7. Deploy a web application using docker

- Name: Om Deore
- PRN: 22110318
- Div: TY(A)
- RollNo: 321022
- github: All Assignments

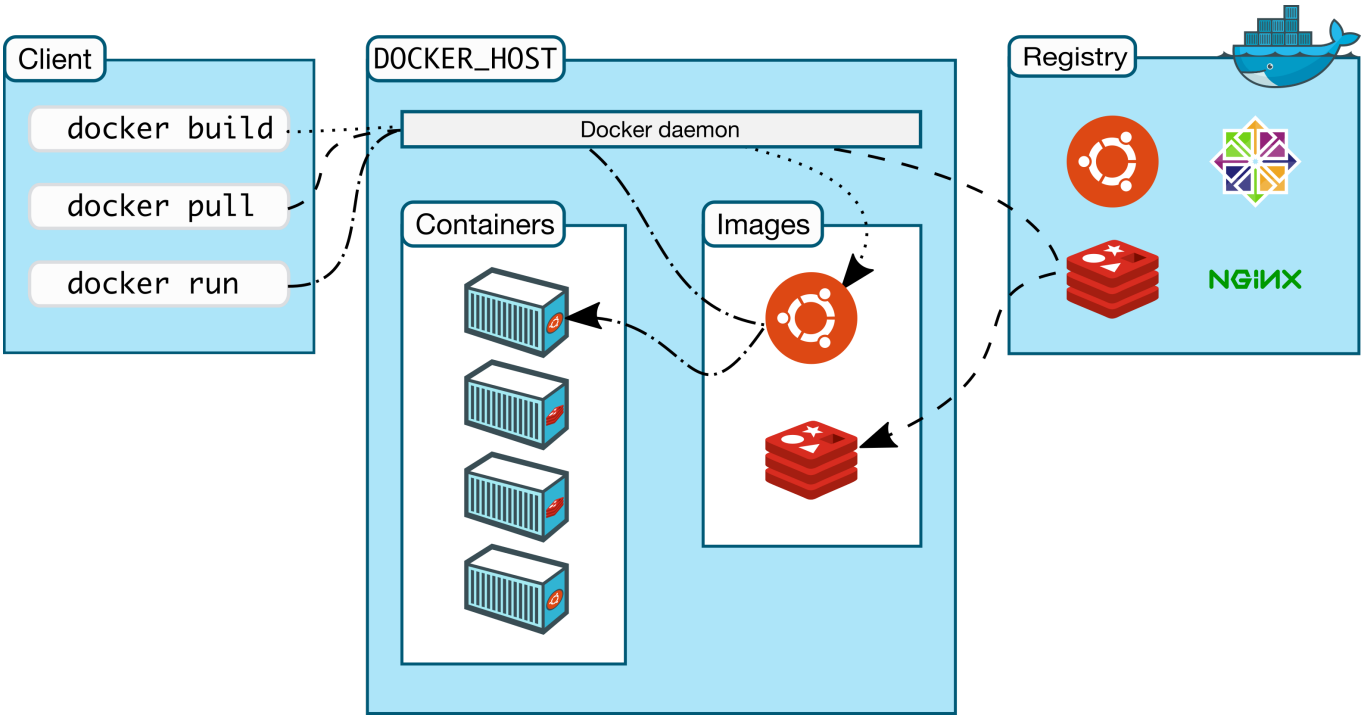
Theory

1) What is Docker?

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, you can significantly reduce the delay between writing code and running it in production.

2) Docker Architecture

Docker architecture consists of three main components:
• Docker Engine : The runtime that runs and manages containers on a host system.
• Docker Images : Lightweight, standalone, executable packages that contain everything needed to run a piece of software, including the code, runtime, libraries, and dependencies.
• Docker Registries : Repositories for Docker images, allowing users to store and share container images.



