

# In-class Assignment 2

Instructor: Qasim Ali

## Develop and Deploy a Machine Learning Application using Docker

Group Name: C

Student Name: Ram Krishna Dhakal

### **Objective**

The objective of this assignment is to develop a simple machine learning application, containerize it using Docker, and deploy it on GitHub. This exercise will help you understand the principles of containerization, version control, and basic machine learning application development.

### **Prerequisites**

- Basic understanding of Python programming
- Basic understanding of machine learning concepts
- Familiarity with Git and GitHub
- Basic knowledge of Docker

### **Assignment Steps**

#### **Step 1: Set Up the VM**

##### **1. Update the System**

- Ensure your VM is running an updated version of Ubuntu. Run the following commands:

```
sudo apt update sudo apt upgrade -y
```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth... [x]

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth... [x]

SSH-in-browser

UPLOAD FILE DOWNLOAD FILE ! 📁 🚪

Unpacking ubuntu-pro-client-110... (34-20.04) over (33.2-20.04) ...  
Preparing to unpack .../10-ubuntu-pro-client\_34-20.04\_amd64.deb ...  
Unpacking ubuntu-pro-client (34-20.04) over (33.2-20.04) ...  
Preparing to unpack .../11-ubuntu-adantage-tools\_34-20.04\_all.deb ...  
Unpacking ubuntu-adantage-tools (34-20.04) over (33.2-20.04) ...  
Preparing to unpack .../12-vim\_238a8.1.2269~lubuntu5.25\_amd64.deb ...  
Preparing vim (2:8.1.2269~lubuntu5.25) over (2:8.1.2269~lubuntu5.23) ...  
Preparing to unpack .../13-vim-tiny\_238a8.1.2269~lubuntu5.25\_amd64.deb ...  
Unpacking vim-tiny (2:8.1.2269~lubuntu5.25) over (2:8.1.2269~lubuntu5.23) ...  
Preparing to unpack .../14-vim-runtime\_238a8.1.2269~lubuntu5.25\_all.deb ...  
Unpacking vim-runtime (2:8.1.2269~lubuntu5.25) over (2:8.1.2269~lubuntu5.23) ...  
Preparing to unpack .../15-xxd\_238a8.1.2269~lubuntu5.25\_amd64.deb ...  
Unpacking xxd (2:8.1.2269~lubuntu5.25) over (2:8.1.2269~lubuntu5.23) ...  
Preparing to unpack .../16-vim-common\_238a8.1.2269~lubuntu5.25\_all.deb ...  
Unpacking vim-common (2:8.1.2269~lubuntu5.25) over (2:8.1.2269~lubuntu5.23) ...  
Preparing to unpack .../17-aparmor\_2.13.3~7ubuntu5.4\_amd64.deb ...  
Unpacking aparmor (2.13.3~7ubuntu5.4) over (2.13.3~7ubuntu5.3) ...  
Preparing to unpack .../18-libpcap0.8\_amd64 (1.9.1~3ubuntu1.20.04.1\_amd64.deb ...  
Unpacking libpcap0.8\_amd64 (1.9.1~3ubuntu1.20.04.1) over (1.9.1~3) ...  
Preparing to unpack .../19-python3-update-manager\_1%3a20.04.10.23\_all.deb ...  
Unpacking python3-update-manager (1:20.04.10.23) over (1:20.04.10.21) ...  
Preparing to unpack .../20-update-manager-core\_1%3a20.04.10.23\_all.deb ...  
Unpacking update-manager-core (1:20.04.10.23) over (1:20.04.10.21) ...  
Preparing to unpack .../21-curl\_7.68.0~lubuntu2.24\_amd64.deb ...  
Unpacking curl (7.68.0~lubuntu2.24) over (7.68.0~lubuntu2.23) ...  
Preparing to unpack .../22-libcurl4\_7.68.0~lubuntu2.24\_amd64.deb ...  
Unpacking libcurl4\_amd64 (7.68.0~lubuntu2.24) over (7.68.0~lubuntu2.23) ...  
Preparing to unpack .../23-libcurl3-gnutls\_7.68.0~lubuntu2.24\_amd64.deb ...  
Unpacking libcurl3-gnutls\_amd64 (7.68.0~lubuntu2.24) over (7.68.0~lubuntu2.23) ...  
Selecting previously unselected package linux-modules-5.15.0-1069-gcp.  
Preparing to unpack .../24-linux-modules-5.15.0-1069-gcp\_5.15.0-1069.77-20.04.1\_amd64.deb ...  
Unpacking linux-modules-5.15.0-1069-gcp (5.15.0-1069.77-20.04.1) ...  
[Progress: 100% [███████████]]

File Home Insert Layout References Share

Deployment of AI\_Docker\_Group\_C\_Ram Krishna Dhakal (... Ram Krishna Dhakal

1. Update the System

- Ensure your VM is running an updated version of Ubuntu. Run the following commands:  
`sudo apt update` `sudo apt upgrade -y`

2. Install Necessary Packages

- Install curl and git:  
`sudo apt install -y curl git`

Step 2: Install Docker

Remove Old Versions

Page 1 of 6 834 words English (U.S.) 100% Give Feedback to Microsoft

9°C Clear

Search

12:48 AM 2024-10-03

## 2. Install Necessary Packages

- Install curl and git:

```
sudo apt install -y curl git
```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth... └─ ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth...

