Name: Aducal, John Mark S.	**Date Performed:** 11 / 15 / 2022
Course/Section: CPE232 – CPE31S24	**Date Submitted:** 11 / 16 / 2022
Instructor: Engr. Jonathan Taylar	**Semester and SY:** 1st Sem SY 2022-2023

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.

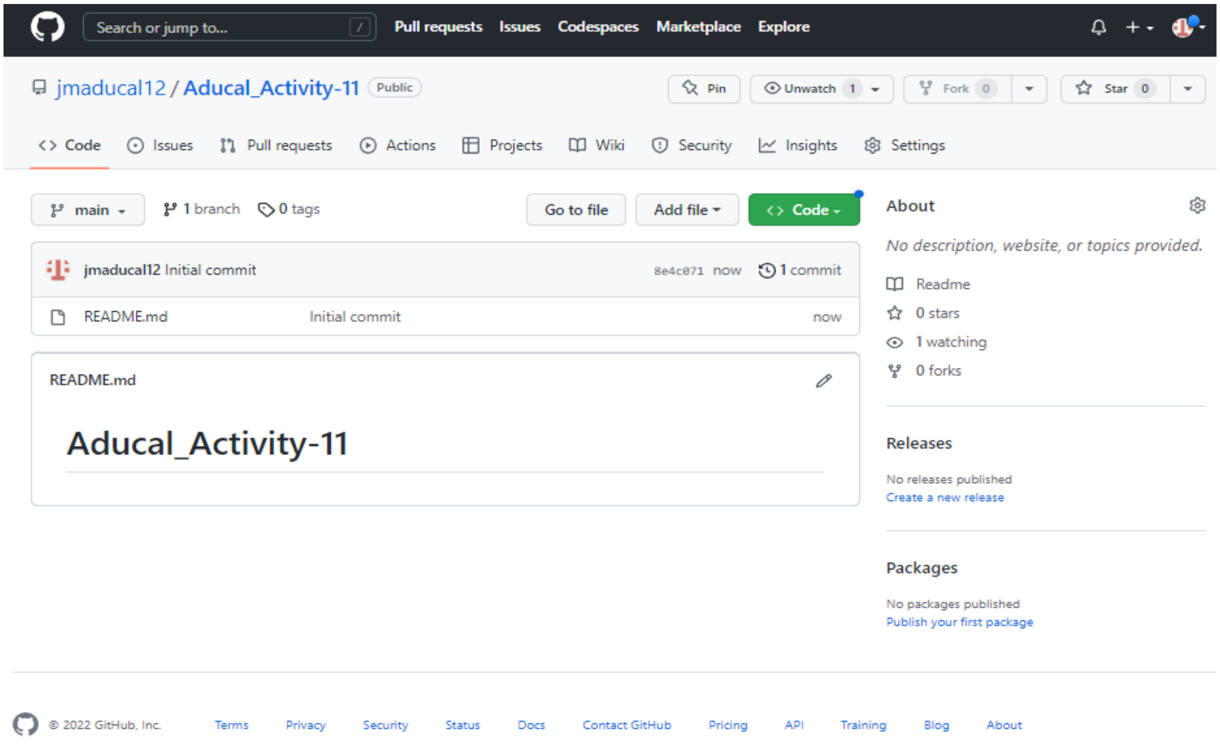
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

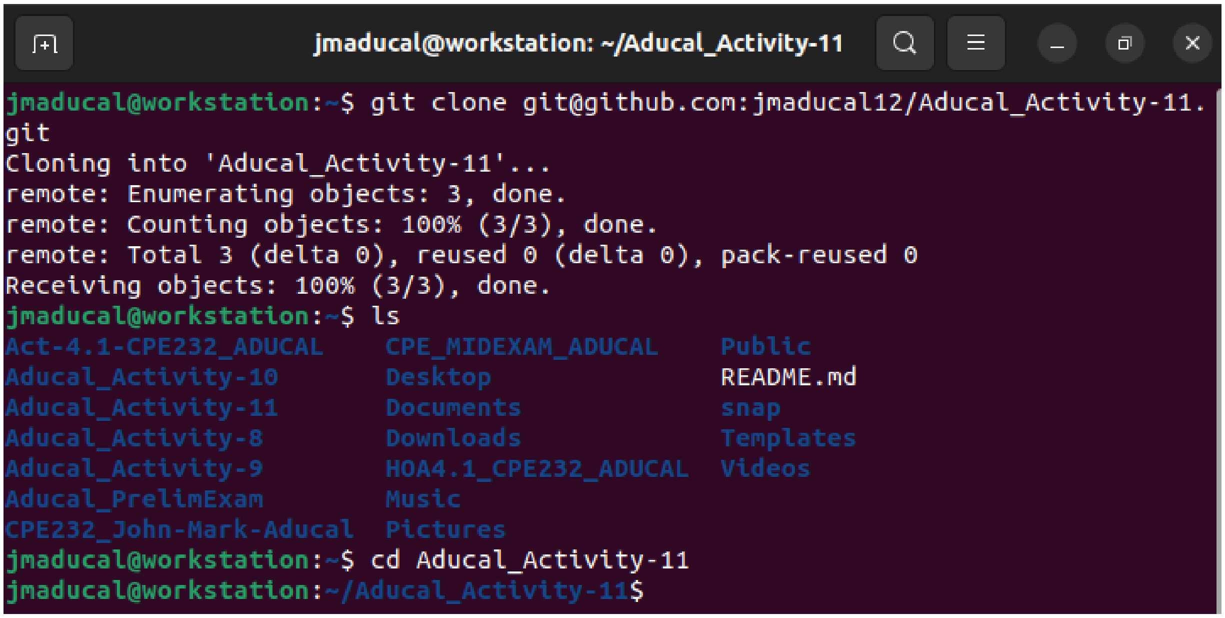
1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. **Output** (screenshots and explanations)

Task 1: Create a new repository for this activity.



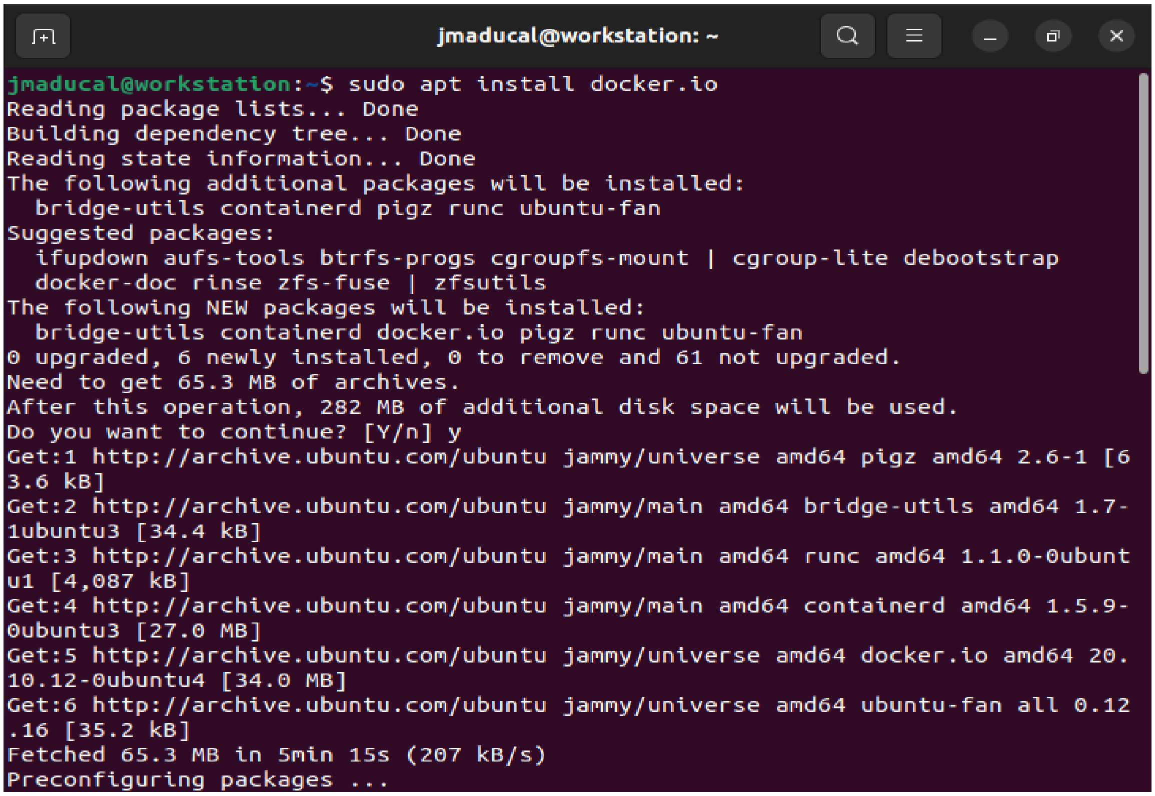
I created a new repository named Aducal_Activity-11.



