Name : Aryan Patel
Email : aaryanpatel03@gmail.com
Domain : Devops

WEEK 4 Assignment (Docker)

1. Introduction to containerization and Docker fundamentals, Basic Commands.

2. Docker installation and basic container operations, Build an image from Dockerfile.

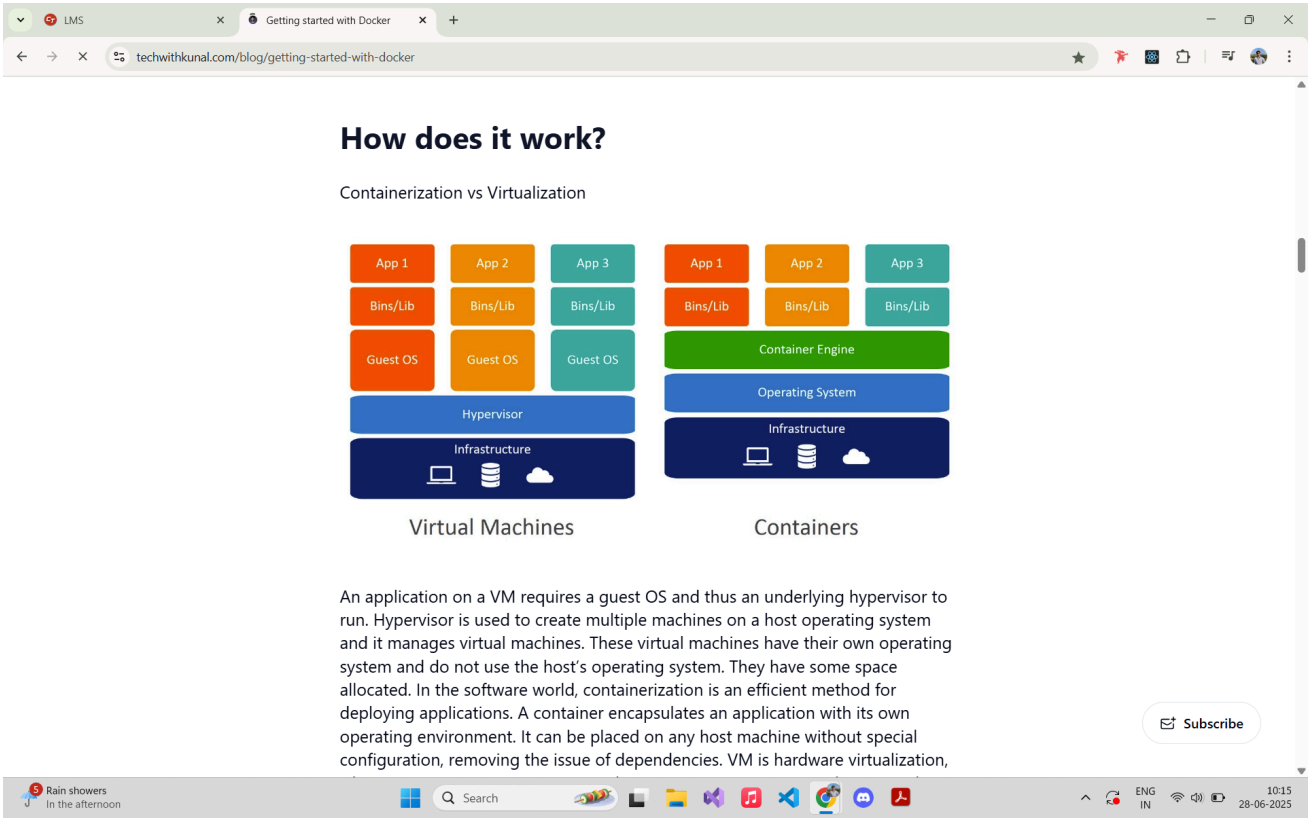
Steps :

1. What is Containerization?

A lightweight alternative to full machine virtualization. Packages the application and its dependencies into a single image that runs consistently in any environment.

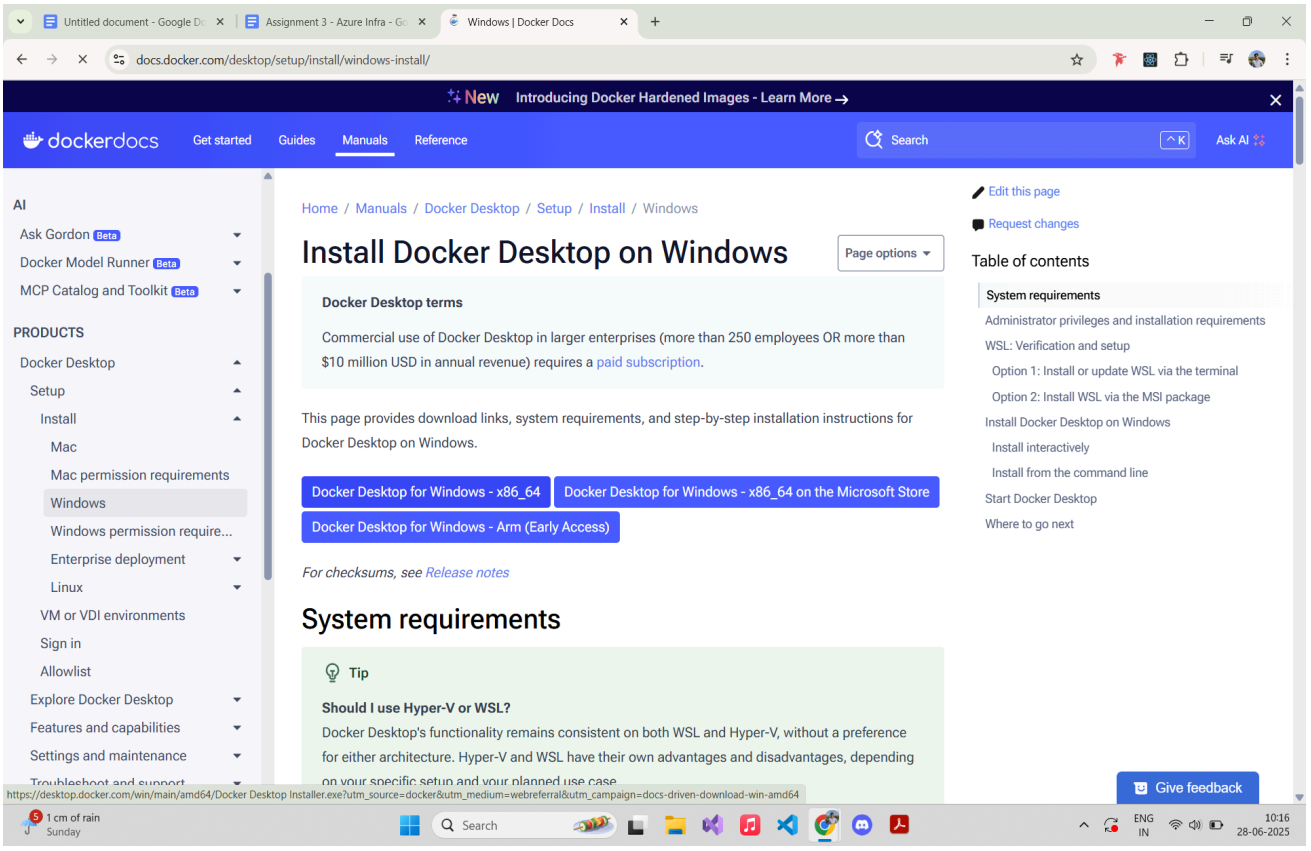
Benefits :