SSH-in-browser    UPLOAD FILE    DOWNLOAD FILE    ⚙️

```
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for ca-certificates (20240203-20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-1069-gcp
Processing triggers for linux-image-5.15.0-1069-gcp (5.15.0-1069.77~20.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-1069-gcp
/etc/kernel/postinst.d/dzz-upgrade-grub:
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file '/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-5.15.0-1069-gcp
Found initrd image: /boot/initrd.img-5.15.0-1069-gcp
Found linux image: /boot/vmlinuz-5.15.0-1067-gcp
Found initrd image: /boot/initrd.img-5.15.0-1067-gcp
Found Ubuntu 20.04.6 LTS (20.04) on /dev/sdal
Adding boot menu entry for UEFI Firmware Settings
done
drakrishna19@drakrishna19:~$ sudo apt install -y curl git
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.24).
curl set to manually installed.
git is already the newest version (1:2.25.1-1ubuntu3.13).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
drakrishna19@drakrishna19:~$ [ ]
```

Deployment\_of\_AI\_Docker\_Group\_C\_Ram\_Krishna\_Dhakal ( ... )

File Home Insert Layout References

Share

2. Install Necessary Packages

- Install curl and git:

```
sudo apt install -y curl git
```

Step 2: Install Docker

3. Remove Old Versions

- Remove any old versions of Docker if present:

```
sudo apt remove docker docker-engine docker.io containerd runc
```

4. Set Up the Docker Repository

- Run the following commands to set up the Docker repository:

```
sudo apt update
```

```
sudo apt install -y apt-transport-https ca-certificates
```

Page 2 of 7 834 words English (U.S.) 100% Give Feedback to Microsoft

from McCowan... Closed road

Search

12:51 AM 2024-10-03

## Step 2: Install Docker

### 3. Remove Old Versions

- Remove any old versions of Docker if present:

```
sudo apt remove docker docker-engine docker.io containerd runc
```

The left window shows the terminal output of the command:

```
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth... - ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth...
SSH-in-browser UPLOAD FILE DOWNLOAD FILE
done.
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-1067-gcp
Processing triggers for linux-image-5.15.0-1069-gcp (5.15.0-1069.77~20.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-1069-gcp
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file '/etc/default/grub'.
Sourcing file '/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file '/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-5.15.0-1069-gcp
Found initrd image: /boot/initrd.img-5.15.0-1069-gcp
Found linux image: /boot/vmlinuz-5.15.0-1067-gcp
Found initrd image: /boot/initrd.img-5.15.0-1067-gcp
Found Ubuntu 20.04.6 LTS (20.04) on /dev/sdal
Adding boot menu entry for UEFI Firmware Settings
done
drakkrishna19@docker:~$ sudo apt install -y curl git
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.24).
curl set to manually installed.
git is already the newest version (1:2.25.1-1ubuntu3.13).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
drakkrishna19@docker:~$ sudo apt remove docker docker-engine docker.io containerd runc
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package docker-engine
drakkrishna19@docker:~$
```

The right window shows the Microsoft Word document content:

**Step 2: Install Docker**

**3. Remove Old Versions**

- Remove any old versions of Docker if present:

```
sudo apt remove docker docker-engine docker.io containerd runc
```

**4. Set Up the Docker Repository**

- Run the following commands to set up the Docker repository:

```
sudo apt update
sudo apt install -y apt-transport-https ca-certificates
curl [REDACTED] -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

### 4. Set Up the Docker Repository

- Run the following commands to set up the Docker repository:

```
sudo apt update
```

```
sudo apt install -y apt-transport-https ca-certificates
```

```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth...
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/instances/docker?auth...
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package docker-engine
dramkrishna19@docker:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dramkrishna19@docker:~$ sudo apt install -y apt-transport-https ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20240203-20.04.1).
ca-certificates is set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1704 B of archives.
After this operation, 162 kB of additional disk space will be used.
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-tr
apt-transport-https all 2.0.10 [1704 B]
Fetched 1704 B in 0s (75.0 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 91713 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.10_all.deb ...
Unpacking apt-transport-https (2.0.10) ...
Setting up apt-transport-https (2.0.10) ...
dramkrishna19@docker:~$ 

```

File Home Insert Layout References Share ...

Deployment\_of\_AI\_Docker\_Group\_C\_Ram\_Krishna\_Dhakal ( ... )

4. Set Up The Docker Repository

- Run the following commands to set up the Docker repository:

```

sudo apt update
sudo apt install -y apt-transport-https ca-certificates
curl gnupg lsb-release curl -fsSL https://download.docker.com/linux/ubuntu/gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

5. Install Docker Engine

- Install Docker Engine using the following command:

curl gnupg lsb-release curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

```

apt install docker.io # version 24.0.7-0ubuntu2-20.04.1
See 'snap info docker' for additional versions.

dramkrishna19@docker:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 128 kB in 1s (158 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dramkrishna19@docker:~$ sudo apt install -y apt-transport-https ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20240203-20.04.1).
apt-transport-https is already the newest version (20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
dramkrishna19@docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gp
q | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N)
N
Enter new filename:
gpg: signal Interrupt caught ... exiting
dramkrishna19@docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gp
g | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N)
y
dramkrishna19@docker:~$ 

```

File Home Insert Layout References Review View Help Share ...