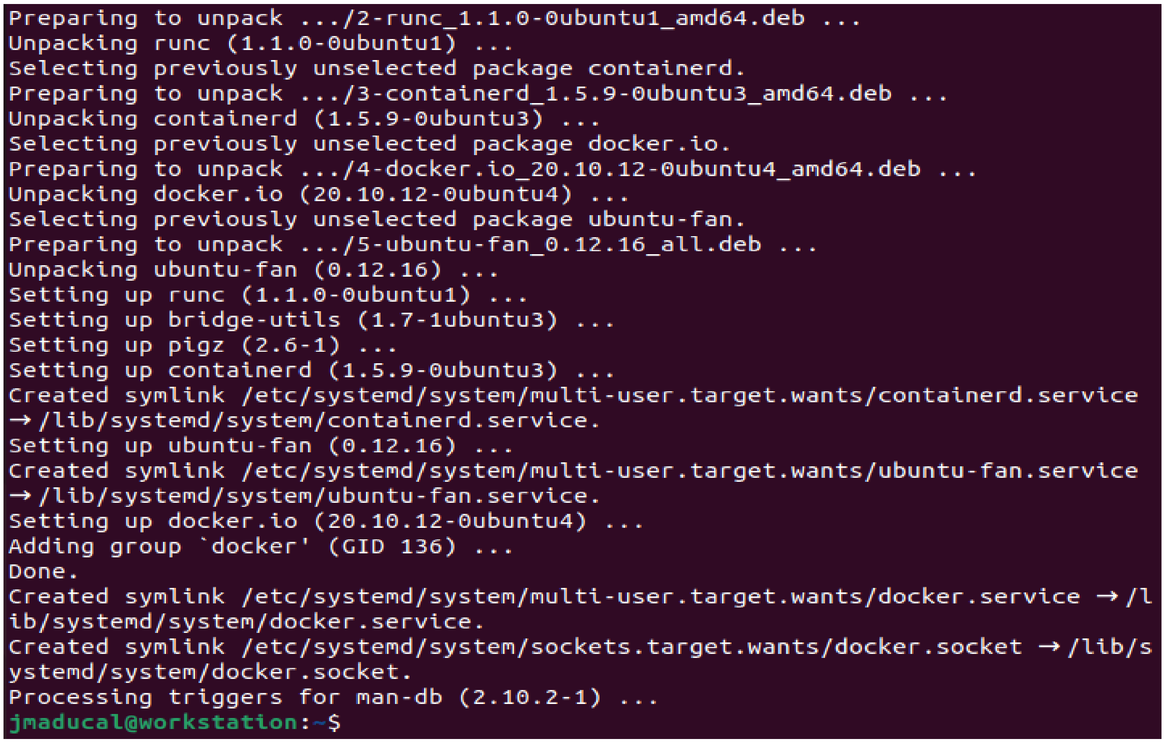
```
jmaducal@workstation:~$ git clone git@github.com:jmaducal12/Aducal_Activity-11.
git
Cloning into 'Aducal_Activity-11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
jmaducal@workstation:~$ ls
Act-4.1-CPE232_ADUCAL     CPE_MIDEXAM_ADUCAL     Public
Aducal_Activity-10        Desktop                README.md
Aducal_Activity-11        Documents              snap
Aducal_Activity-8         Downloads              Templates
Aducal_Activity-9         HOA4.1_CPE232_ADUCAL   Videos
Aducal_PrelimExam         Music
CPE232_John-Mark-Aducal   Pictures
jmaducal@workstation:~$ cd Aducal_Activity-11
jmaducal@workstation:~/Aducal_Activity-11$
```

I used git clone command to copy the new repository I have created into my workstation.Now we can use the new repository we created earlier, using cd command to change directory into Aducal_Activity-11.

Task 2: Install Docker and enable docker socket

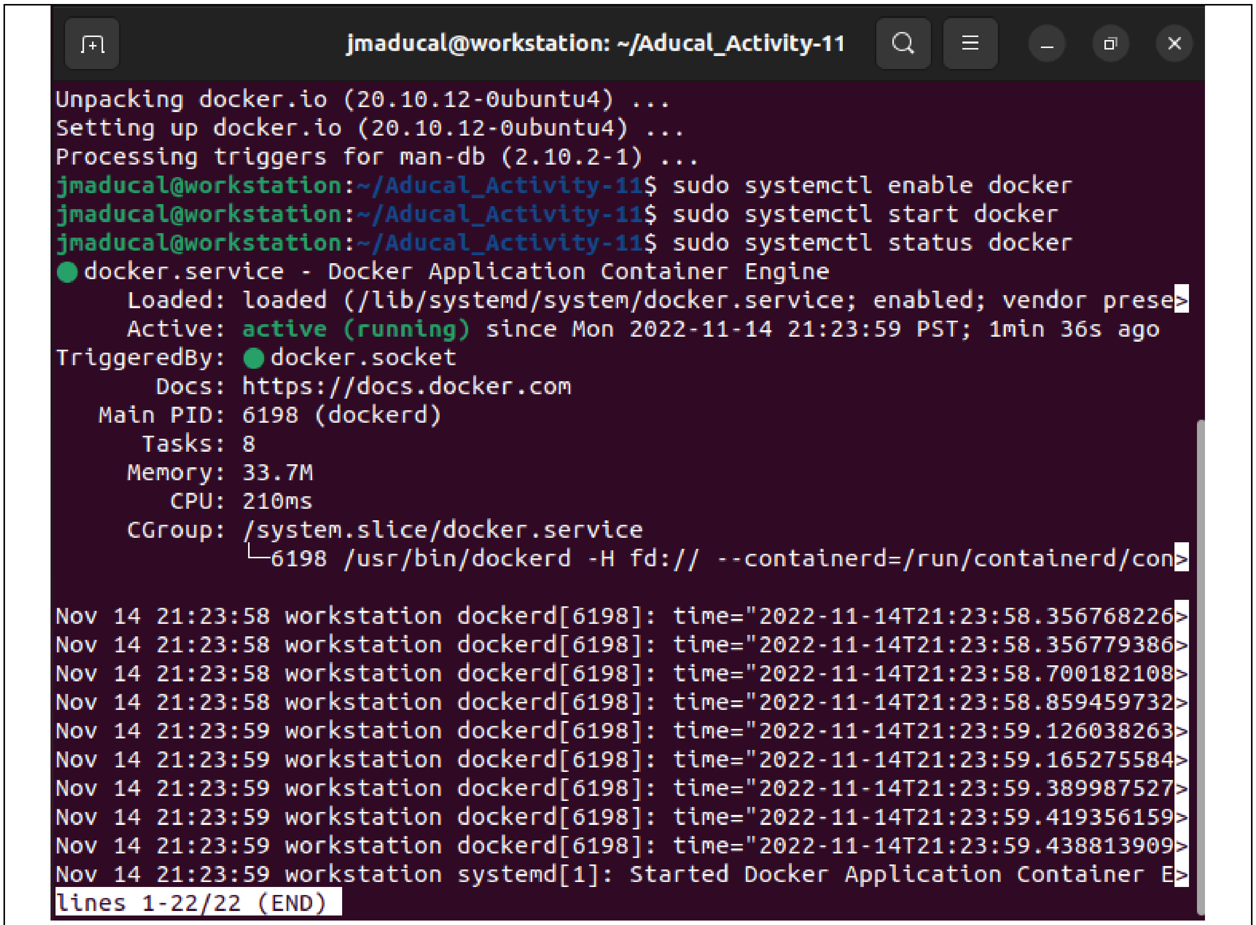


```
jmaducal@workstation:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs 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 61 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 282 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [6
3.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-
1ubuntu3 [34.4 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.0-0ubunt
u1 [4,087 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd64 1.5.9-
0ubuntu3 [27.0 MB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64 20.
10.12-0ubuntu4 [34.0 MB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12
.16 [35.2 kB]
Fetched 65.3 MB in 5min 15s (207 kB/s)
Preconfiguring packages ...
```



```
Preparing to unpack .../2-runc_1.1.0-0ubuntu1_amd64.deb ...
Unpacking runc (1.1.0-0ubuntu1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.9-0ubuntu3_amd64.deb ...
Unpacking containerd (1.5.9-0ubuntu3) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_20.10.12-0ubuntu4_amd64.deb ...
Unpacking docker.io (20.10.12-0ubuntu4) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../5-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up runc (1.1.0-0ubuntu1) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.5.9-0ubuntu3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service
→ /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service
→ /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.12-0ubuntu4) ...
Adding group `docker' (GID 136) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /l
ib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/s
ystemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
jmaducal@workstation:~$
```

