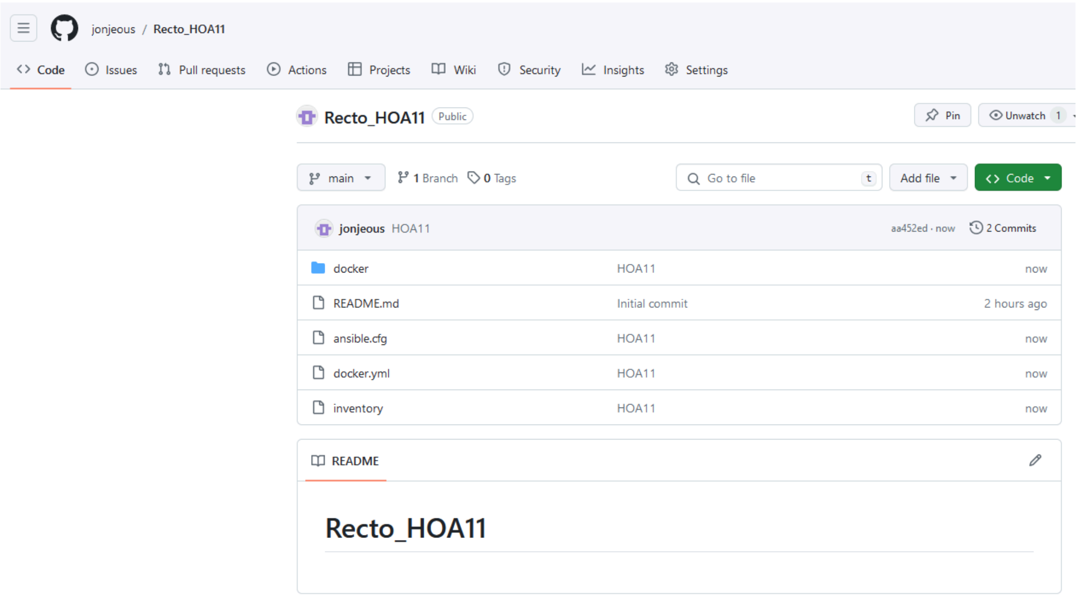
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
```



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo systemctl status mariadb.s
ervice
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Tue 2024-04-16 16:34:21 PST; 2h 6min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 982 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4656)
   CGroup: /system.slice/mariadb.service
           └─982 /usr/sbin/mysqld

Apr 16 16:34:12 localmachine-VirtualBox systemd[1]: Starting MariaDB 10.1.48 da
Apr 16 16:34:16 localmachine-VirtualBox mysqld[982]: 2024-04-16 16:34:16 139875
Apr 16 16:34:21 localmachine-VirtualBox systemd[1]: Started MariaDB 10.1.48 dat
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: /usr/bin
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: Looking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: Looking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1518]: This ins
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1528]: Checking
Apr 16 16:34:22 localmachine-VirtualBox /etc/mysql/debian-start[1532]: Triggeri
jonjeous@localmachine-VirtualBox:~/Recto_HOA11$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.1.48-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

6. Commit to repository



jonjeous HOA11 aa452ed · now 2 Commits

docker	HOA11	now
README.md	Initial commit	2 hours ago
ansible.cfg	HOA11	now
docker.yml	HOA11	now
inventory	HOA11	now

Recto_HOA11

https://github.com/jonjeous/Recto_HOA11.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   Containerization simplifies the movement of software by packaging everything it needs to run consistently across different environments, ensuring uniform behavior regardless of where it's deployed. It optimizes resource usage, allowing multiple applications to run efficiently on the same system without performance degradation.

Conclusions:

In conclusion, the implementation of containerization, coupled with Infrastructure as Code (IaC) principles using Ansible, offers a streamlined approach to software development and deployment. By utilizing Docker for containerization, applications can be easily packaged, deployed, and managed, enhancing portability, consistency, and resource efficiency. Ansible automation further accelerates the deployment process, ensuring seamless provisioning and configuration of infrastructure components.