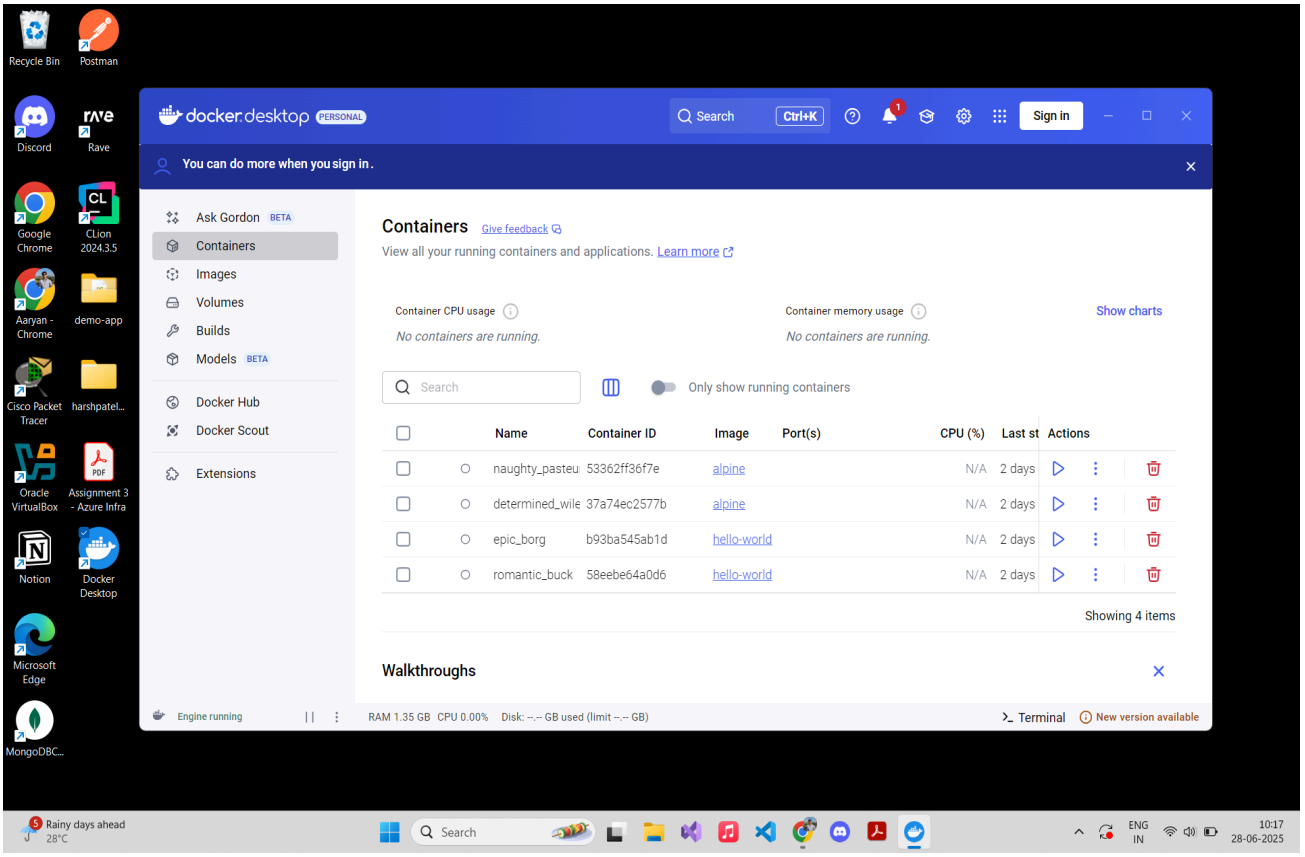
- Portability
- Efficiency
- Scalability
- Isolation



Reference Link : https://www.techwithkunal.com/blog/getting-started-with-docker

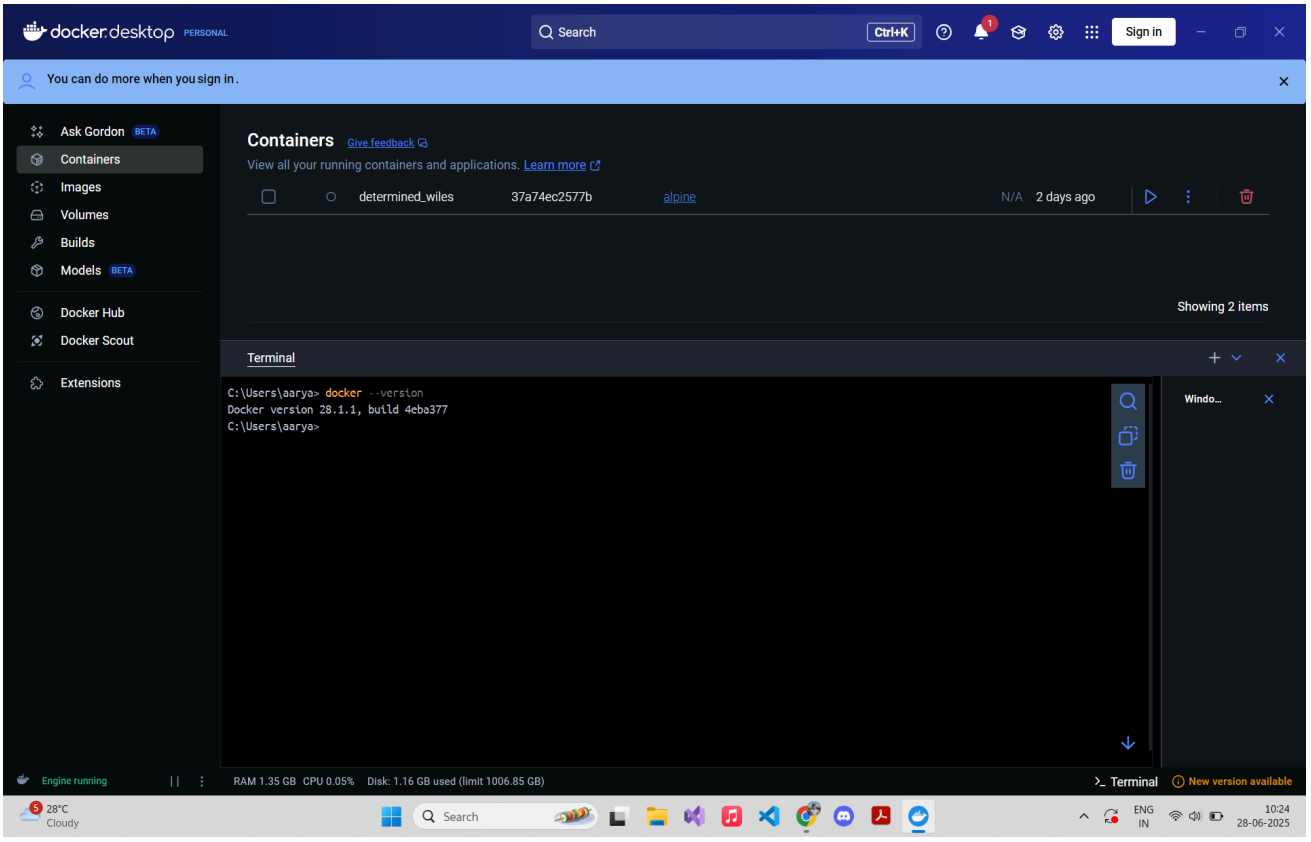
2. Install Docker Desktop.