Deployment\_of\_AI\_Docker\_Group\_C\_Ram\_Krishna\_Dhakal ( ... )

4. Set Up The Docker Repository

- Run the following commands to set up the Docker repository:

```

sudo apt update
curl gnupg lsb-release curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

5. Install Docker Engine

- Install Docker Engine using the following command:

```

sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io

```

echo "deb [arch=\$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] <https://download.docker.com/linux/ubuntu> \$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

dramkrishna19@docker:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 128 kB in 1s (158 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dramkrishna19@docker:~$ sudo apt install -y apt-transport-https ca-certificates
3
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20240203-20.04.1).
apt-transport-https is already the newest version (2.0.10).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
dramkrishna19@docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N)
N
Enter new filename:
gpg: signal Interrupt caught ... exiting

dramkrishna19@docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N)
y
dramkrishna19@docker:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
dramkrishna19@docker:~$ 

```

**echo "deb [arch=\$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

**5. Install Docker Engine**

- o Install Docker Engine using the following command:

```

sudo apt update sudo apt install -y docker-ce docker-ce-cli containerd.io

```

**6. Verify Docker Installation**

**sudo apt update sudo apt install -y docker-ce docker-ce-cli containerd.io**

```

deb ...
Unpacking docker-ce-cli (5:27.3.1-1~ubuntu.20.04-focal) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.3.1-1~ubuntu.20.04-focal_amd64.deb ...
Unpacking docker-ce (5:27.3.1-1~ubuntu.20.04-focal) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.3.1-1~ubuntu.20.04-focal_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.3.1-1~ubuntu.20.04-focal) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.7-1~ubuntu.20.04-focal_amd64.deb ...
Unpacking docker-compose-plugin (2.29.7-1~ubuntu.20.04-focal) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../7-slirp4netns_0.4.3-1_amd64.deb ...
Unpacking slirp4netns (0.4.3-1) ...
Setting up slirp4netns (0.4.3-1) ...
Setting up docker-buildx-plugin (0.17.1-1~ubuntu.20.04-focal) ...
Setting up containerd.io (1.22.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.20.04-focal) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.20.04-focal) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.20.04-focal) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.20.04-focal) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.24) ...
dramkrishna19@docker:~$ 

```

**6. Verify Docker Installation**

- o Verify that Docker is installed correctly by running:

```

sudo docker run hello-world

```

**Step 3: Create a Dockerfile for the ML Application**

**7. Create Project Directory**

- o Create a directory for your project and navigate into it:

```

mkdir ml-app cd ml-app

```

**8. Create a Dockerfile**

- o Create a Dockerfile with the following content:

```

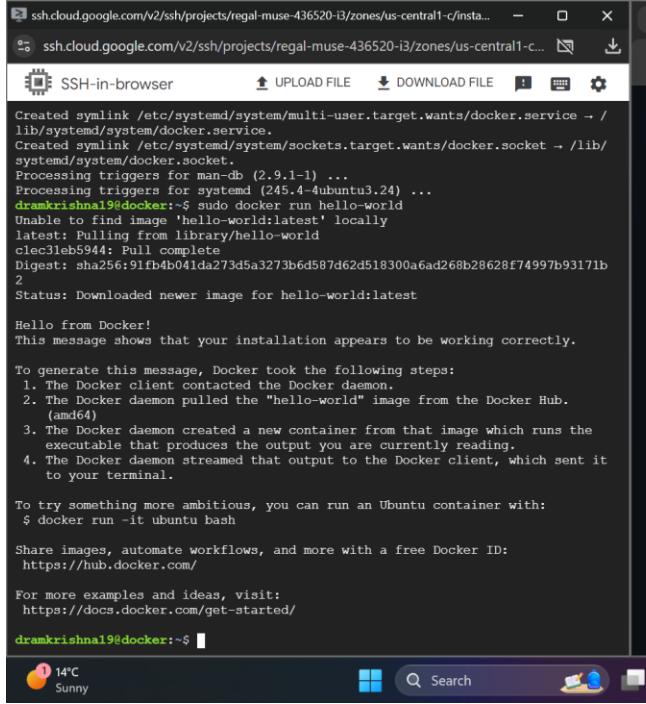
# Use an official Python runtime as a parent image
FROM python:3.9-slim
# Set the working directory
WORKDIR /usr/src/app
# Copy the current directory contents into the container at /usr/src/app

```

## 6. Verify Docker Installation

- Verify that Docker is installed correctly by running:

sudo docker run hello-world



The terminal window shows the command being run and its output. It includes Docker daemon logs, the Dockerfile content, and the final "Hello from Docker!" message.

```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.24) ...
drakmkrishna1@drak:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
clec31eb5944: Pull complete
Digest: sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b
2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

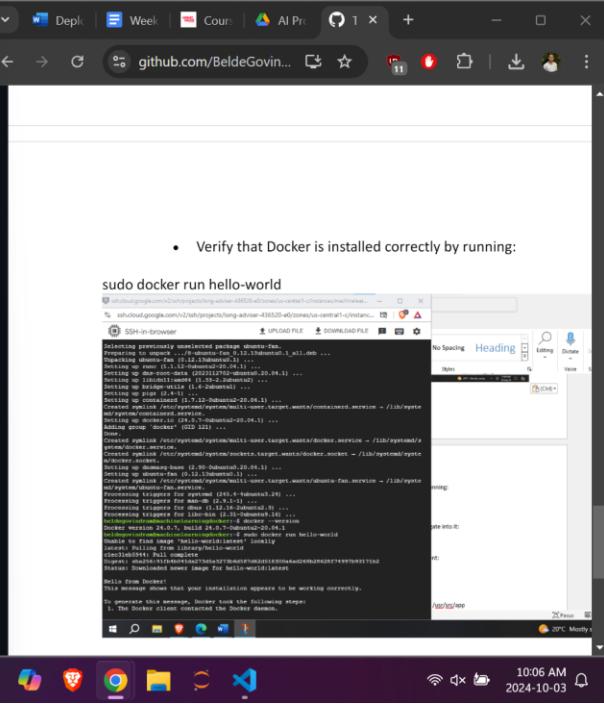
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the 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/

drakmkrishna1@drak:~$
```