3) Difference between Docker and VM

1. Virtualisation : From our understanding thus far, both virtual machines and Docker containers provide isolated environments to run applications. The key difference between the two is in *how* they facilitate this isolation. Recall that a VM boots up its own guest OS. Therefore, it virtualises both the operating system kernel and the application layer. A Docker container virtualises *only* the application layer, and runs on top of the host operating system
2. Compatibility : A virtual machine uses its own operating system and is *independent* of the host operating system that it's running on. Therefore, a VM is compatible with all operating systems. A Docker container, on the other hand, is compatible with *any* Linux distribution. You may run into some problems running Docker on a Windows machine or an older Mac.
3. Size : A Docker image is lightweight and is typically in the order of kilobytes. A Docker image denotes the artifact containing the application, its associated dependencies, and configuration. A running instance of the Docker image is called a container. A VM instance can be as large as a few gigabytes or even terabytes
4. Performance : In terms of performance, Docker containers provide near-native performance. Because they are lightweight, you can start them in a few milliseconds. Starting a VM is equivalent to setting up a standalone machine inside your computer. It can take as long as a few minutes to start a VM instance.
5. Security : Docker containers run on top of the host operating system. Therefore, if the host OS is susceptible to security vulnerabilities, so are the Docker containers. Virtual machines, on the other hand, boot up their own operating system, and are more secure.

Recall: each virtual machine is a fully blown machine running inside another. If you have stringent security constraints to be met for sensitive applications, you should consider using a virtual machine instead.

4) Docker Commands

- Build an Image from a Dockerfile

```
docker build -t <image_name>
```

- List local images

```
docker images
```

- Create and run a container from an image, with a custom name:

```
docker run --name <container_name> <image_name>
```

- Run a container with and publish a container's port(s) to the host.

```
docker run -p <host_port>:<container_port> <image_name>
```

- Run a container in the background

```
docker run -d <image_name>
```

- Start or stop an existing container:

```
docker start|stop <container_name> (or <container-id>)
```

- Login into Docker

```
docker login -u <username>
```

- Publish an image to Docker Hub

```
docker push <username>/<image_name>
```

- Search Hub for an image

```
docker search <image_name>
```

- Pull an image from a Docker Hub

```
docker pull <image_name>
```

5) What is Dockerfile

A Dockerfile is a text document that contains instructions for building a Docker image. It specifies the base image to use, any additional dependencies to install, and commands to run when the container starts. Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. This page describes the commands you can use in a Dockerfile.

6) Docker-compose and Docker-swarm

1. Docker-compose - A tool for defining and running multi- container Docker applications. It uses a YAML file to configure the application's services, networks, and volumes, making it easy to manage complex applications.
2. Docker-swarm - A native clustering and orchestration tool for Docker. It allows users to create and manage a cluster of Docker nodes, providing features like service discovery, load balancing, and rolling updates for containerized applications.