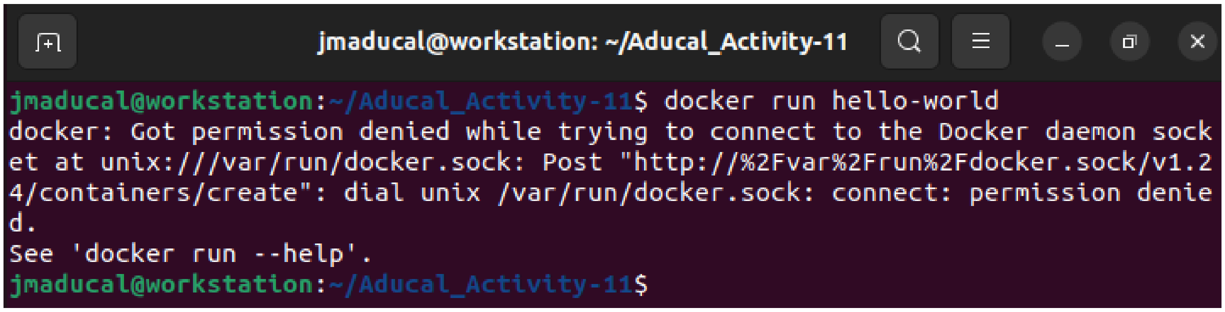
I have already installed docker in my workstation.



```
Unpacking docker.io (20.10.12-0ubuntu4) ...
Setting up docker.io (20.10.12-0ubuntu4) ...
Processing triggers for man-db (2.10.2-1) ...
jmaducal@workstation:~/Aducal_Activity-11$ sudo systemctl enable docker
jmaducal@workstation:~/Aducal_Activity-11$ sudo systemctl start docker
jmaducal@workstation:~/Aducal_Activity-11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
     Active: active (running) since Mon 2022-11-14 21:23:59 PST; 1min 36s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 6198 (dockerd)
      Tasks: 8
     Memory: 33.7M
        CPU: 210ms
     CGroup: /system.slice/docker.service
             └─6198 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>

Nov 14 21:23:58 workstation dockerd[6198]: time="2022-11-14T21:23:58.356768226>
Nov 14 21:23:58 workstation dockerd[6198]: time="2022-11-14T21:23:58.356779386>
Nov 14 21:23:58 workstation dockerd[6198]: time="2022-11-14T21:23:58.700182108>
Nov 14 21:23:58 workstation dockerd[6198]: time="2022-11-14T21:23:58.859459732>
Nov 14 21:23:59 workstation dockerd[6198]: time="2022-11-14T21:23:59.126038263>
Nov 14 21:23:59 workstation dockerd[6198]: time="2022-11-14T21:23:59.165275584>
Nov 14 21:23:59 workstation dockerd[6198]: time="2022-11-14T21:23:59.389987527>
Nov 14 21:23:59 workstation dockerd[6198]: time="2022-11-14T21:23:59.419356159>
Nov 14 21:23:59 workstation dockerd[6198]: time="2022-11-14T21:23:59.438813909>
Nov 14 21:23:59 workstation systemd[1]: Started Docker Application Container E>
lines 1-22/22 (END)
```

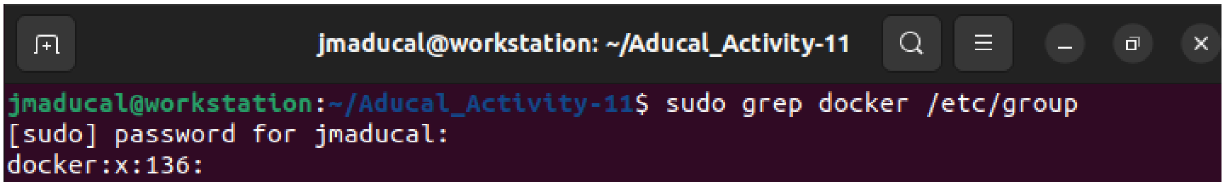
The docker service is already running at ubuntu system and docker socket was enabled.

Task 3: Add to Docker group your current user.



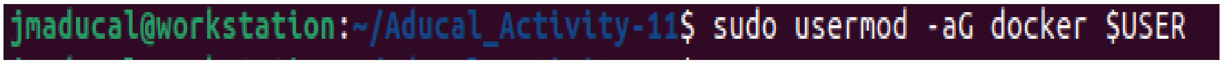
```
jmaducal@workstation:~/Aducal_Activity-11$ docker run hello-world
docker: Got permission denied while trying to connect to the Docker daemon sock
et at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.2
4/containers/create": dial unix /var/run/docker.sock: connect: permission denie
d.
See 'docker run --help'.
jmaducal@workstation:~/Aducal_Activity-11$
```

By default, docker command should run with root privilege or with sudo command. To run docker as non-root user in Ubuntu, we need to add the current user to the docker group. Otherwise, we will receive this error again.