The browser window displays the GitHub repository page for the project, showing the README and other repository details.

## Step 3: Create a Dockerfile for the ML Application

### 7. Create Project Directory

- Create a directory for your project and navigate into it:

```
mkdir ml-app cd ml-app
```

```

dramkrishna19@docker:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
Digest: sha256:91fb4b04da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b
2
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/

dramkrishna19@docker:~$ mkdir ml-app
dramkrishna19@docker:~$ cd myapp
-bash: cd: myapp: No such file or directory
dramkrishna19@docker:~$ mkdir ml-app
mkdir: cannot create directory 'ml-app': File exists
dramkrishna19@docker:~$ cd ml-app
dramkrishna19@docker:~/ml-app$ 
```

**Step 3: Create a Dockerfile for the ML Application**

- Create Project Directory**
  - Create a directory for your project and navigate into it:

**8. Create a Dockerfile**

- Create a **Dockerfile** with the following content:

```

# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory
WORKDIR /usr/src/app

# Copy the current directory contents into the container at /usr/src/app
*****YOU NEED TO WRITE COMMAND HERE*****

# Install any needed packages specified in requirements.txt RUN pip install --no-cache-dir -r requirements.txt # Make port 80 available to the world outside this container EXPOSE 80

# Run app.py when the container launches CMD ["python", "app.py"] 
```

## 8. Create a Dockerfile

- Create a **Dockerfile** with the following content:

```

# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory
WORKDIR /usr/src/app

# Copy the current directory contents into the container at /usr/src/app
*****YOU NEED TO WRITE COMMAND HERE*****


# Install any needed packages specified in requirements.txt RUN pip install --no-cache-dir -r requirements.txt # Make port 80 available to the world outside this container EXPOSE 80

# Run app.py when the container launches CMD ["python", "app.py"] 
```

```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/insta... - □ ×
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/insta... □

SSH-in-browser UPLOAD FILE DOWNLOAD FILE ! ⚙️

E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
draskrishna19@docker:~/ml-app$ fuse an official python runtime as parent image

Command 'fuse' not found, but can be installed with:

apt install fuse-emulator-gtk # version 1.5.7+dfsg1-3, or
apt install fuse-emulator-sdl # version 1.5.7+dfsg1-3

Ask your administrator to install one of them.

draskrishna19@docker:~/ml-app$ FROM python:3.9-slim
FROM: command not found
draskrishna19@docker:~/ml-app$ nano Dockerfile
draskrishna19@docker:~/ml-app$ cat Dockerfile
# Use an official Python runtime as a parent image

FROM python:3.9-slim
# Set the working directory
WORKDIR /usr/src/app
# Copy the current directory contents into the container
COPY . .
#Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
#Make port 80 available to the world outside this container
EXPOSE 80
#Run app.py when the container launches
CMD ["python","app.py"]
draskrishna19@docker:~/ml-app$ 

# Use an official Python runtime as a parent image
FROM python:3.9-slim
# Set the working directory
WORKDIR /usr/src/app
# Copy the current directory contents into the container
COPY . .
# Copy the current directory contents into the container at /usr/src/app
# Copy the current directory contents into the container at /usr/src/app
*****YOU NEED TO WRITE COMMAND HERE****

# Install any needed packages specified in requirements.txt # RUN pip install --no-cache-dir -r requirements.txt # Make port 80 available to the world outside this container EXPOSE 80

# Run app.py when the container launches CMD ["python", "app.py"]

9. Create requirements.txt File
○ Create a requirements.txt file with the following content:
Flask
Numpy
Pandas

```

## 9. Create requirements.txt File

- Create a **requirements.txt** file with the following content:

Flask

Numpy

Pandas

scikit-learn

```

COPY .
#Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

#Make port 80 available to the world outside this container
EXPOSE 80

#Run app.py when the container launches
CMD ["python","app.py"]
dramkrishnal9@docker:~/ml-app$ docker build -t my-python-app
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build
dramkrishnal9@docker:~/ml-app$ docker run -p 80:80 my-python-app
docker: permission denied while trying to connect to the Docker daemon socket
at unix:///var/run/docker.sock: Head "http://127.0.0.1:4243/v1.40/containers/json?path=/var/run/docker.sock/_ping"
: dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
dramkrishnal9@docker:~/ml-app$ touch requirements.txt
dramkrishnal9@docker:~/ml-app$ nano requirements.txt
dramkrishnal9@docker:~/ml-app$ cat requirements.txt
dramkrishnal9@docker:~/ml-app$ nano requirements.txt
dramkrishnal9@docker:~/ml-app$ cat requirements.txt
dramkrishnal9@docker:~/ml-app$ nano requirements.txt
dramkrishnal9@docker:~/ml-app$ cat requirements.txt
Flask
Numpy
Pandas
scikit-learn
dramkrishnal9@docker:~/ml-app$ 
```