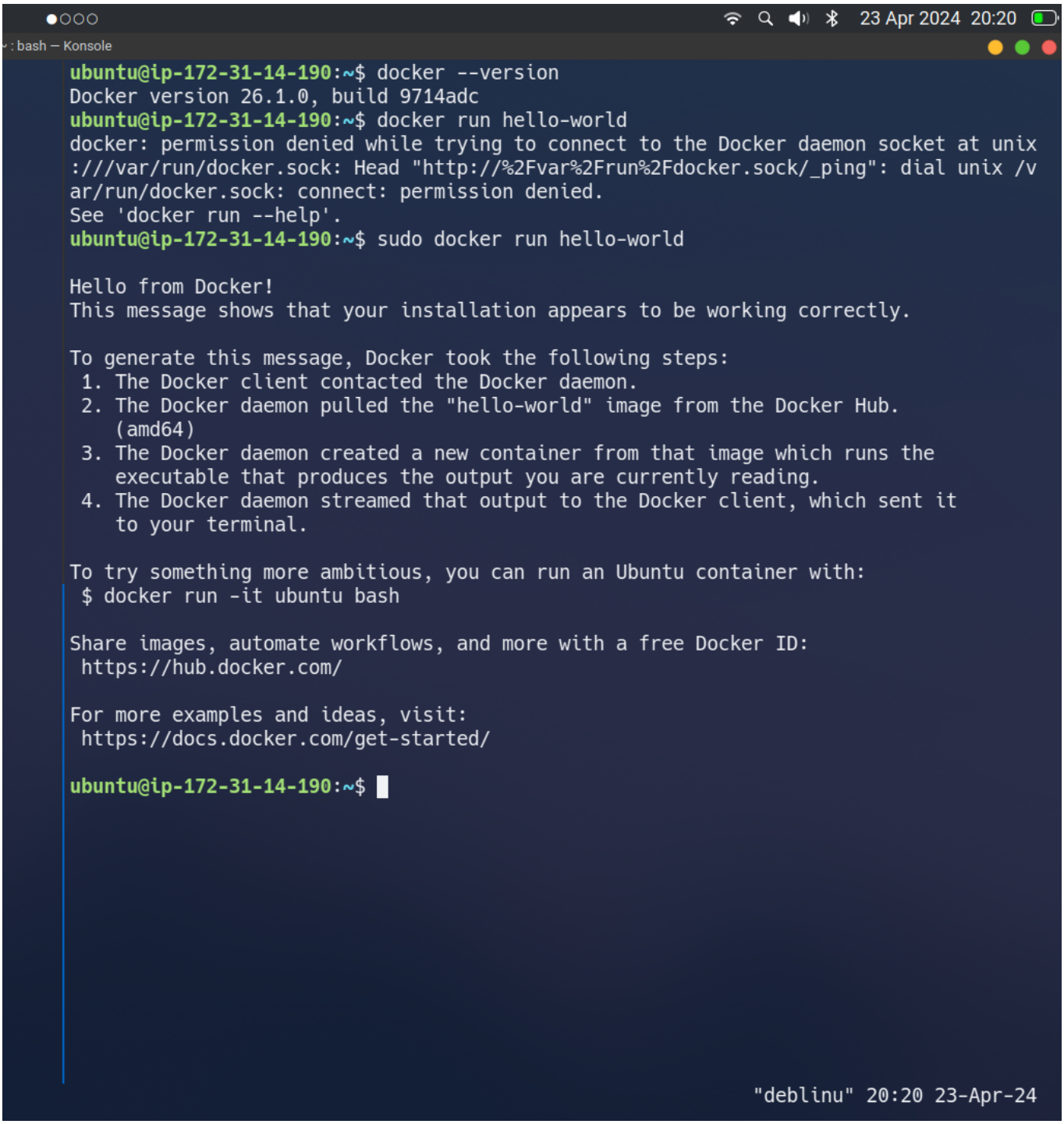
Implementation:

Step 1) Install Docker
Run following script in bash

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
```

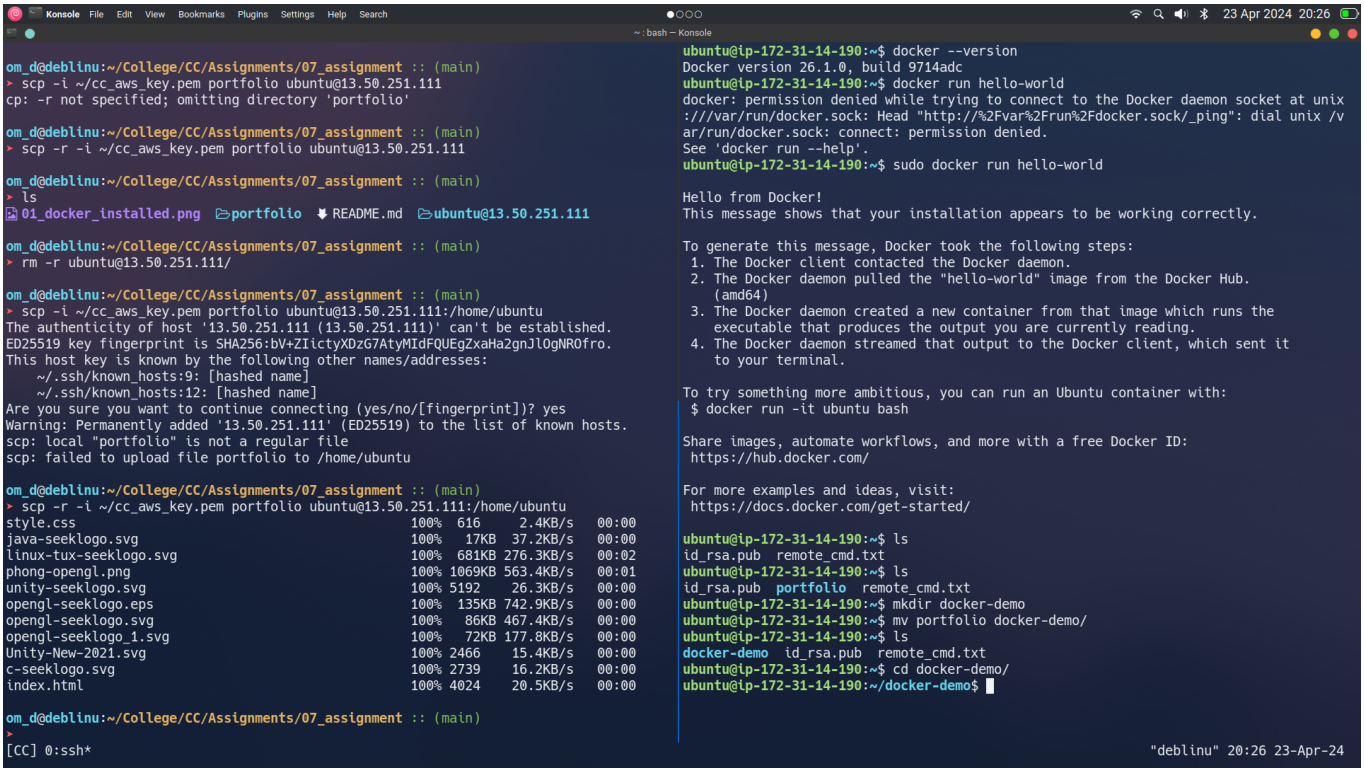
```
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin
```

After running this script verify your docker install and run demo docker image 'hello-world'