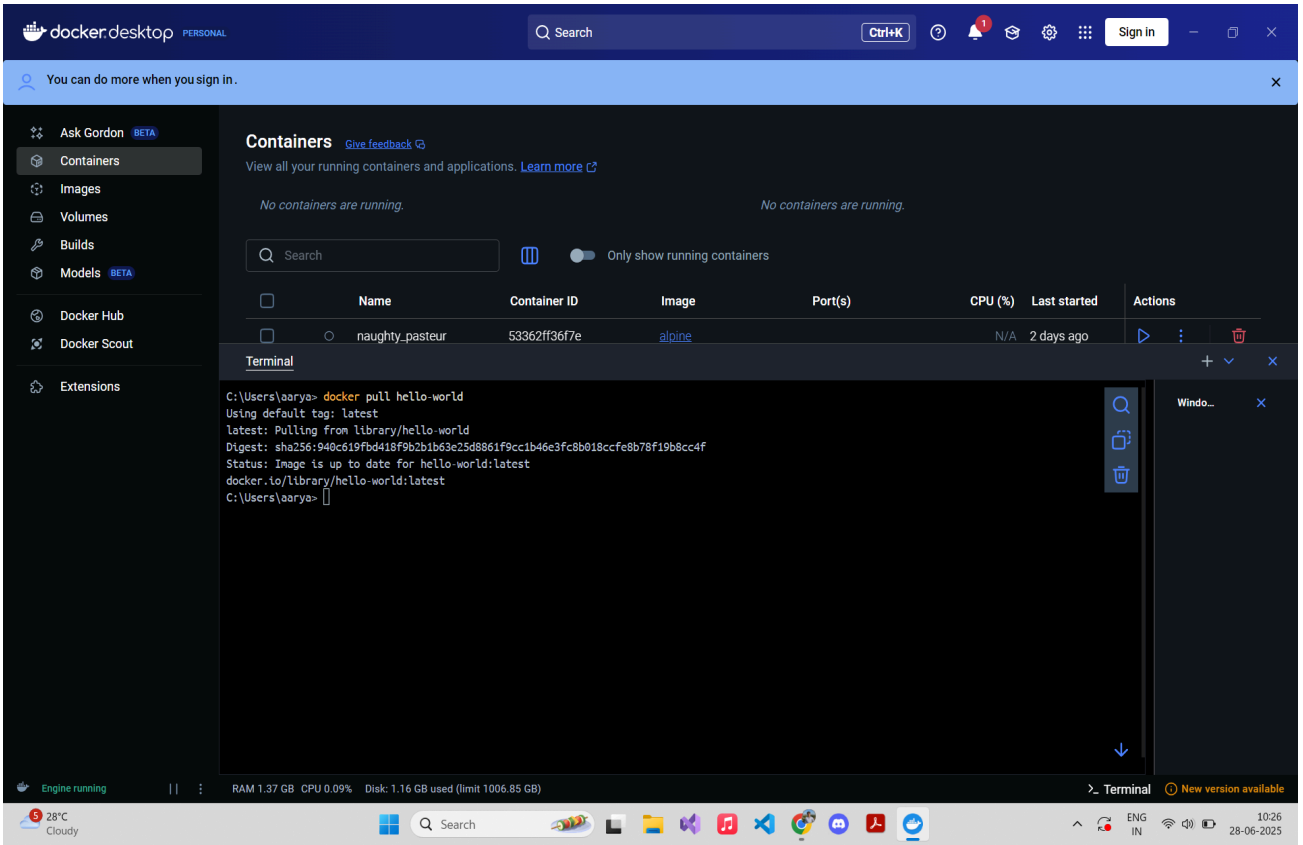


3. Docker fundamentals and basic commands.

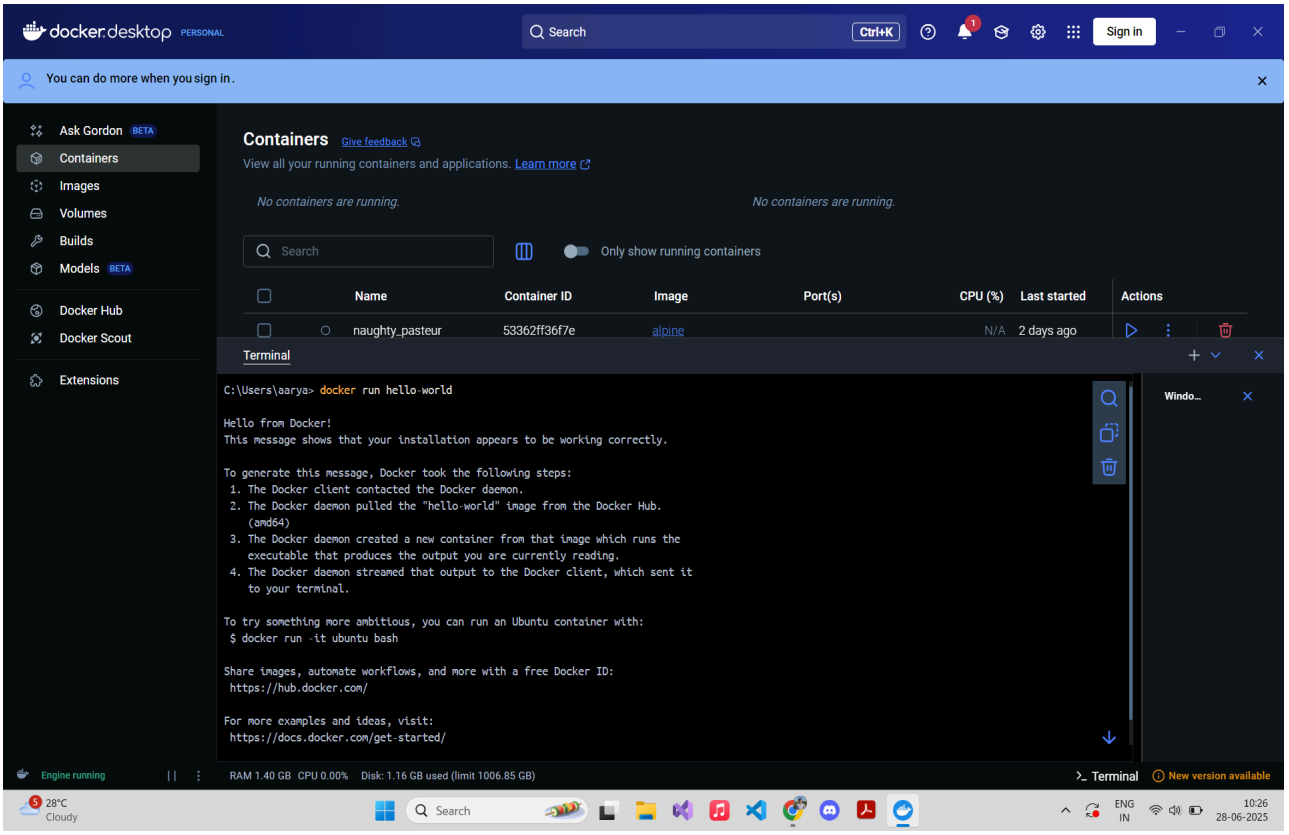
- Check Docker Version



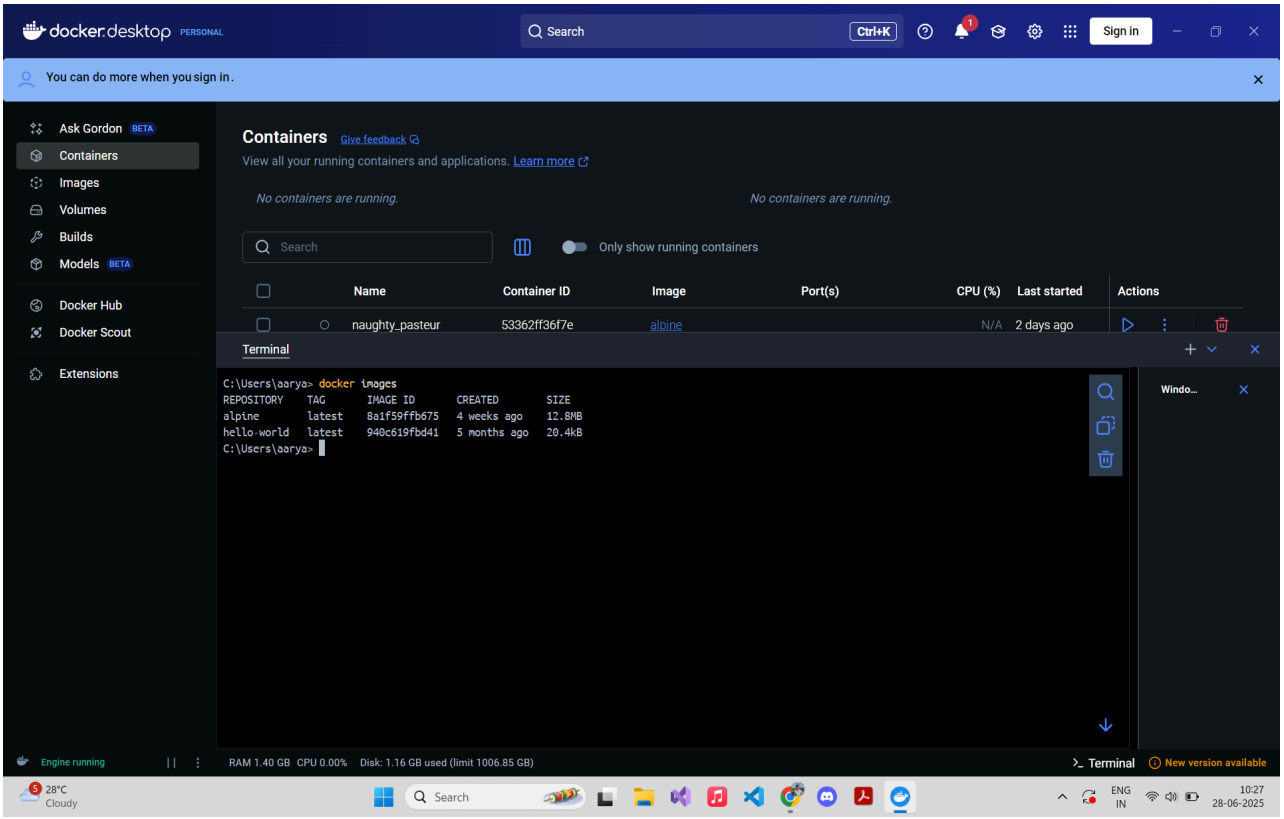
- Pull an Image