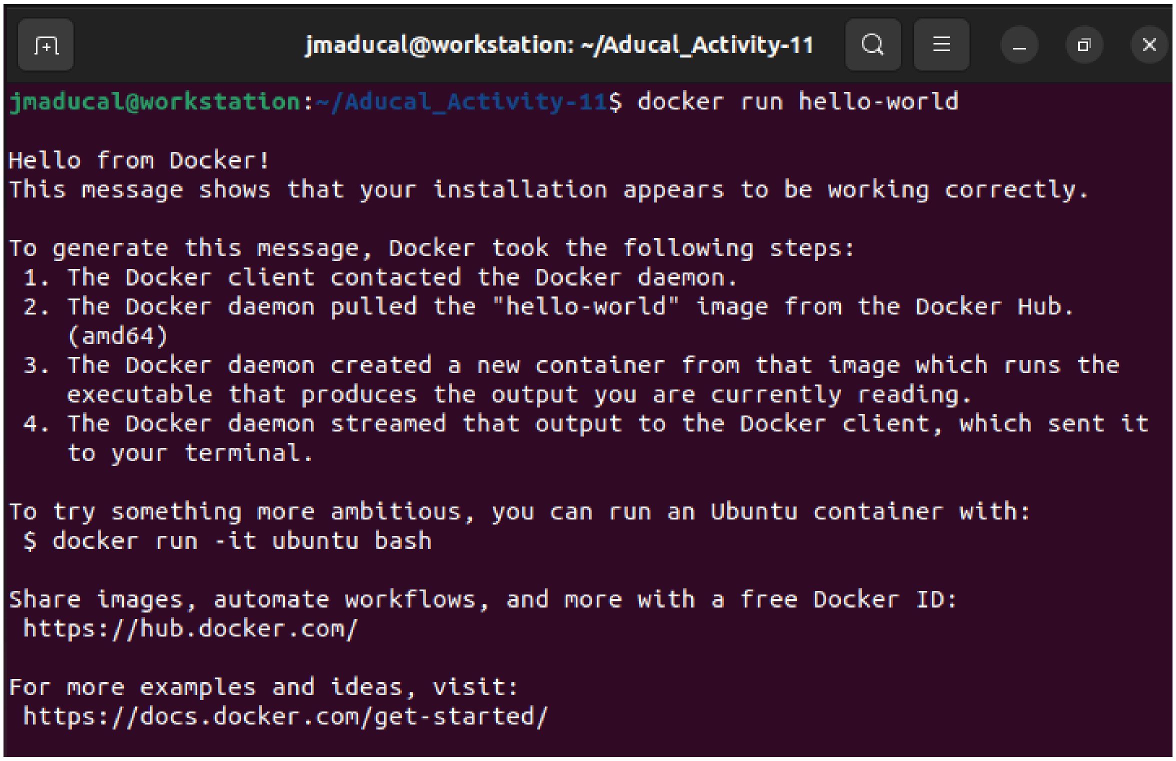
```
jmaducal@workstation:~/Aducal_Activity-11$ sudo grep docker /etc/group
[sudo] password for jmaducal:
docker:x:136:
```

First is to check if the docker group is already exists in our Ubuntu system.



```
jmaducal@workstation:~/Aducal_Activity-11$ sudo usermod -aG docker $USER
```

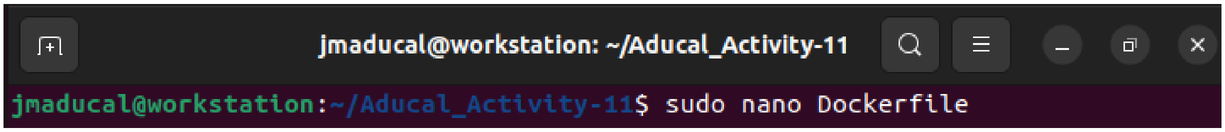
Add the current user to docker group using $USER.



```
jmaducal@workstation:~/Aducal_Activity-11$ 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/
```

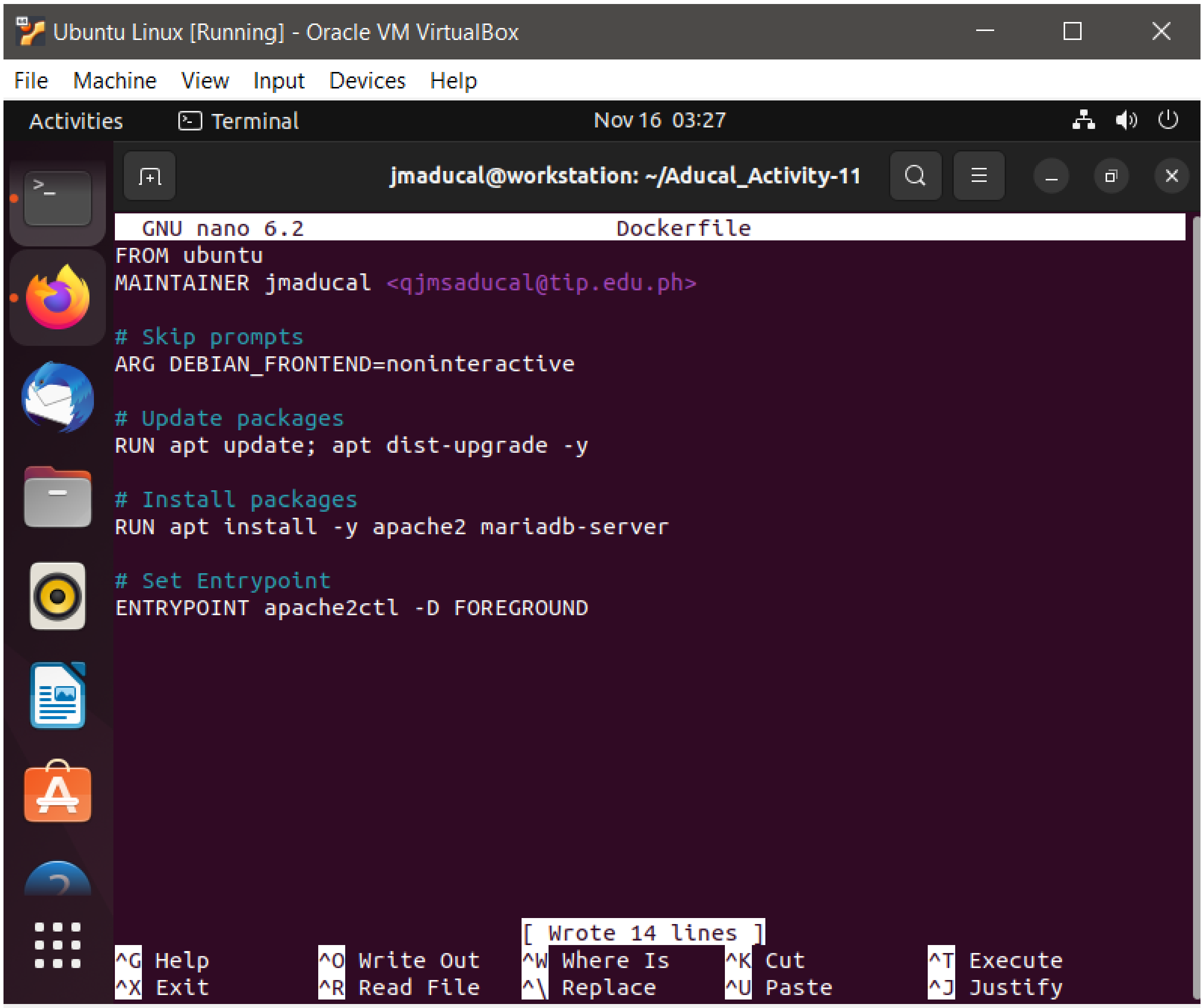
Restart your Ubuntu virtual machine for changes to take effect. Then after we can now run docker command without using sudo command or have a root privilege.

Task 4: Create a Dockerfile to Install web and Dbserver



```
jmaducal@workstation:~/Aducal_Activity-11$ sudo nano Dockerfile
```

I have created a new file named Docker file.

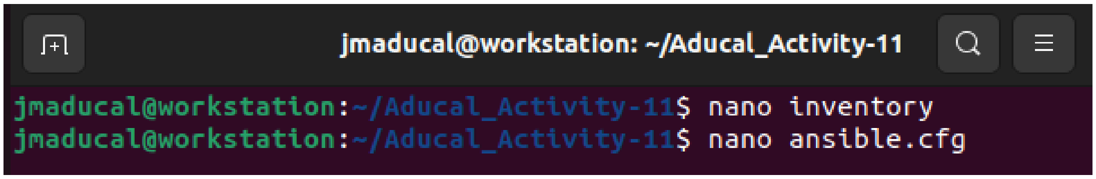


Ubuntu Linux [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Nov 16 03:27
jmaducal@workstation: ~/Aducal_Activity-11

```
  GNU nano 6.2                          Dockerfile
FROM ubuntu
MAINTAINER jmaducal <qjmsaducal@tip.edu.ph>

# Skip prompts
ARG DEBIAN_FRONTEND=noninteractive

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

# Install packages
RUN apt install -y apache2 mariadb-server

# Set Entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND




                              [ Wrote 14 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify
```

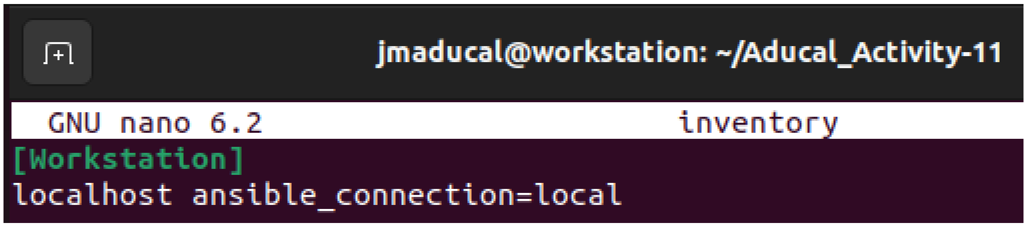
This are the contents inside of Dockerfile.