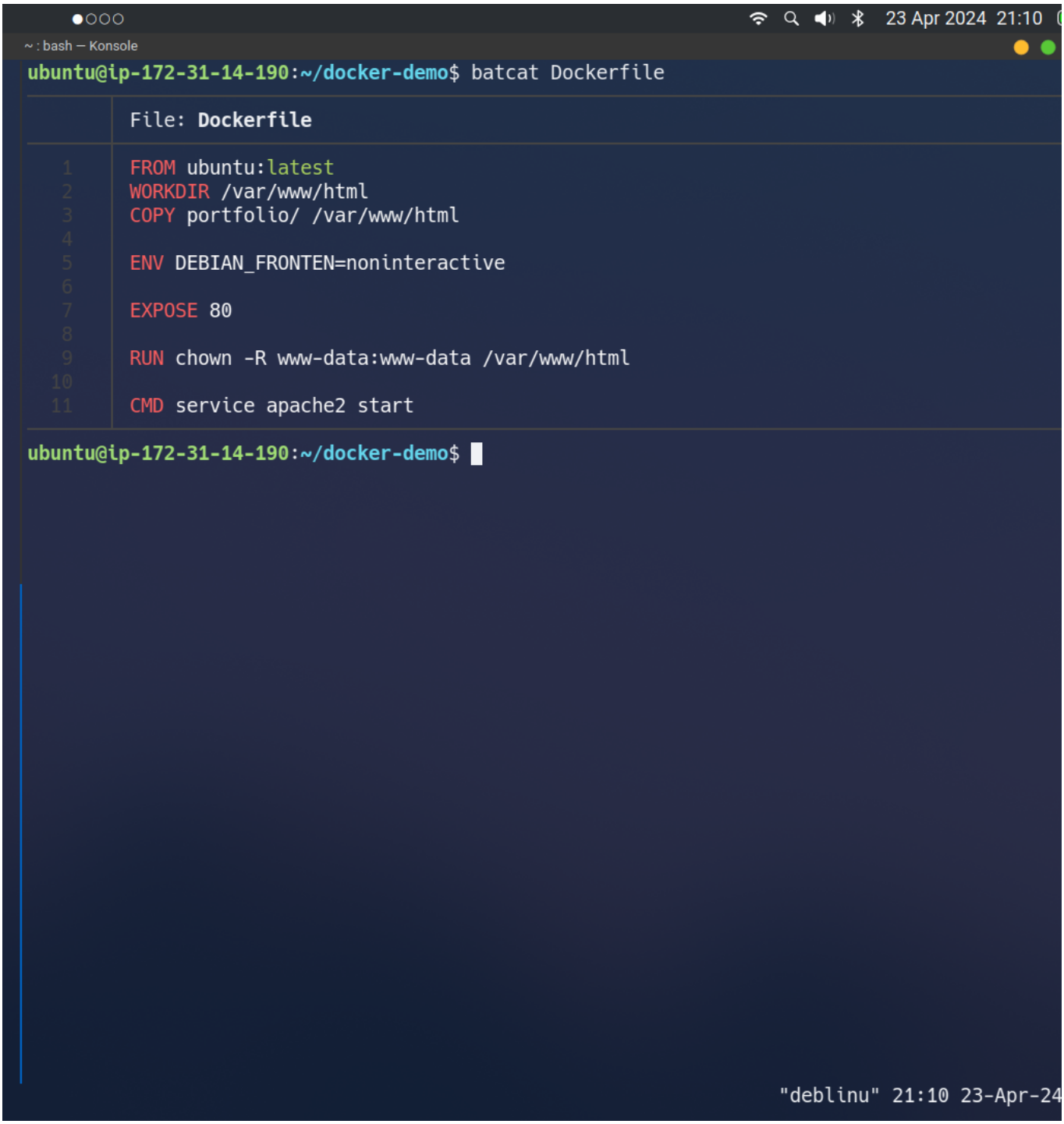
Step 2) Getting files into current folder

scp project folder into our server

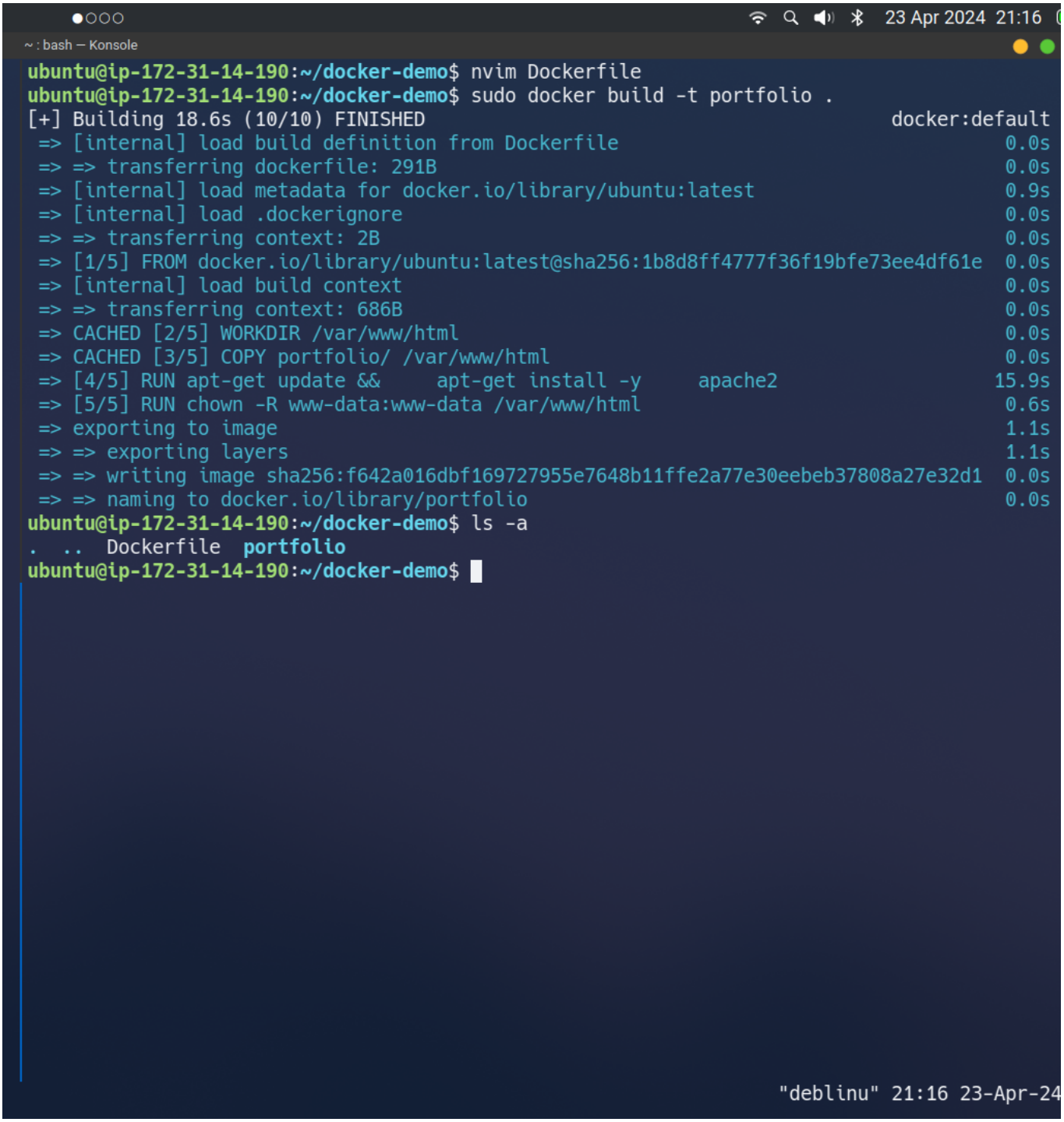


Step 3) Dockerfile

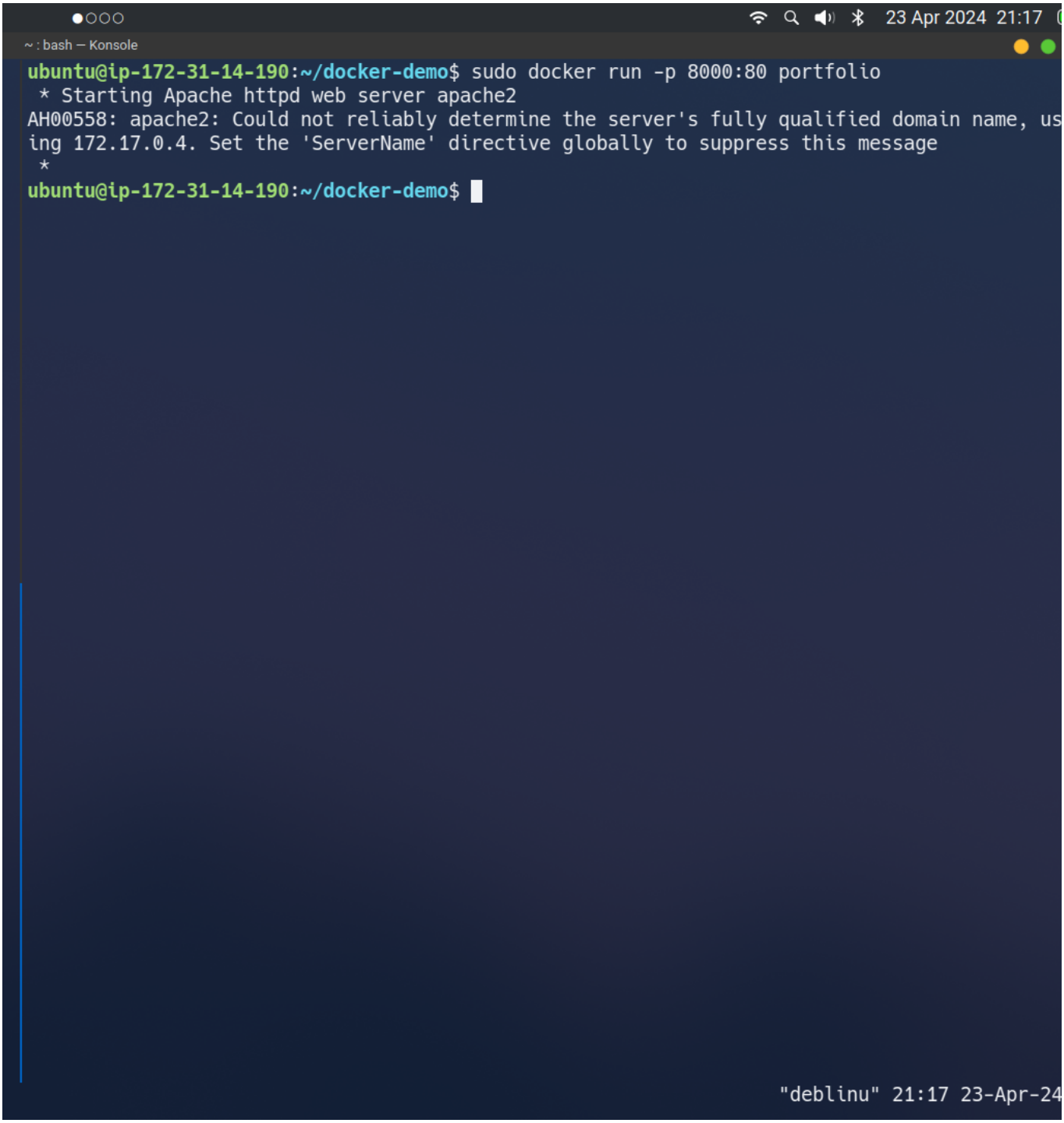
Create a docker file for this application


23 Apr 2024 21:10
~ : bash — Konsole