## Step 4: Develop the Machine Learning Application

### 10. Create a Simple ML Model

- Create a script **train\_model.py** to train a simple machine learning model and save it. For simplicity, we'll use the Iris dataset and a decision tree classifier.

```

from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
import pickle

```

```

# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target

```

```

# Train a decision tree classifier
clf = DecisionTreeClassifier() clf.fit(X, y)

```

```

# Save the model to a file
with open('model.pkl', 'wb') as f:
    pickle.dump(clf, f)

```

```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/insta...
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/inst...
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
dramkrishna19@docker:~/ml-app$ pip3 --version
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
dramkrishna19@docker:~/ml-app$ pip3 install scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.3.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.1 MB) |████████| 11.1 MB 1.3 MB/s
Collecting numpy<2.0,>=1.17.3
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB) |████████| 17.3 MB 27.5 MB/s
Collecting joblib>=1.1.1
  Downloading joblib-1.4.2-py3-none-any.whl (301 kB) |████████| 301 kB 45.2 MB/s
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Collecting scipy>=1.5.0
  Downloading scipy-1.10.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.5 MB) |████████| 34.5 MB 34.1 MB/s
Installing collected packages: numpy, joblib, threadpoolctl, scipy, scikit-learn
WARNING: The scripts f2py, f2py3 and f2py3.8 are installed in '/home/dramkrishna19/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed joblib-1.4.2 numpy-1.24.4 scikit-learn-1.3.2 scipy-1.10.1 threadpoolctl-3.5.0
dramkrishna19@docker:~/ml-app$ 

```

11. Run the Model Training Script

- Run the `train_model.py` script to generate `model.pkl`:

```
python train_model.py
```

```

ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/insta...
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/inst...
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

Collecting scikit-learn
  Downloading scikit_learn-1.3.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.1 MB) |████████| 11.1 MB 1.3 MB/s
Collecting numpy<2.0,>=1.17.3
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB) |████████| 17.3 MB 27.5 MB/s
Collecting joblib>=1.1.1
  Downloading joblib-1.4.2-py3-none-any.whl (301 kB) |████████| 301 kB 45.2 MB/s
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Collecting scipy>=1.5.0
  Downloading scipy-1.10.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.5 MB) |████████| 34.5 MB 34.1 MB/s
Installing collected packages: numpy, joblib, threadpoolctl, scipy, scikit-learn
WARNING: The scripts f2py, f2py3 and f2py3.8 are installed in '/home/dramkrishna19/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed joblib-1.4.2 numpy-1.24.4 scikit-learn-1.3.2 scipy-1.10.1 threadpoolctl-3.5.0
dramkrishna19@docker:~/ml-app$ python train_model.py

Command 'python' not found, did you mean:

  command 'python3' from deb python3
  command 'python' from deb python-is-python3

dramkrishna19@docker:~/ml-app$ python3 train_model.py
dramkrishna19@docker:~/ml-app$ ls
Dockerfile model.pkl requirements.txt train_model.py
dramkrishna19@docker:~/ml-app$ 

```

12. Integrate the Model into the Flask App

- Update `app.py` to load the trained model and use it for predictions:

11. Run the Model Training Script
- Run the `train_model.py` script to generate `model.pkl`:

12. Integrate the Model into the Flask App
- Update `app.py` to load the trained model and use it for predictions:

```

from flask import Flask, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Load the trained model
with open('model.pkl', 'rb') as f:
    model = pickle.load(f)

@app.route('/')

```

Page 12 of 14 834 words English (U.S.)

100% Give Feedback to Microsoft  
10:43 AM 2024-10-03

```
from flask import Flask, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Load the trained model
with open('model.pkl', 'rb') as f:
    model = pickle.load(f)

@app.route('/')
def hello_world():
    return 'Hello, Docker!'

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json(force=True)
    prediction = model.predict(np.array(data['input']).reshape(1, -1))
    return jsonify({'prediction': int(prediction[0])})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80)
```

The screenshot displays a Windows desktop environment with three main windows:

- Top Left Window:** An SSH-in-browser session showing the contents of a file named `app.py`. The code implements a Flask application for prediction and serves it on port 80.
- Top Right Window:** A Microsoft Word document titled "Deployment\_of\_AI\_Docker\_Group\_C\_Ram Krishna Dhakal". It contains code snippets for a Flask application and a Dockerfile, along with a section titled "13. Update the Project Directory" which lists required files: `Dockerfile` and `requirements.txt`.
- Bottom Window:** Another Microsoft Word document with the same title and content as the top one, including the "13. Update the Project Directory" section.

Instead of using port 80 (which requires elevated permissions), I have used the port to something higher, like 5000, which doesn't require sudo:

### 13. Update the Project Directory

- Ensure your project directory contains the following files:
  - **Dockerfile**
  - **requirements.txt**

- **train\_model.py**
- **app.py**
- **model.pkl** (generated after running **train\_model.py**)

The screenshot shows a terminal window on the left and a browser window on the right.