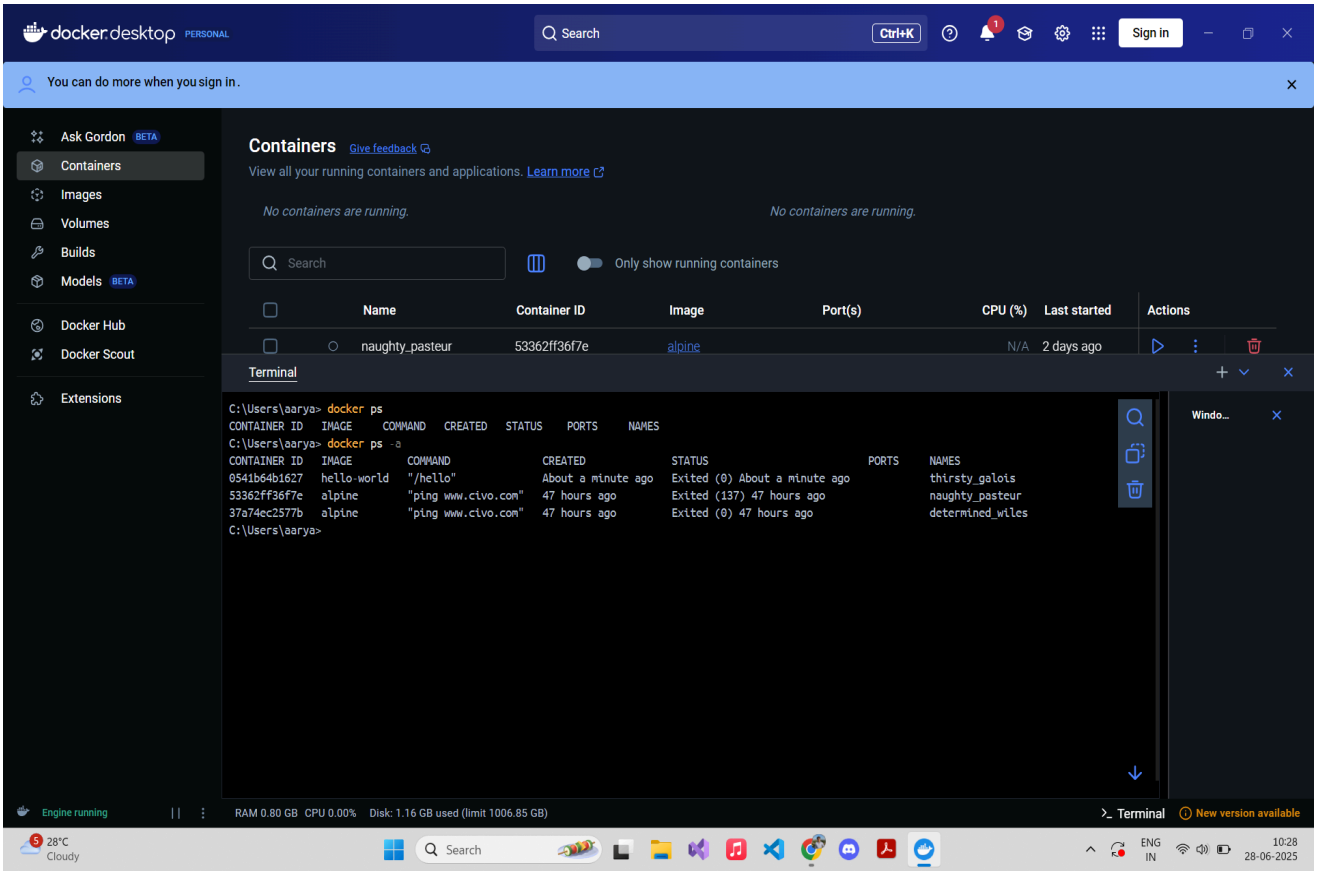
- Run a Container



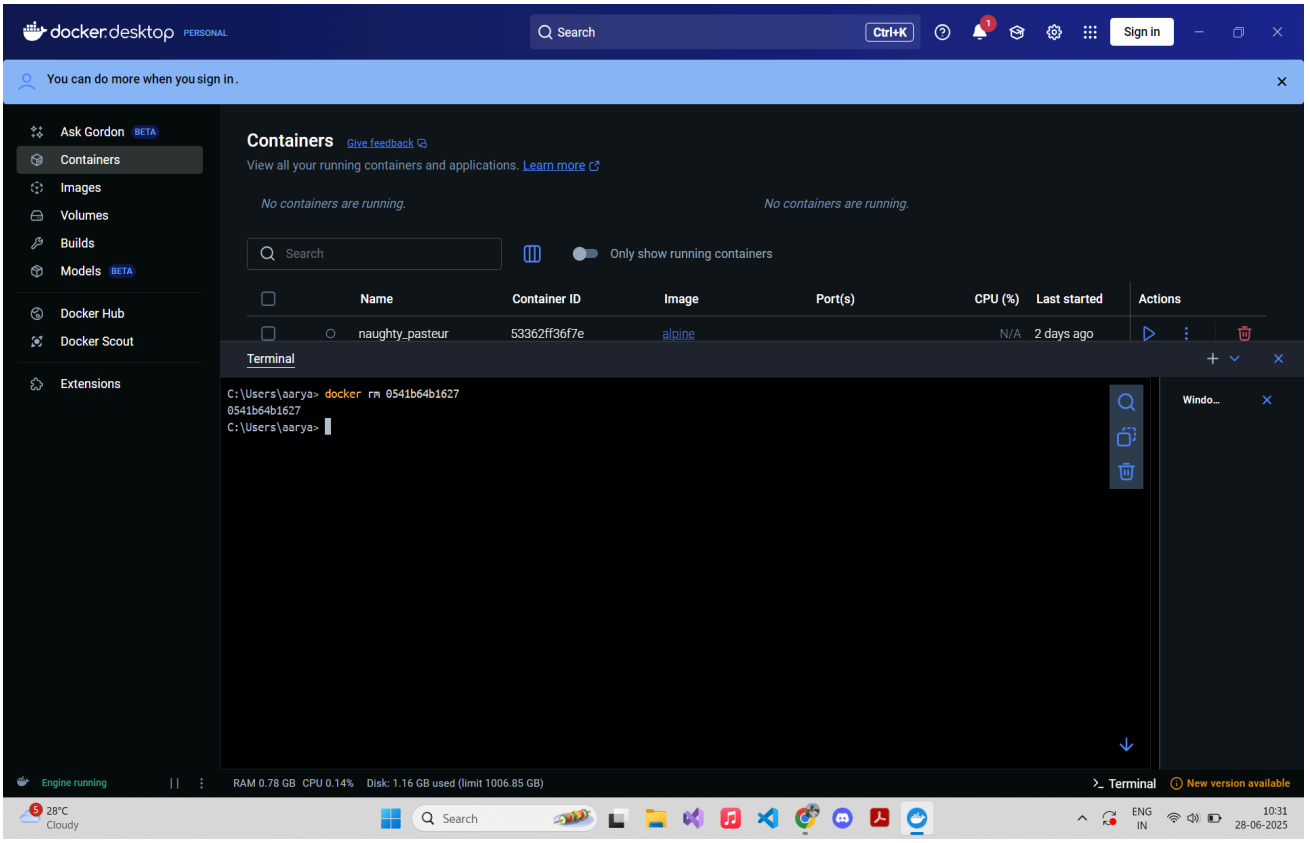
- List Docker Images



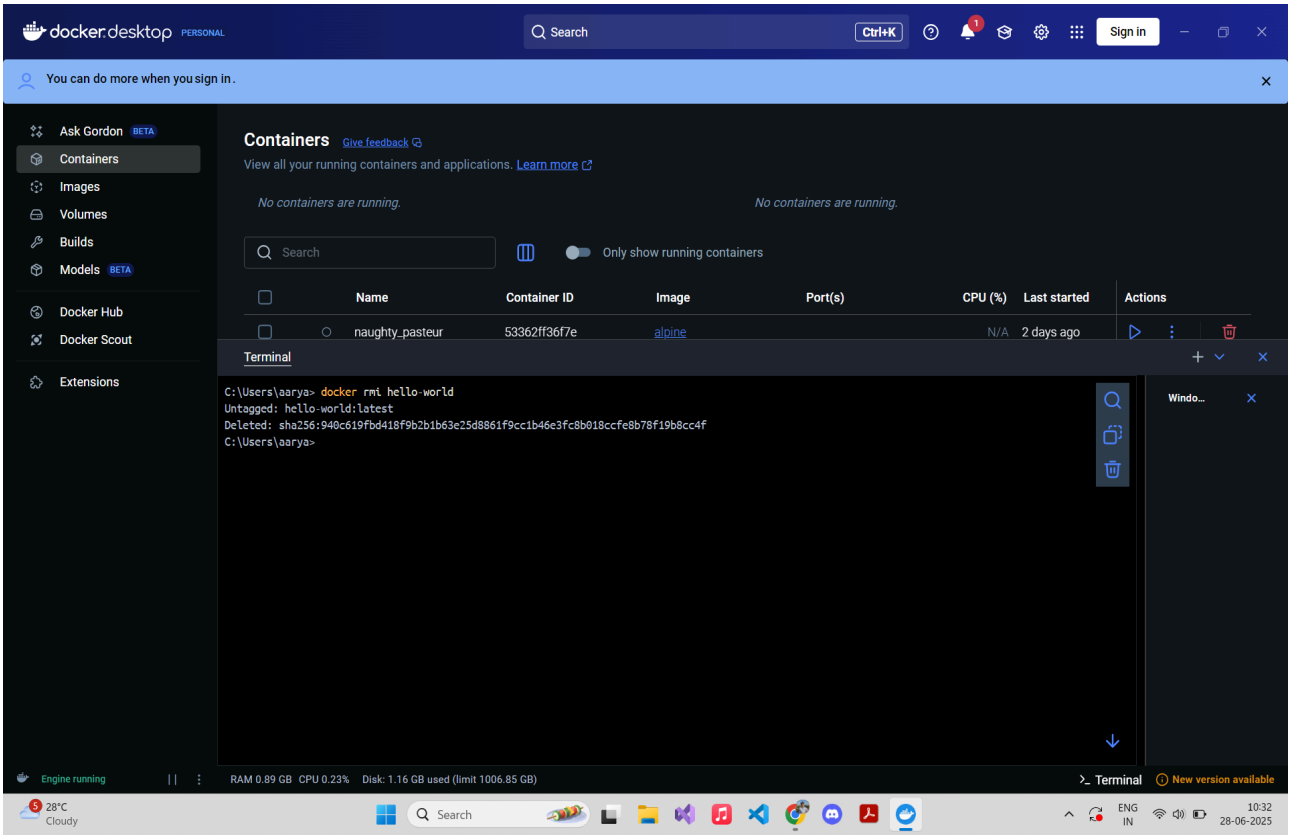
- List Running Containers
- List All Containers (Running + Stopped)