Task 5: Install and build Dockerfile using ansible



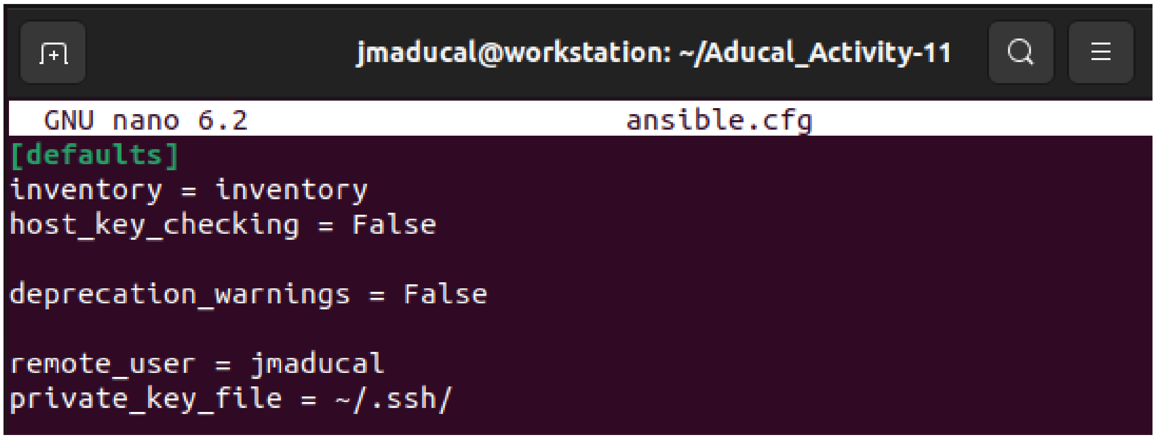
```
jmaducal@workstation:~/Aducal_Activity-11$ nano inventory
jmaducal@workstation:~/Aducal_Activity-11$ nano ansible.cfg
```

I created new inventory and ansible.cfg files.



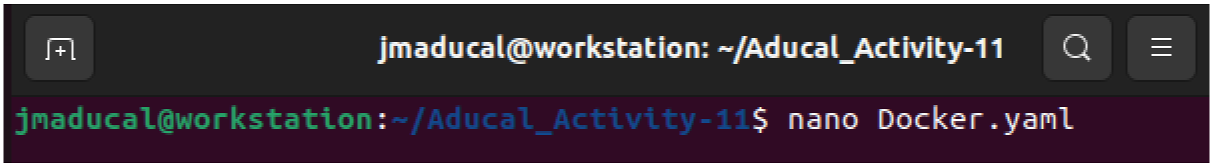
```
  GNU nano 6.2                           inventory
[Workstation]
localhost ansible_connection=local
```

The inventory file contains the local IP address of the workstation.



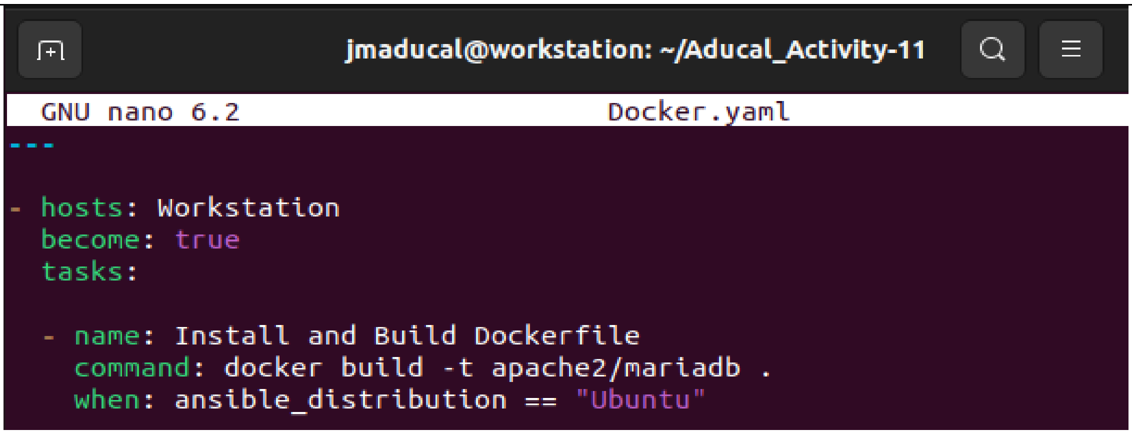
```
  GNU nano 6.2                           ansible.cfg
[defaults]
inventory = inventory
host_key_checking = False

deprecation_warnings = False

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

This are the contents of ansible.cfg file.



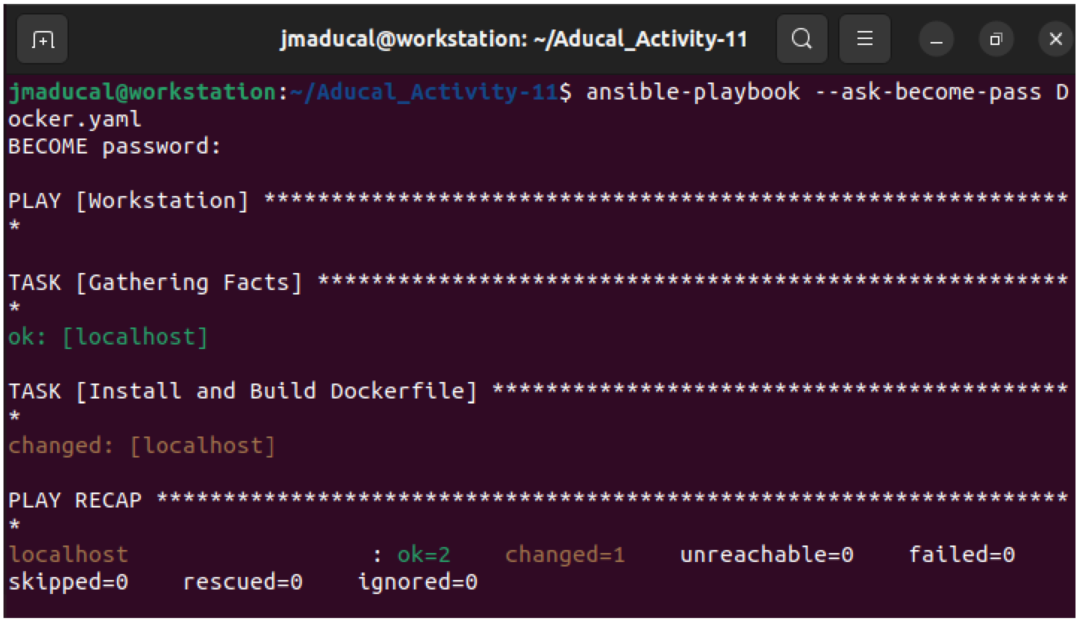
```
jmaducal@workstation:~/Aducal_Activity-11$ nano Docker.yaml
```

I created a new file name Docker.yaml



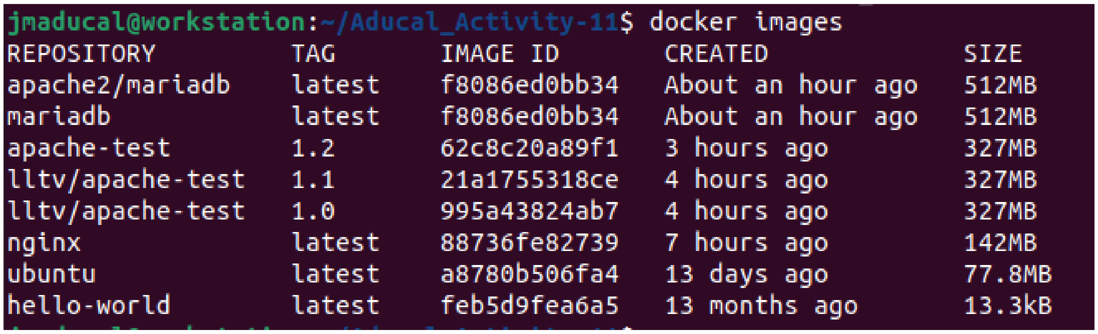
```
 GNU nano 6.2                          Docker.yaml
---