**Terminal Output:**

```
dramkrishna19@docker:~/ml-app$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
Permission denied

dramkrishna19@docker:~/ml-app$ sudo python3 app.py
Traceback (most recent call last):
  File "app.py", line 1, in <module>
    from flask import Flask, request, jsonify
ModuleNotFoundError: No module named 'flask'
dramkrishna19@docker:~/ml-app$ nano app.py
dramkrishna19@docker:~/ml-app$ python3 app.py
  File "app.py", line 1
    From flask import Flask, request, jsonify
          ^
SyntaxError: invalid syntax
dramkrishna19@docker:~/ml-app$ nano app.py
dramkrishna19@docker:~/ml-app$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.128.0.13:5000
Press CTRL+C to quit
dramkrishna19@docker:~/ml-app$ python3 train_model.py
dramkrishna19@docker:~/ml-app$ ls
Dockerfile app.py model.pkl requirements.txt train_model.py
dramkrishna19@docker:~/ml-app$ touch app.py
dramkrishna19@docker:~/ml-app$ nano app.py
dramkrishna19@docker:~/ml-app$ python3 train_model.py
dramkrishna19@docker:~/ml-app$ ls
Dockerfile app.py model.pkl requirements.txt train_model.py
dramkrishna19@docker:~/ml-app$ 
```

**Browser Log:**

AzureLoyalistCollege-my.share... Deployment of AI\_Docker\_Group\_C\_Ram Krishna Dhakal ( ...

File Home Insert Layout References Review View Help

Using port 80 (which requires elevated permissions), I have used the port 5000, which doesn't require sudo:  
date the Project Directory

○ Ensure your project directory contains the following files:

- Dockerfile
- requirements.txt
- train\_model.py
- app.py
- model.pkl (generated after running train\_model.py)

Build and Run the Docker Container  
Build the Docker Image  
○ Build the Docker image with the following command:  
sudo docker build -t ml-app

Page 14 of 16 22 of 857 words English (U.S.)

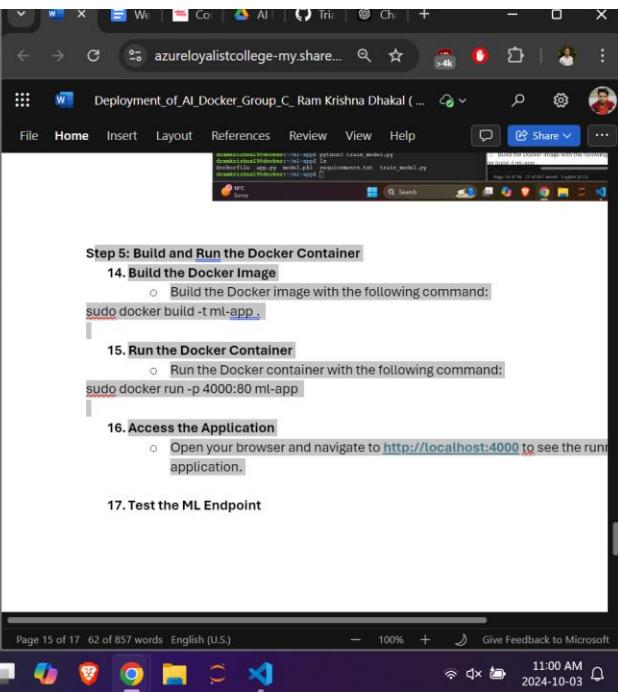
## Step 5: Build and Run the Docker Container

### 14. Build the Docker Image

- Build the Docker image with the following command:

sudo docker build -t ml-app .

```
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/inst...  
SSH-in-browser  
UPLOAD FILE DOWNLOAD FILE ! ☰  
  
docker: Error response from daemon: pull access denied for ml-app, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.  
See 'docker run --help'.  
draknarkish@draknarkish:~/ml-app$ sudo docker build -t ml-app .  
draknarkish@draknarkish:~/ml-app$ docker build -t ml-app .  
  [1/1] Building 15.5s (7/8) docker:default  
    => [internal] load build definition from Dockerfile 0.0s  
    => transferring Dockerfile: 484B 0.0s  
    => [internal] load metadata for docker.io/library/python:3.9-slim 0.0s  
    => [internal] load .dockerignore 0.0s  
    => transferring context: 2B 0.0s  
    => [1/4] FROM docker.io/library/python:3.9-slim@sha256:4f94609e5a997d 4.3s  
    =>  resolve docker.io/library/python:3.9-slim@sha256:4f94609e5a997d 0.0s  
    =>  sha256:9deec08abbahbd3878f2008f9445f19a52284 14.74MB / 14.74MB 0.4s  
    =>  sha256:f9794609e5a997d01c6086a6ac564591854031 10.41KB / 10.41KB 0.0s  
    =>  sha256:93ab510a8e31001ea79c4ef30ceee628268646 1.75KB / 1.75KB 0.0s  
    =>  sha256:b6307c037d03e63983654703064ca04872511e4 5.20KB / 5.20KB 0.0s  
    =>  sha256:302e3e498053a7b5332a79e8febe1ce0028e 29.13MB / 29.13MB 0.6s  
    =>  sha256:a0965d31919510506d8856650a96e967dae 3.51MB / 3.51MB 0.2s  
    =>  sha256:22a08b8d0d4f53ad5493dabf2af00ccde91abb3771fb21 248B / 248B 0.3s  
    =>  extracting sha256:302e3e498053a7b5332a79e8febe1ce0028e97clce 2.0s  
    =>  extracting sha256:40965d31919510506d8856650a96e967daae964e 0.2s  
    =>  extracting sha256:fdd6ec08abbahbd3878f2008f9445f19a5224d1b7e77 1.1s  
    =>  extracting sha256:d2a08b8d0d4f53ad5493dabf2af00ccde91abb3771fb218 0.0s  
    => [internal] load build context: 0.0s  
    => transferring context: 4.20kB 0.0s  
    [2/4] WORKDIR /usr/src/app 0.0s  
    [3/4] COPY . 0.0s  
[4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s  
    => # Collecting six==1.5  
    => # Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)  
    => # Installing collected packages: pytz, zipp, tzdata, threadpoolctl, si  
    => # x, NumPy, MarkupSafe, joblib, itsdangerous, click, blinker, Werkzeug  
    => # , scipy, python-dateutil, Jinja2, importlib-metadata, scikit-learn,  
    => # Pandas, Flask
```