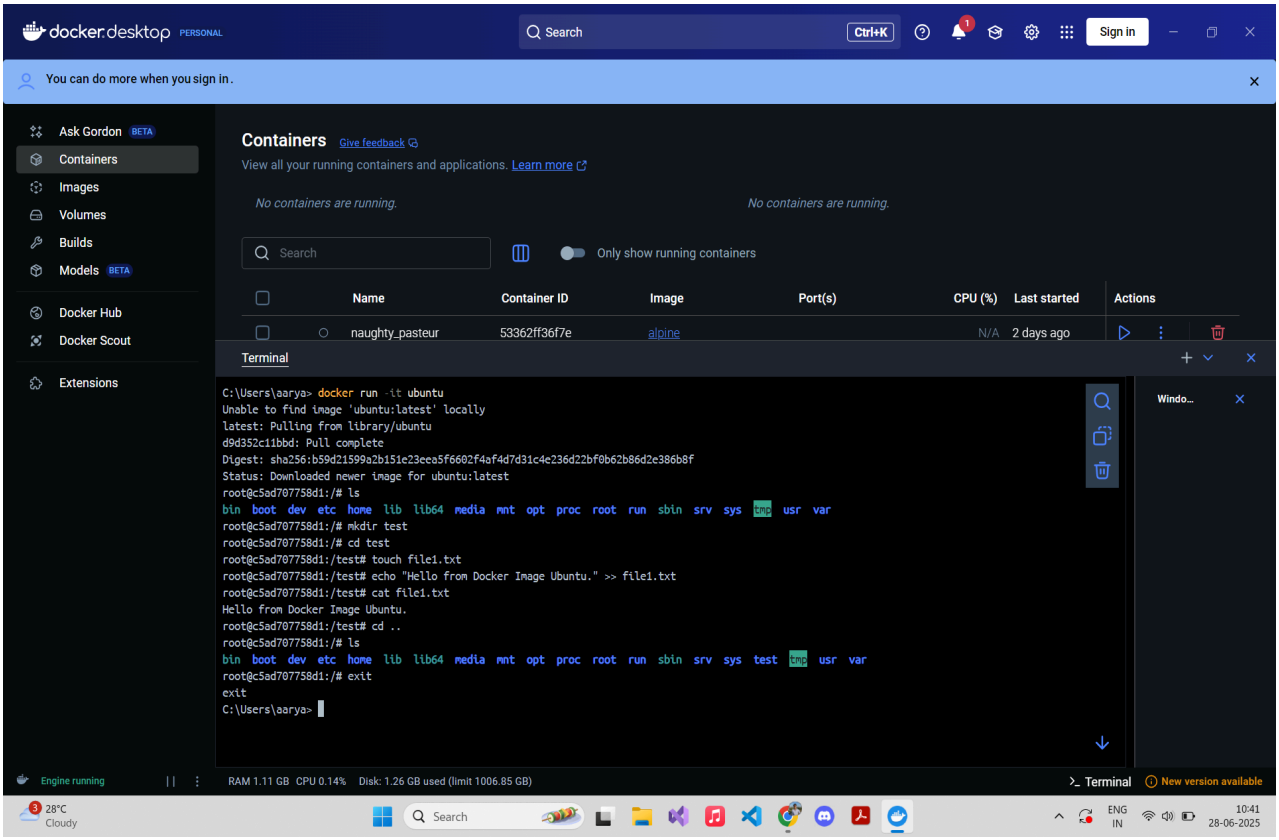
- Remove a Container



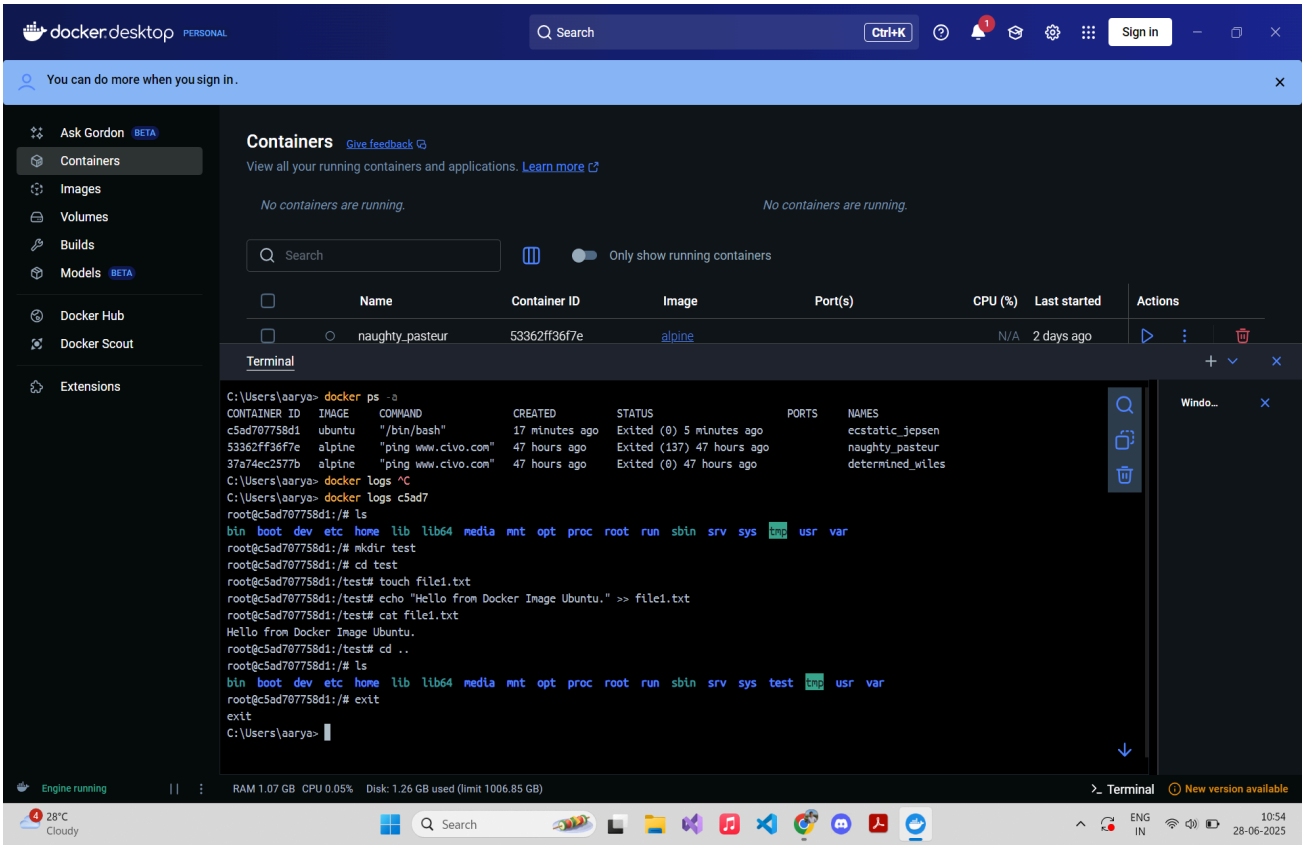
- Remove an Image