- hosts: Workstation
  become: true
  tasks:

  - name: Install and Build Dockerfile
    command: docker build -t apache2/mariadb .
    when: ansible_distribution == "Ubuntu"
```

This are the contents of Docker.yaml file.



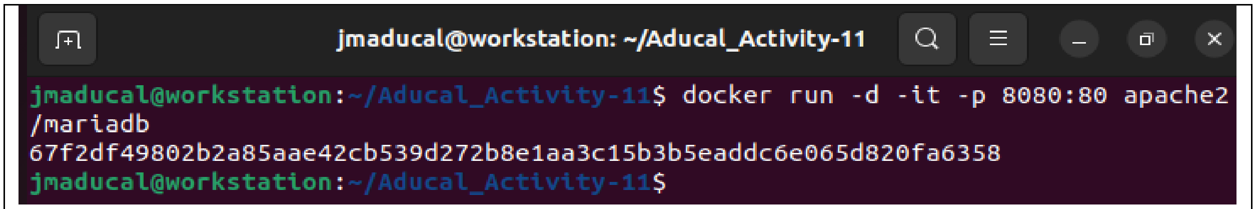
```
jmaducal@workstation:~/Aducal_Activity-11$ ansible-playbook --ask-become-pass D
ocker.yaml
BECOME password:

PLAY [Workstation] ************************************************************
*

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

TASK [Install and Build Dockerfile] *******************************************
*
changed: [localhost]

PLAY RECAP ********************************************************************
*
localhost                  : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

After executing Docker.yaml file using ansible, I'd already installed and build Dockerfile in our workstation using ansible playbook.



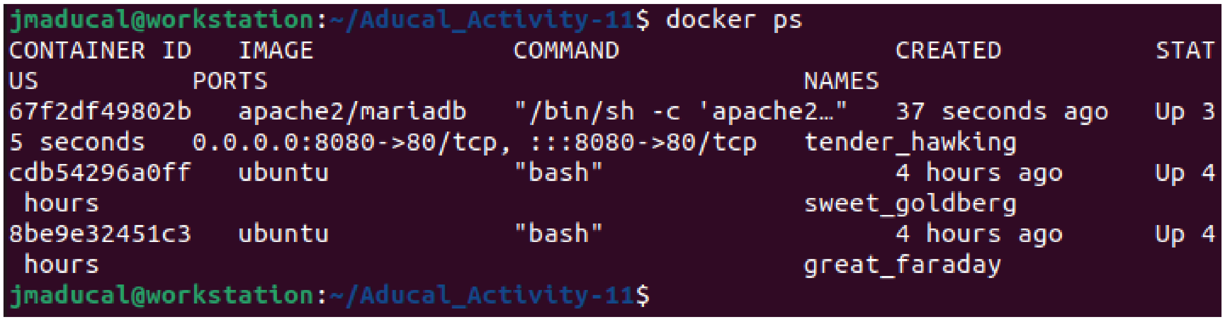
```
jmaducal@workstation:~/Aducal_Activity-11$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED            SIZE
apache2/mariadb     latest    f8086ed0bb34   About an hour ago   512MB
mariadb             latest    f8086ed0bb34   About an hour ago   512MB
apache-test         1.2       62c8c20a89f1   3 hours ago        327MB
lltv/apache-test    1.1       21a1755318ce   4 hours ago        327MB
lltv/apache-test    1.0       995a43824ab7   4 hours ago        327MB
nginx               latest    88736fe82739   7 hours ago        142MB
ubuntu              latest    a8780b506fa4   13 days ago        77.8MB
hello-world         latest    feb5d9fea6a5   13 months ago      13.3kB
```

To check if installation is successful is to run the command docker images. apache2/mariadb image was successfully built.



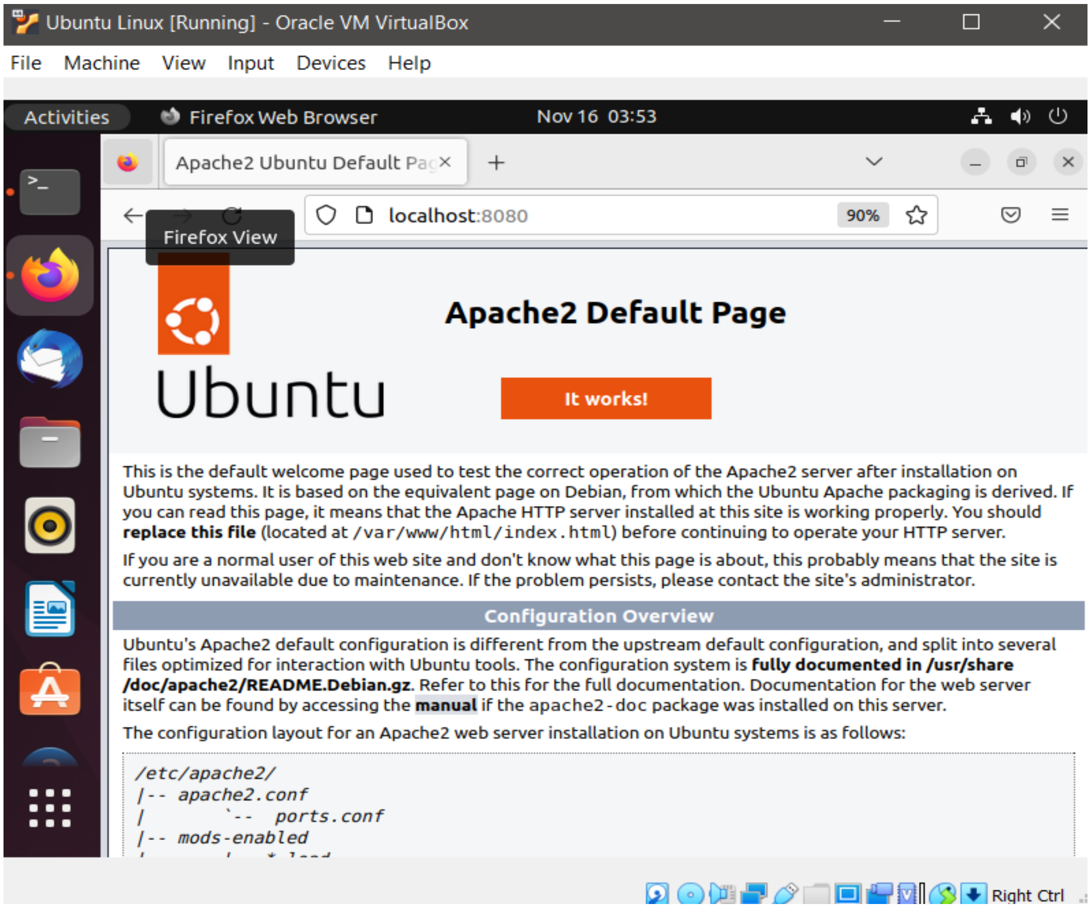
```
jmaducal@workstation:~/Aducal_Activity-11$ docker run -d -it -p 8080:80 apache2
/mariadb
67f2df49802b2a85aae42cb539d272b8e1aa3c15b3b5eaddc6e065d820fa6358
jmaducal@workstation:~/Aducal_Activity-11$
```

Next is to run the apache2/mariadb image



```
jmaducal@workstation:~/Aducal_Activity-11$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STAT
US          PORTS                                   NAMES
67f2df49802b   apache2/mariadb    "/bin/sh -c 'apache2…"   37 seconds ago   Up 3
5 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   tender_hawking
cdb54296a0ff   ubuntu             "bash"                   4 hours ago      Up 4
 hours                                              sweet_goldberg
8be9e32451c3   ubuntu             "bash"                   4 hours ago      Up 4
 hours                                              great_faraday
jmaducal@workstation:~/Aducal_Activity-11$
```

Then run the docker ps to check the containers that is running.