## 15. Run the Docker Container

- Run the Docker container with the following command:

```
sudo docker run -p 4000:80 ml-app
```

```
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/inst... - □
% ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/inst...
SSH-in-browser UPLOAD FILE DOWNLOAD FILE
 ! 📈 🛡️ ⚙️

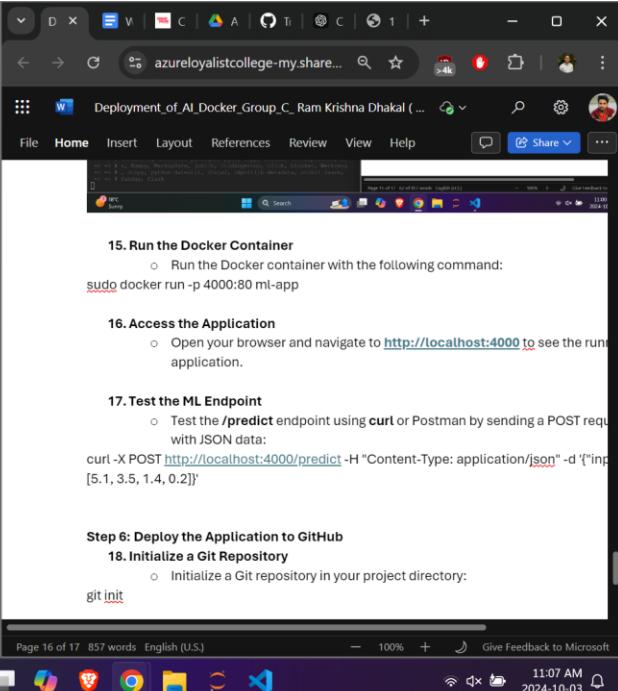
drakmkrishna19@docker:~/ml-app$ scikit-learn==1.3.2  

scikit-learn==1.3.2: command not found  

drakmkrishna19@docker:~/ml-app$ sudo docker build -t ml-app .  

[+] Building 0.4s (9/9) FINISHED docker:default
--> [internal] load build definition from Dockerfile 0.0s
--> => transferring dockerfile: 484B 0.0s
--> [internal] load metadata for docker.io/library/python:3.9-slim 0.0s
--> [internal] load .dockerignore 0.0s
--> => transferring content: 2B 0.0s
--> [1/4] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997d 0.0s
--> [internal] load build context 0.0s
--> => transferring context: 156B 0.0s
--> CACHED [2/4] WORKDIR /usr/src/app 0.0s
--> CACHED [3/4] COPY . 0.0s
--> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s
--> exporting to image 0.0s
--> => exporting layers 0.0s
--> => writing image sha256:779505063b0894377380bd7c08ad40194fff8c2c8f6 0.0s
--> => naming to docker.io/library/ml-app 0.0s
drakmkrishna19@docker:~/ml-app$ sudo docker run -p 4000:80 ml-app  

/usr/lib/python3.9/site-packages/sklearn/base.py:376: InconsistentVersionWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.3
.2 when using version 1.5.2. This might lead to breaking code or invalid results.
Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limits
warnings.warn(
    * Serving Flask app 'app'
    * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
t. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0:0)
* Running on http://127.0.0.1:80
* Running on http://172.17.0.2:80
Press CTRL+C to quit
□
```



## **16. Access the Application**

- Open your browser and navigate to <http://localhost:4000> to see the running application.

## 17. Test the ML Endpoint

- Test the **/predict** endpoint using **curl** or Postman by sending a POST request with JSON data:

```
curl -X POST http://localhost:4000/predict -H "Content-Type: application/json" -d '{"input": [5.1, 3.5, 1.4, 0.2]}'
```