- Create and Run Interactive Ubuntu Container

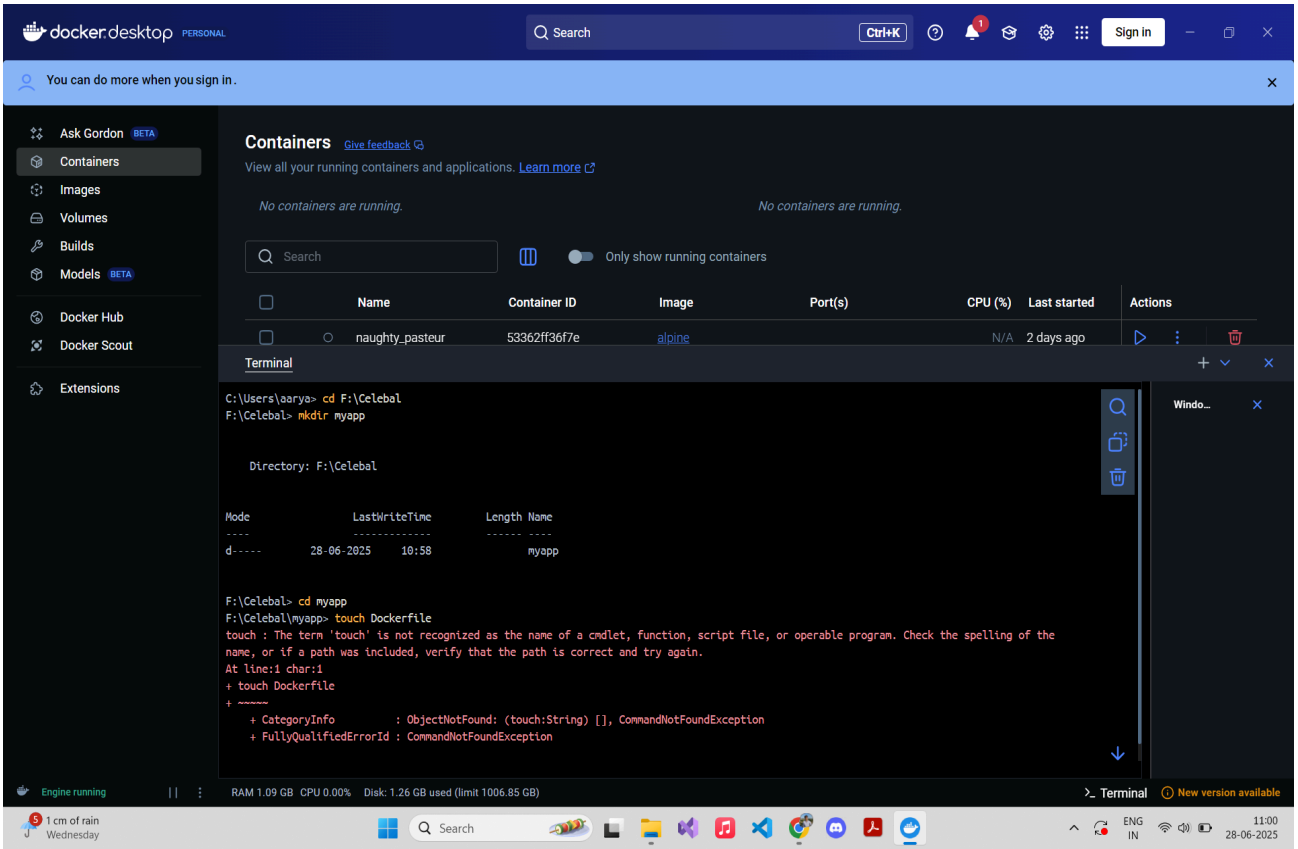


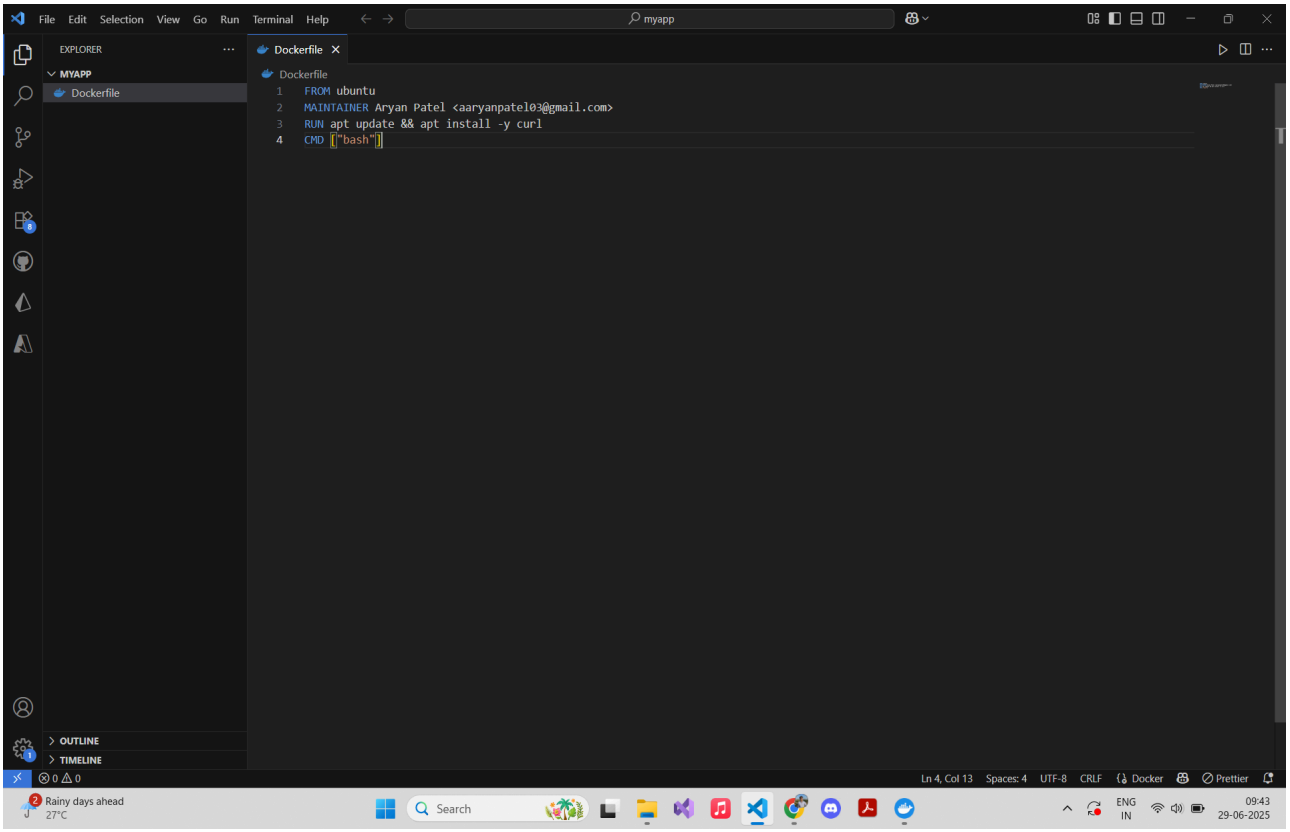
4. Basic Container Operations.