Apache2 Default Page

It works!

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

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

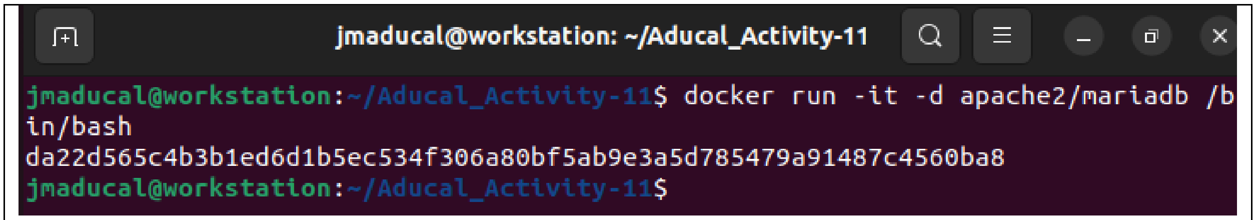
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server 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:

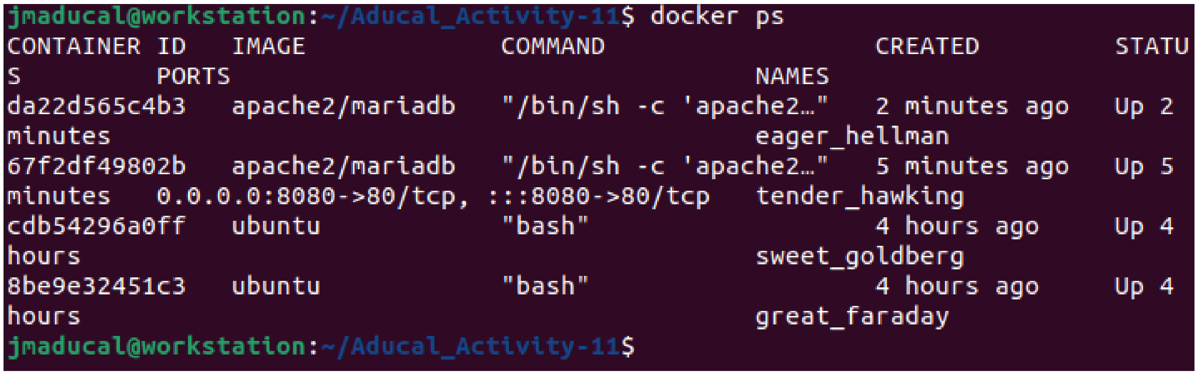
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
```

Now go to the browser and test if apache works, In my end it works already.



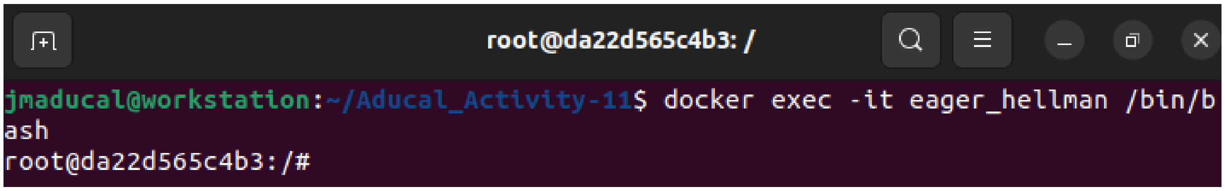
```
jmaducal@workstation:~/Aducal_Activity-11$ docker run -it -d apache2/mariadb /b
in/bash
da22d565c4b3b1ed6d1b5ec534f306a80bf5ab9e3a5d785479a91487c4560ba8
jmaducal@workstation:~/Aducal_Activity-11$
```

Next is to run the apache2/mariadb image again with /bin/bash



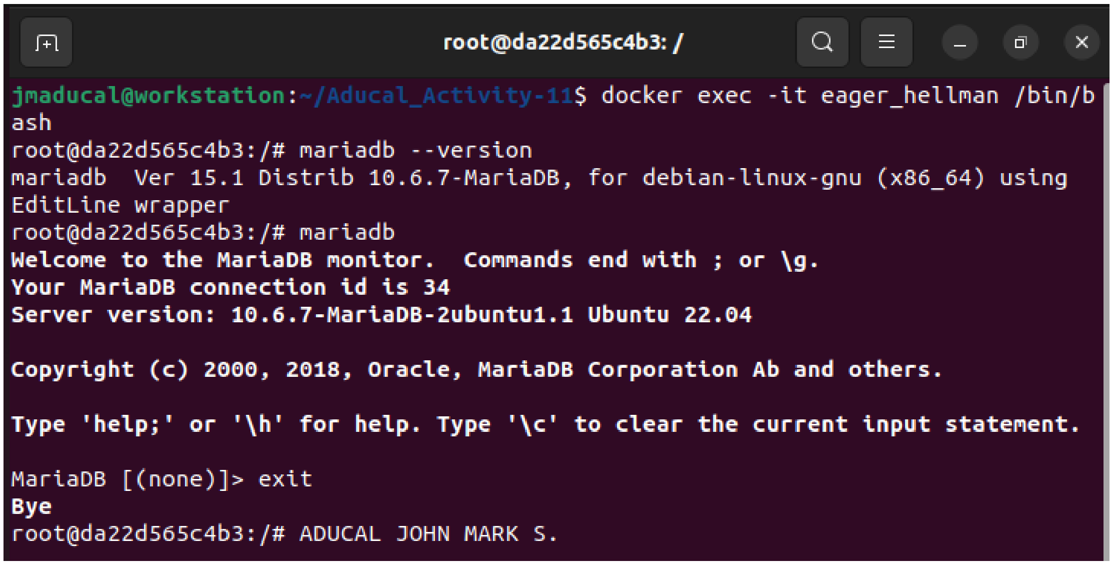
```
jmaducal@workstation:~/Aducal_Activity-11$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATU
S          PORTS                                  NAMES
da22d565c4b3   apache2/mariadb   "/bin/sh -c 'apache2…"   2 minutes ago   Up 2
minutes                                           eager_hellman
67f2df49802b   apache2/mariadb   "/bin/sh -c 'apache2…"   5 minutes ago   Up 5
minutes    0.0.0.0:8080->80/tcp, :::8080->80/tcp   tender_hawking
cdb54296a0ff   ubuntu            "bash"                   4 hours ago     Up 4
hours                                             sweet_goldberg
8be9e32451c3   ubuntu            "bash"                   4 hours ago     Up 4
hours                                             great_faraday
jmaducal@workstation:~/Aducal_Activity-11$
```

Then try to connect to the container ID da22d565c4b3, Image Name: apache2/mariadb