The terminal window on the left shows the following Docker build logs:

```
[+] Building 0.4s (9/9) FINISHED
--> [internal] load build definition from Dockerfile          docker:default
--> [internal] transfering dockerfile: 484B                  0.0s
--> [internal] load metadata for docker.io/library/python:3.9-slim   0.0s
--> [internal] load .dockerignore                           0.0s
--> [internal] transfering context: 2B                   0.0s
--> [1/4] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997d 0.0s
--> [internal] load build context                         0.0s
--> [internal] transfering context: 156B                 0.0s
--> [internal] CACHED [2/4] WORKDIR /usr/src/app        0.0s
--> [internal] CACHED [3/4] COPY . .                      0.0s
--> [internal] CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s
--> [internal] exporting to image                        0.0s
--> [internal] exporting layers                         0.0s
--> [internal] writing image sha256:779505063b0894377380bd7c08ad40194ffff8c2c8f6 0.0s
--> [internal] naming to docker.io/library/ml-app       0.0s
drakrishnal19@docker:~/ml-app$ sudo docker run -p 4000:80 ml-app
/usr/local/lib/python3.9/site-packages/sklearn/base.py:376: InconsistentVersionWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.3 .2 when using version 1.5.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
  * Serving Flask app 'app'
  * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:80
  * Running on http://172.17.0.2:80
Press CTRL+C to quit
^Cdrakrishnal19@docker:~/ml-app$ curl -X POST http://localhost:4000/predict -H "Content-Type: application/json" -d '{"input": [5.1, 3.5, 1.4, 0.2]}'
curl: (7) Failed to connect to localhost port 4000: Connection refused
drakrishnal19@docker:~/ml-app$
```

The Microsoft Edge browser window on the right displays a step-by-step guide:

- 16. Access the Application**  
Open your browser and navigate to <http://localhost:4000> to see the running application.
- 17. Test the ML Endpoint**  
Test the **/predict** endpoint using **curl** or Postman by sending a POST request with JSON data:  
`curl -X POST http://localhost:4000/predict -H "Content-Type: application/json" -d '{"input": [5.1, 3.5, 1.4, 0.2]}'`
- Step 6: Deploy the Application to GitHub**
- 18. Initialize a Git Repository**  
Initialize a Git repository in your project directory:  
`git init`
- 19. Add All Files and Commit**  
Add all files to the repository and commit:  
`git add .`  
`git commit -m "Initial commit"`

## Step 6: Deploy the Application to GitHub

### 18. Initialize a Git Repository

- Initialize a Git repository in your project directory:

```
git init
```

The screenshot shows a dual-monitor setup. The left monitor displays a terminal window with a black background and white text, showing the output of a Python script running on a Docker container. The right monitor displays a Microsoft Edge browser window with a white background and black text, showing a GitHub repository page.

**Terminal Output (Left Monitor):**

```
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c/insta... - D X V C A Ti C rl + ...  
ssh.cloud.google.com/v2/ssh/projects/regal-muse-436520-i3/zones/us-central1-c... N D X V C A Ti C rl + ...  
SSH-in-browser UPLOAD FILE DOWNLOAD FILE J K L  
/usr/local/lib/python3.9/site-packages/sklearn/base.py:376: InconsistentVersionWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.3 .2 when using version 1.5.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:  
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limits  
  warnings.warn(  
    * Serving Flask app 'app'  
    * Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
  * Running on all addresses (0.0.0.0)  
  * Running on http://127.0.0.1:80  
  * Running on http://172.17.0.2:80  
Press CTRL+C to quit  
^dramkrishna19@docker:~/ml-app$ curl -X POST http://localhost:4000/predict -H  
"Content-Type: application/json" -d '{"input": [5.1, 3.5, 1.4, 0.2]}'  
curl: (7) Failed to connect to localhost port 4000: Connection refused  
dramkrishna19@docker:~/ml-app$ git init  
Initialized empty Git repository in /home/dramkrishna19/ml-app/.git/  
dramkrishna19@docker:~/ml-app$ git add .  
dramkrishna19@docker:~/ml-app$ git commit -m "Initial commit"  
  
*** Please tell me who you are.  
  
Run  
  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository.  
  
fatal: empty ident name (for <dramkrishna19@docker.us-central1-c.c.regal-muse-436520-i3.terraform>) not allowed  
dramkrishna19@docker:~/ml-app$
```

**Browser Output (Right Monitor):**

Deployment\_of\_AI\_Docker\_Group\_C\_Ram\_Krishna\_Dhalak (https://github.com/...)

File Home Insert Layout References Review View Help Share

18. Initialize a Git Repository

- Initialize a Git repository in your project directory:

```
git init
```

19. Add All Files and Commit

- Add all files to the repository and commit:

```
git add .  
git commit -m "Initial commit"
```

20. Create a New Repository on GitHub

- Create a new repository on GitHub and follow the instructions to push your local repository to GitHub:

```
git remote add origin https://github.com/yourusername/your-repository.git
```

Page 17 of 18 5 of 857 words English (U.S.) 100% Give Feedback to Microsoft 11:11 AM 2024-10-03

## 19. Add All Files and Commit

- Add all files to the repository and commit:

```
git add .
```

```
git commit -m "Initial commit"
```

## 20. Create a New Repository on GitHub

- Create a new repository on GitHub and follow the instructions to push your local repository to GitHub:

```
git remote add origin https://github.com/yourusername/your-repository.git
```

```
git branch -M main
```

```
git push -u origin main
```

## Step 7: Document the Process

### 21. Create a README.md File

- Document the process in a **README.md** file in your repository. Include the following:
  - Overview of the project
  - Instructions to build and run the Docker container
  - Instructions to test the ML endpoint
  - Any other relevant information about the project

## Submission

- Take screenshots of every step you perform and paste in the submission word/pdf file.
- Submit the GitHub repository link of your project.
- Ensure your repository is public and the README.md file is well-documented.