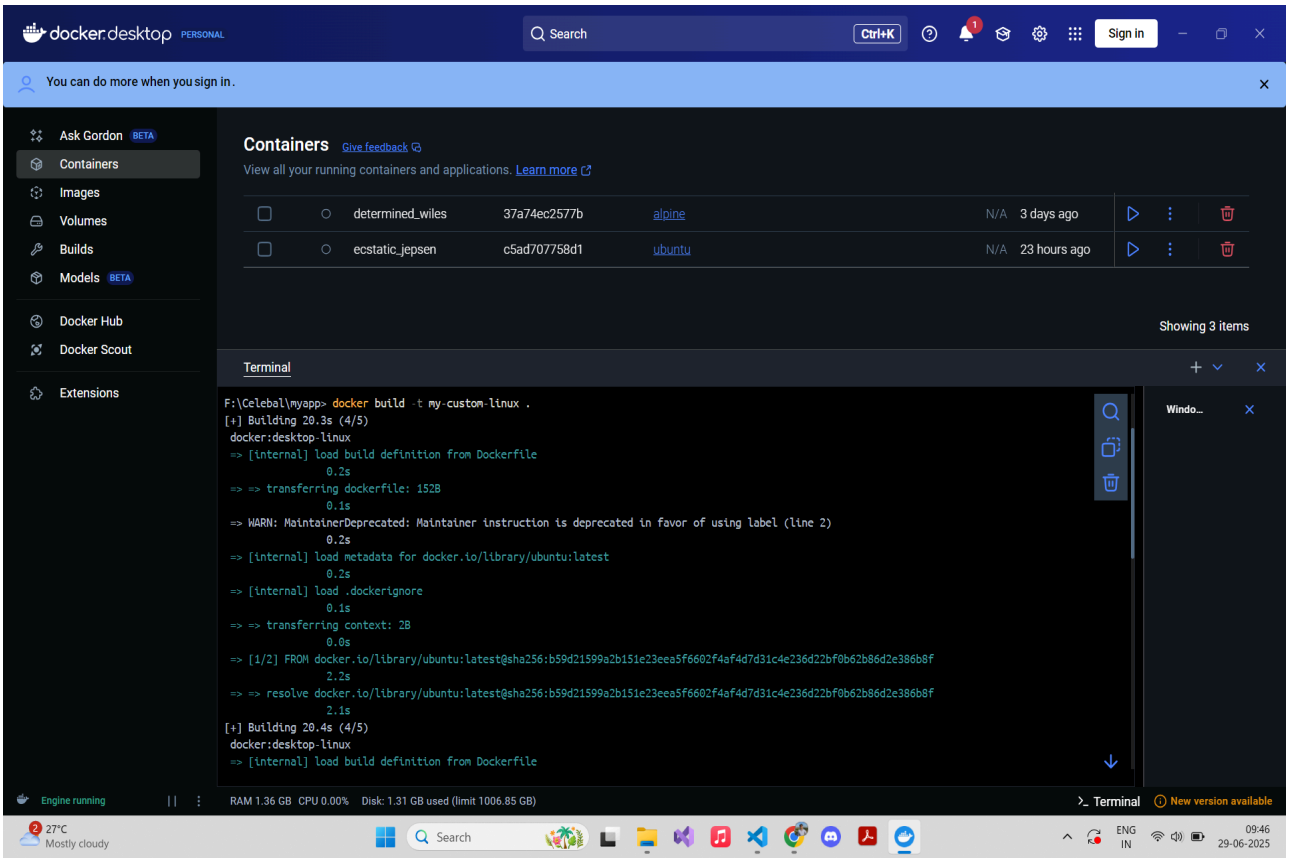


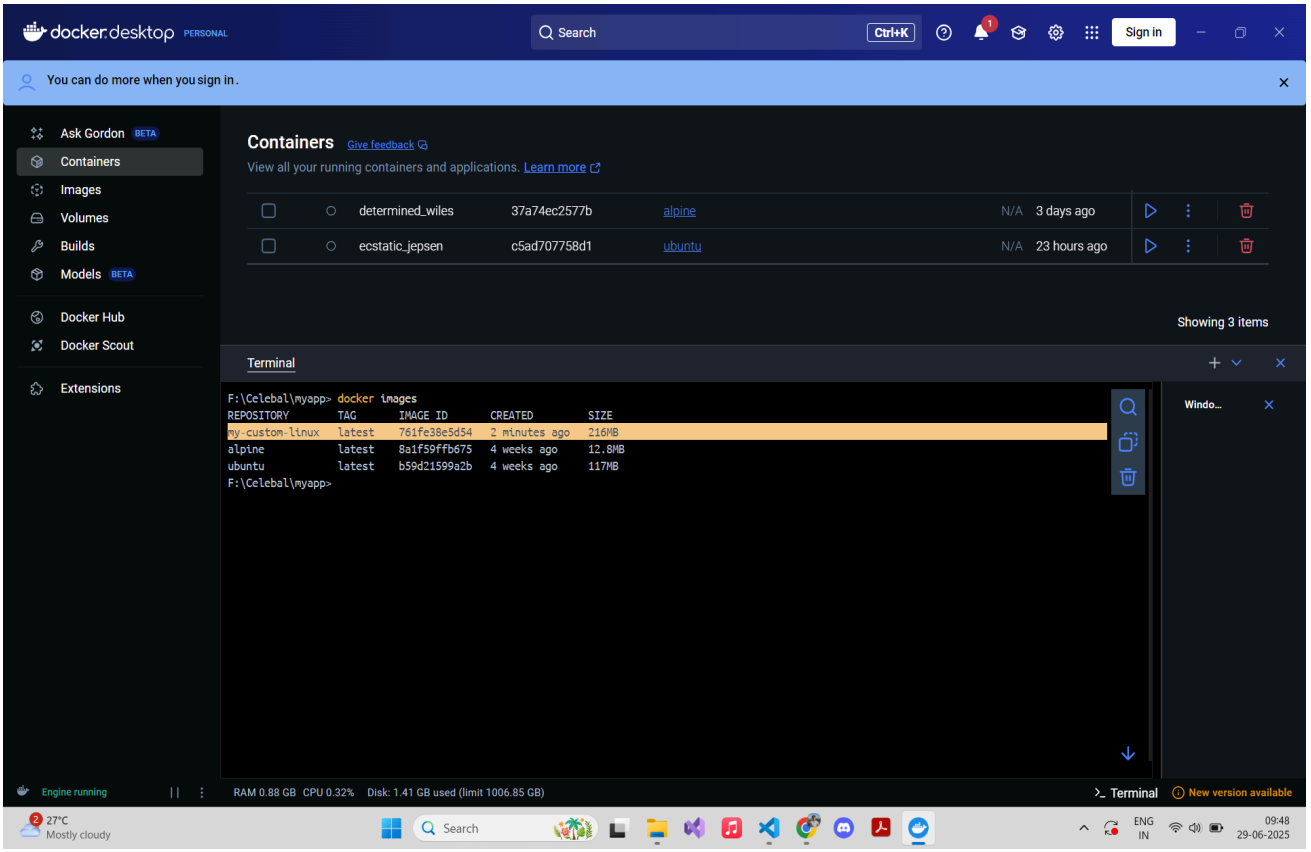
5. Build an Image from a Dockerfile.

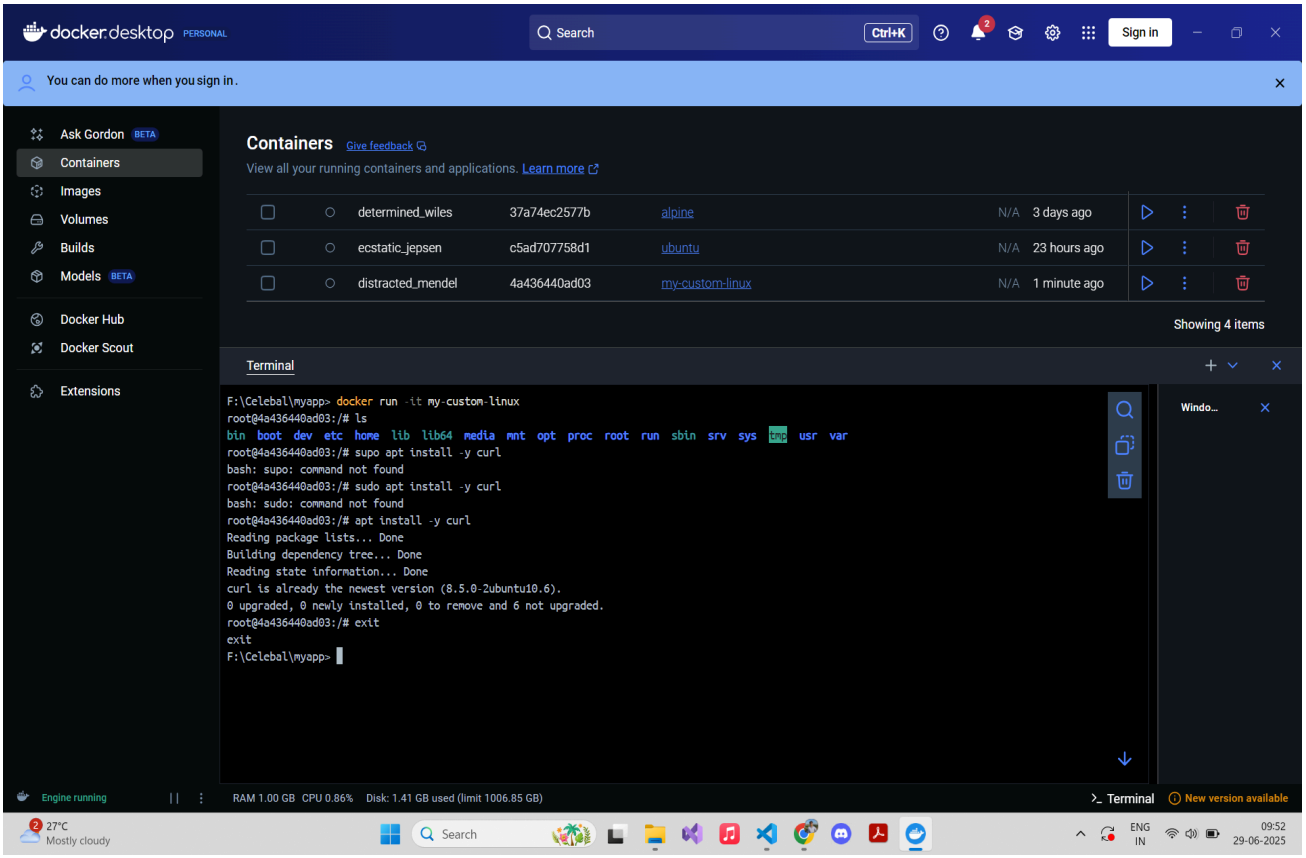
- Create a folder “myapp”, and inside it, create a file named Dockerfile.




File Edit Selection View Go Run Terminal Help
myapp
EXPLORER
MYAPP
Dockerfile
Dockerfile
FROM ubuntu
MAINTAINER Aryan Patel <aaryanpatel03@gmail.com>
RUN apt update && apt install -y curl
CMD ["bash"]
OUTLINE
TIMELINE
Ln 4, Col 13 Spaces: 4 UTF-8 CRLF Docker Prettier
Rainy days ahead
27°C
Search
ENG
IN
09:43
29-06-2025


docker desktop PERSONAL
Search Ctrl+K
Sign in
You can do more when you sign in.
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
determined_wiles 37a74ec2577b alpine N/A 3 days ago
ecstatic_jepsen c5ad707758d1 ubuntu N/A 23 hours ago
Showing 3 items
Terminal
F:\Celebal\myapp> docker build -t my-custom-linux .
[+] Building 20.3s (4/5)
docker:desktop-linux
=> [internal] load build definition from Dockerfile
0.2s
=> => transferring dockerfile: 152B
0.1s
=> WARN: MaintainerDeprecated: Maintainer instruction is deprecated in favor of using label (line 2)
0.2s
=> [internal] load metadata for docker.io/library/ubuntu:latest
0.2s
=> [internal] load .dockerignore
0.1s
=> => transferring context: 2B
0.0s
=> [1/2] FROM docker.io/library/ubuntu:latest@sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
2.2s
=> => resolve docker.io/library/ubuntu:latest@sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
2.1s
[+] Building 20.4s (4/5)
docker:desktop-linux
=> [internal] load build definition from Dockerfile
Windo...
Engine running
RAM 1.36 GB CPU 0.00% Disk: 1.31 GB used (limit 1006.85 GB)
Terminal
New version available
27°C
Mostly cloudy
Search
ENG
IN
09:46
29-06-2025


docker desktop PERSONAL
Search Ctrl+K
Sign in
You can do more when you sign in.
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
determined_wiles 37a74ec2577b alpine N/A 3 days ago
ecstatic_jepsen c5ad707758d1 ubuntu N/A 23 hours ago
Showing 3 items
Terminal
F:\Celebal\myapp> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
my-custom-linux latest 761fe38e5d54 2 minutes ago 216MB
alpine latest 8a1f59ffb675 4 weeks ago 12.8MB
ubuntu latest b59d21599a2b 4 weeks ago 117MB
F:\Celebal\myapp>
Engine running
RAM 0.88 GB CPU 0.32% Disk: 1.41 GB used (limit 1006.85 GB)
Terminal
New version available
27°C
Mostly cloudy
Search
ENG
IN
09:48
29-06-2025