```
jmaducal@workstation:~/Aducal_Activity-11$ docker exec -it eager_hellman /bin/b
ash
root@da22d565c4b3:/#
```

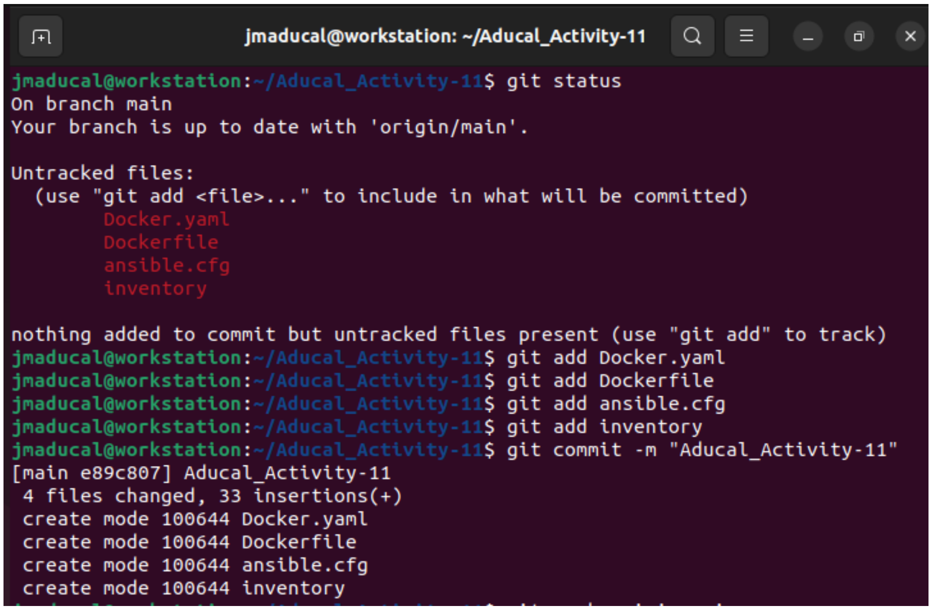
Now we are connected to the container name eager_hellman (apache2/mariadb image)



```
jmaducal@workstation:~/Aducal_Activity-11$ docker exec -it eager_hellman /bin/b
ash
root@da22d565c4b3:/# mariadb --version
mariadb  Ver 15.1 Distrib 10.6.7-MariaDB, for debian-linux-gnu (x86_64) using
EditLine wrapper
root@da22d565c4b3:/# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 10.6.7-MariaDB-2ubuntu1.1 Ubuntu 22.04

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)]> exit
Bye
root@da22d565c4b3:/# ADUCAL JOHN MARK S.
```

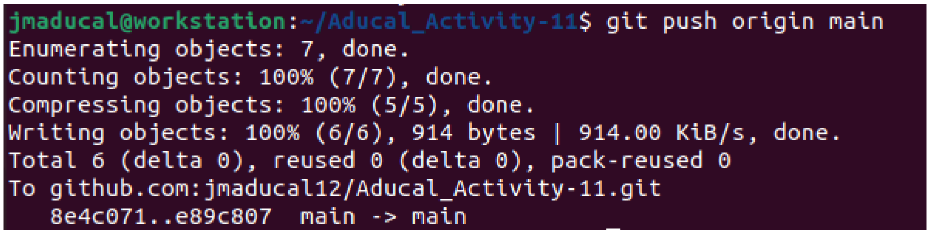
Lastly is to verify whether we successfully installed the mariadb in apache2/mariadb image and this container.

Task 6: Add, Commit and Push it to your repository

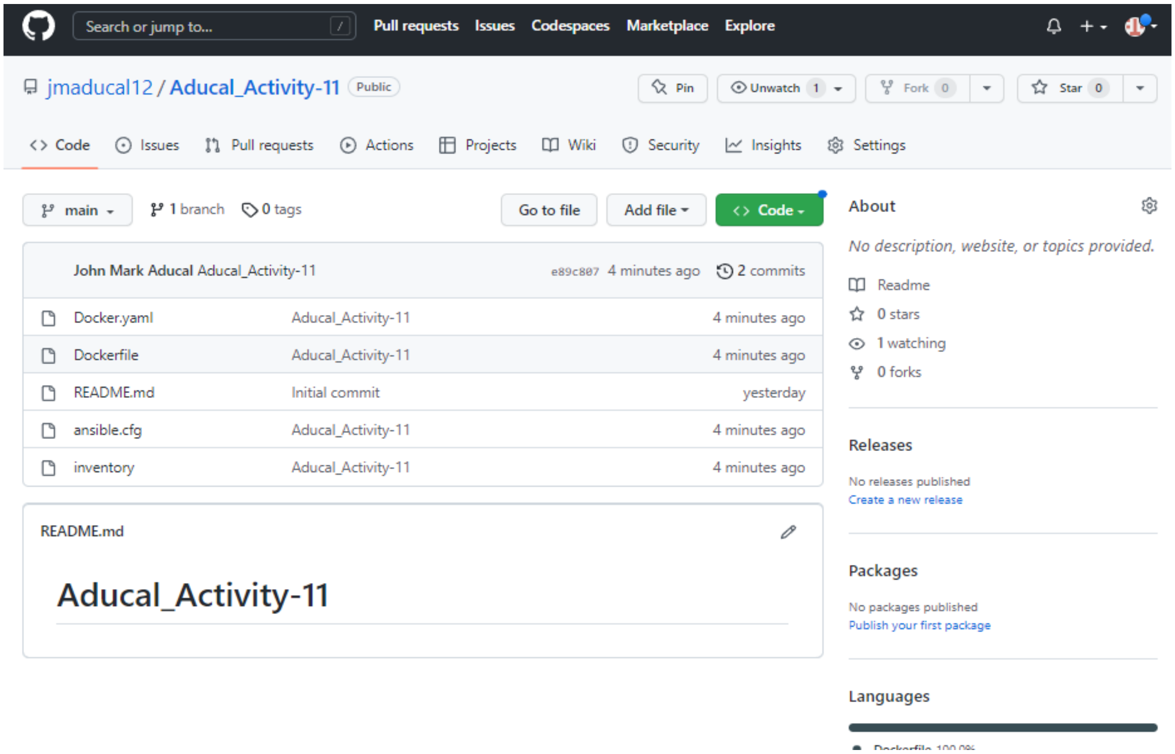


jmaducal@workstation: ~/Aducal_Activity-11

```
jmaducal@workstation:~/Aducal_Activity-11$ 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)
        Docker.yaml
        Dockerfile
        ansible.cfg
        inventory

nothing added to commit but untracked files present (use "git add" to track)
jmaducal@workstation:~/Aducal_Activity-11$ git add Docker.yaml
jmaducal@workstation:~/Aducal_Activity-11$ git add Dockerfile
jmaducal@workstation:~/Aducal_Activity-11$ git add ansible.cfg
jmaducal@workstation:~/Aducal_Activity-11$ git add inventory
jmaducal@workstation:~/Aducal_Activity-11$ git commit -m "Aducal_Activity-11"
[main e89c807] Aducal_Activity-11
 4 files changed, 33 insertions(+)
 create mode 100644 Docker.yaml
 create mode 100644 Dockerfile
 create mode 100644 ansible.cfg
 create mode 100644 inventory
```



```
jmaducal@workstation:~/Aducal_Activity-11$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 914 bytes | 914.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jmaducal12/Aducal_Activity-11.git
   8e4c071..e89c807  main -> main
```



jmaducal12 / Aducal_Activity-11 Public

John Mark Aducal Aducal_Activity-11 e89c807 4 minutes ago 2 commits

Docker.yaml	Aducal_Activity-11	4 minutes ago
Dockerfile	Aducal_Activity-11	4 minutes ago
README.md	Initial commit	yesterday
ansible.cfg	Aducal_Activity-11	4 minutes ago
inventory	Aducal_Activity-11	4 minutes ago

README.md

Aducal_Activity-11

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Dockerfile 100.0%

GitHub Repository Link:

https://github.com/jmaducal12/Aducal_Activity-11.git

Reflections:

Answer the following:

1. **What are the benefits of implementing containerizations?** Docker Containers can be easily copied, deployed and can run almost anywhere. Implementing containerizations can be often cheaper to run than virtual machines, Docker Containers can be run on cloud platforms such as Amazon Web Services, Linode, Digital Ocean, Google Cloud and others.

Conclusions:

From this activity, I learned what is use of docker and how we used to manage our applications and deliver software quickly by building a dockerfile and a docker image inside a container for a quicker deployment, the simplicity of setting up new instances, and a quicker migration of apps and files. Simple management and moving of our applications. Less access is required to work with the code running inside the container and have better security.

John Mark Aducal