ubuntu@ip-172-31-14-190:~/docker-demo$ batcat Dockerfile
File: Dockerfile
1 FROM ubuntu:latest
2 WORKDIR /var/www/html
3 COPY portfolio/ /var/www/html
4
5 ENV DEBIAN_FRONTEN=noninteractive
6
7 EXPOSE 80
8
9 RUN chown -R www-data:www-data /var/www/html
10
11 CMD service apache2 start
ubuntu@ip-172-31-14-190:~/docker-demo$
"deblinu" 21:10 23-Apr-24

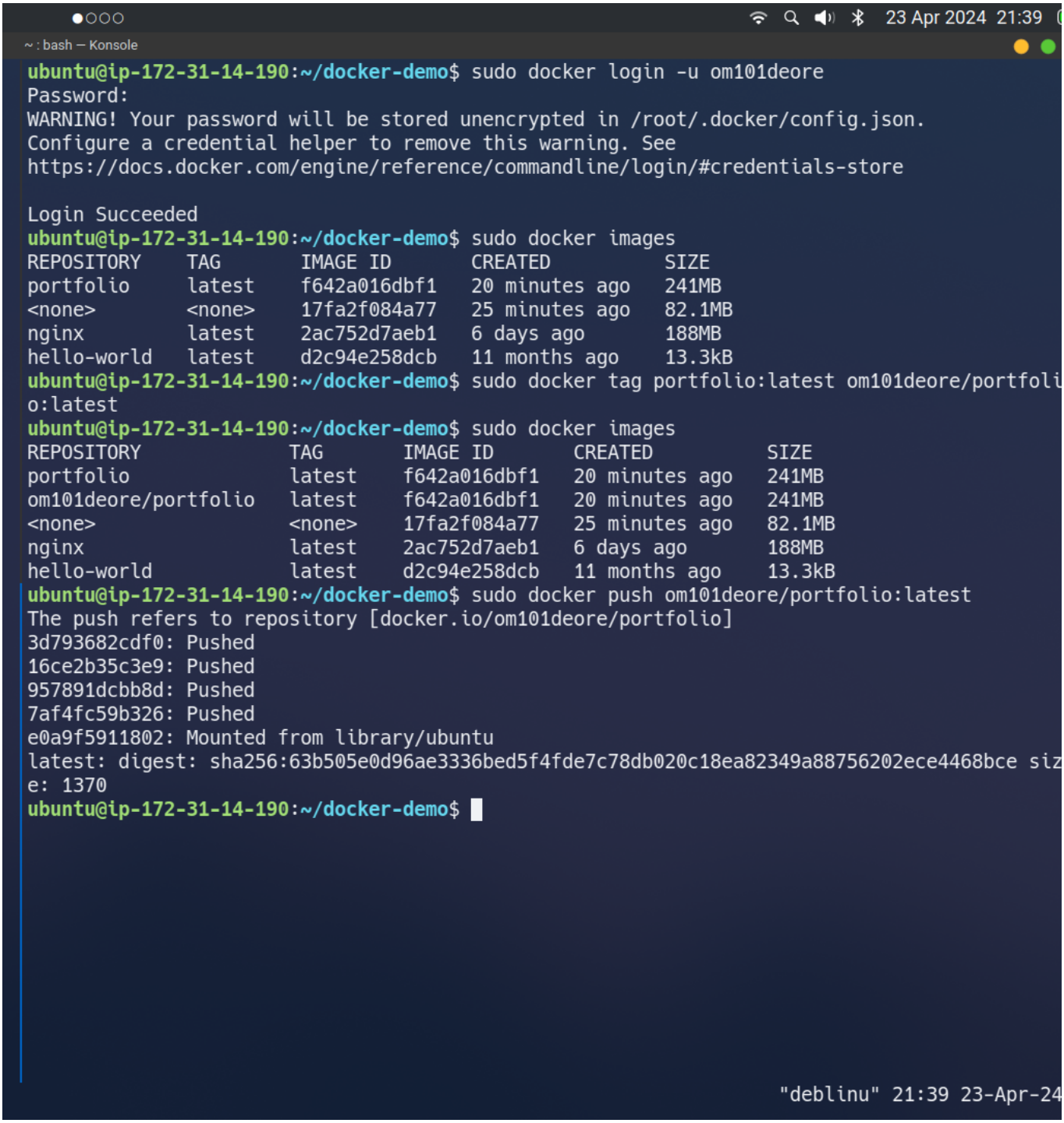
Step 4) Build the docker image



Step 5) run Docker image



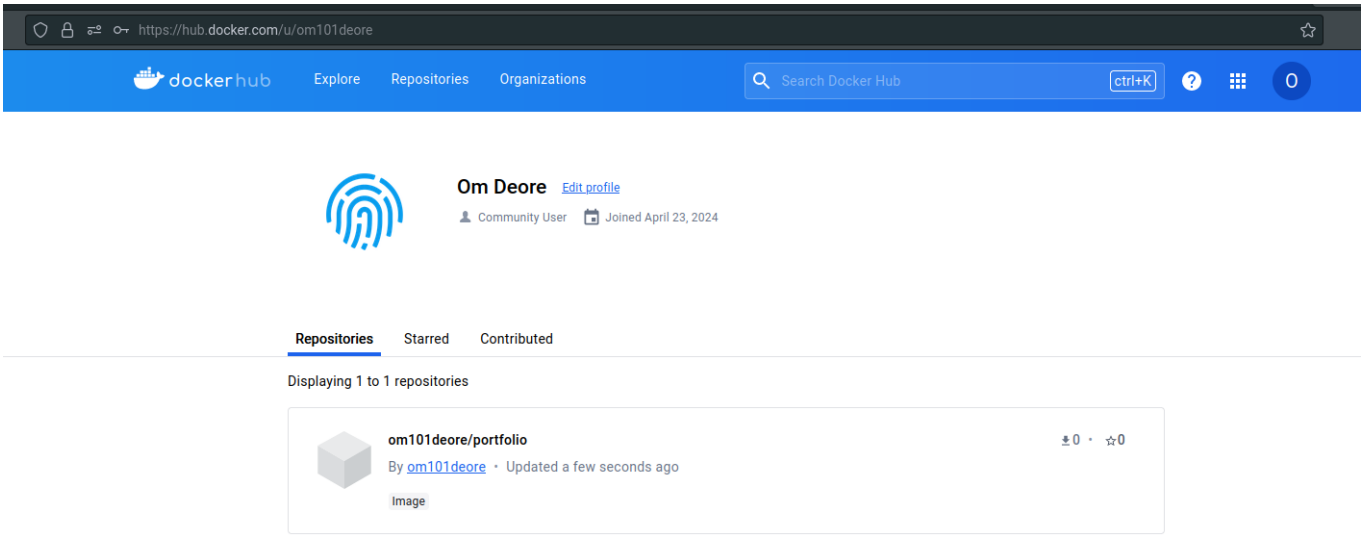
Step 6) Login to docker hub and push image



Step 7) Image on Docker hub

This way we have created our docker image and hosted it on docker hub. Now anyone can use

this image


https://hub.docker.com/u/om101deore
dockerhub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
0
Om Deore
Edit profile
Community User
Joined April 23, 2024
Repositories
Starred
Contributed
Displaying 1 to 1 repositories
om101deore/portfolio
By om101deore • Updated a few seconds ago
Image
0 • ☆0