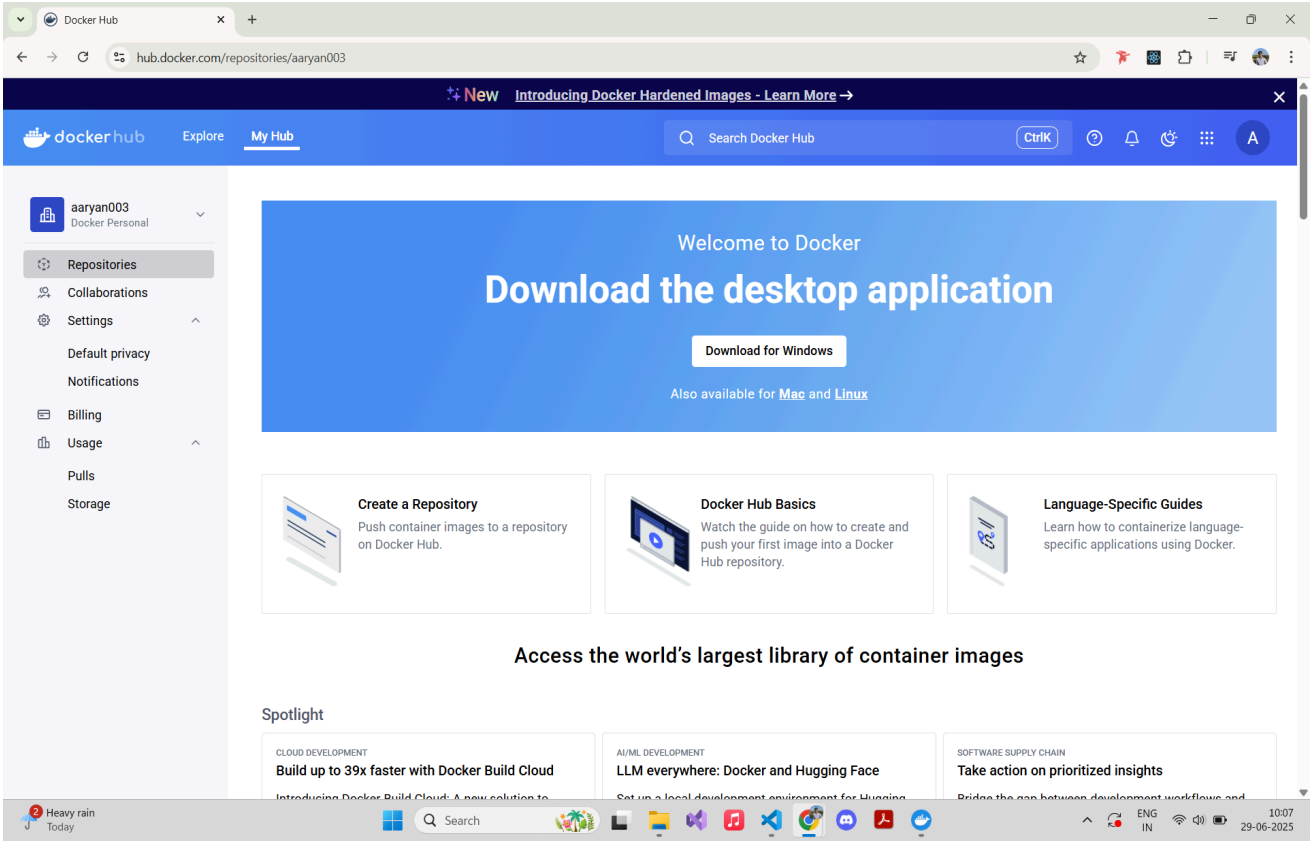

docker desktop PERSONAL
Search Ctrl+K
Sign in
You can do more when you sign in.
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
determined_wiles 37a74ec2577b alpine N/A 3 days ago
ecstatic_jepsen c5ad707758d1 ubuntu N/A 23 hours ago
distracted_mendel 4a436440ad03 my-custom-linux N/A 1 minute ago
Showing 4 items
Terminal
F:\Celebal\myapp> docker run -it my-custom-linux
root@4a436440ad03:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@4a436440ad03:/# supo apt install -y curl
bash: supo: command not found
root@4a436440ad03:/# sudo apt install -y curl
bash: sudo: command not found
root@4a436440ad03:/# apt install -y curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
root@4a436440ad03:/# exit
exit
F:\Celebal\myapp>
Engine running
RAM 1.00 GB CPU 0.86% Disk: 1.41 GB used (limit 1006.85 GB)
Terminal
New version available
27°C
Mostly cloudy
Search
ENG
IN
09:52
29-06-2025

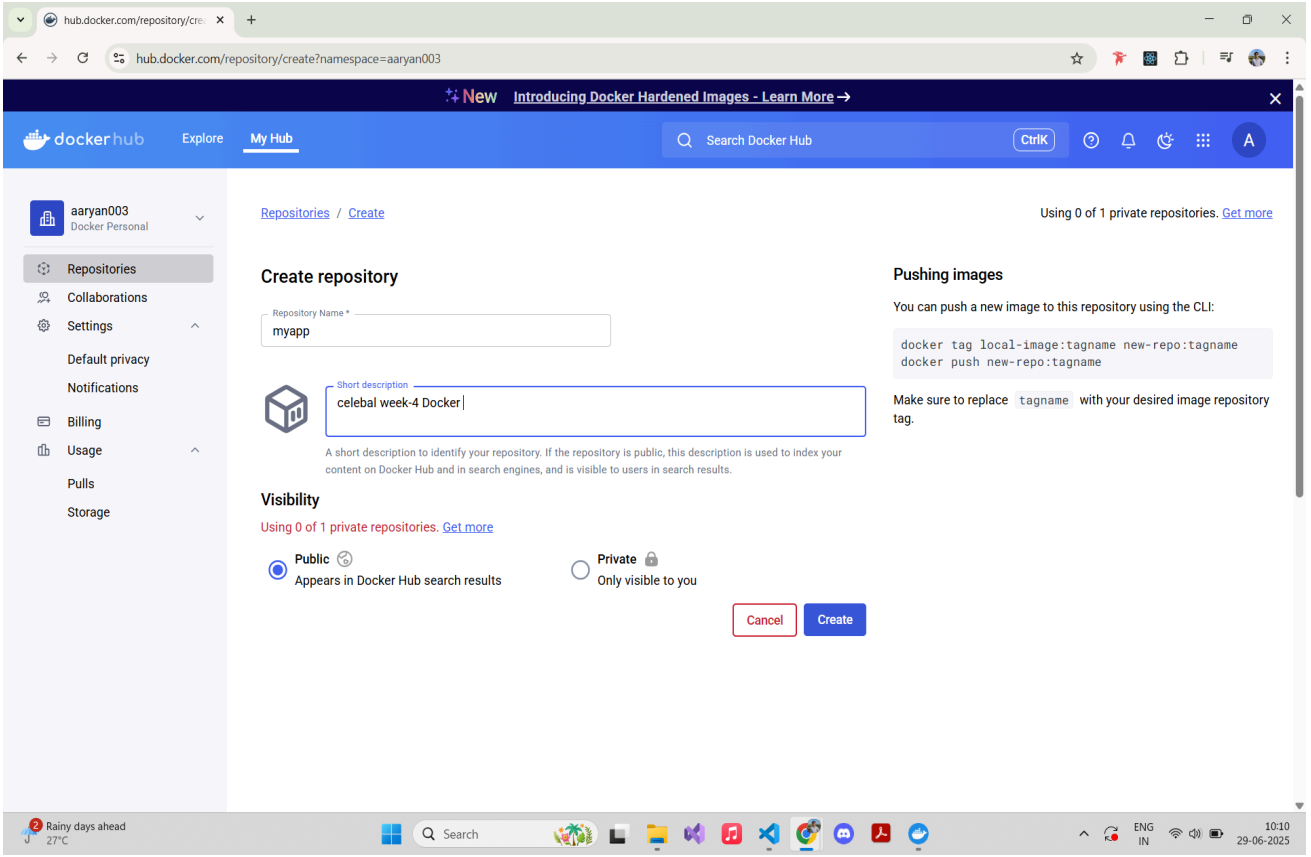
3. Docker Registry, DockerHub, Create a Multi-Stage Build.

Steps :

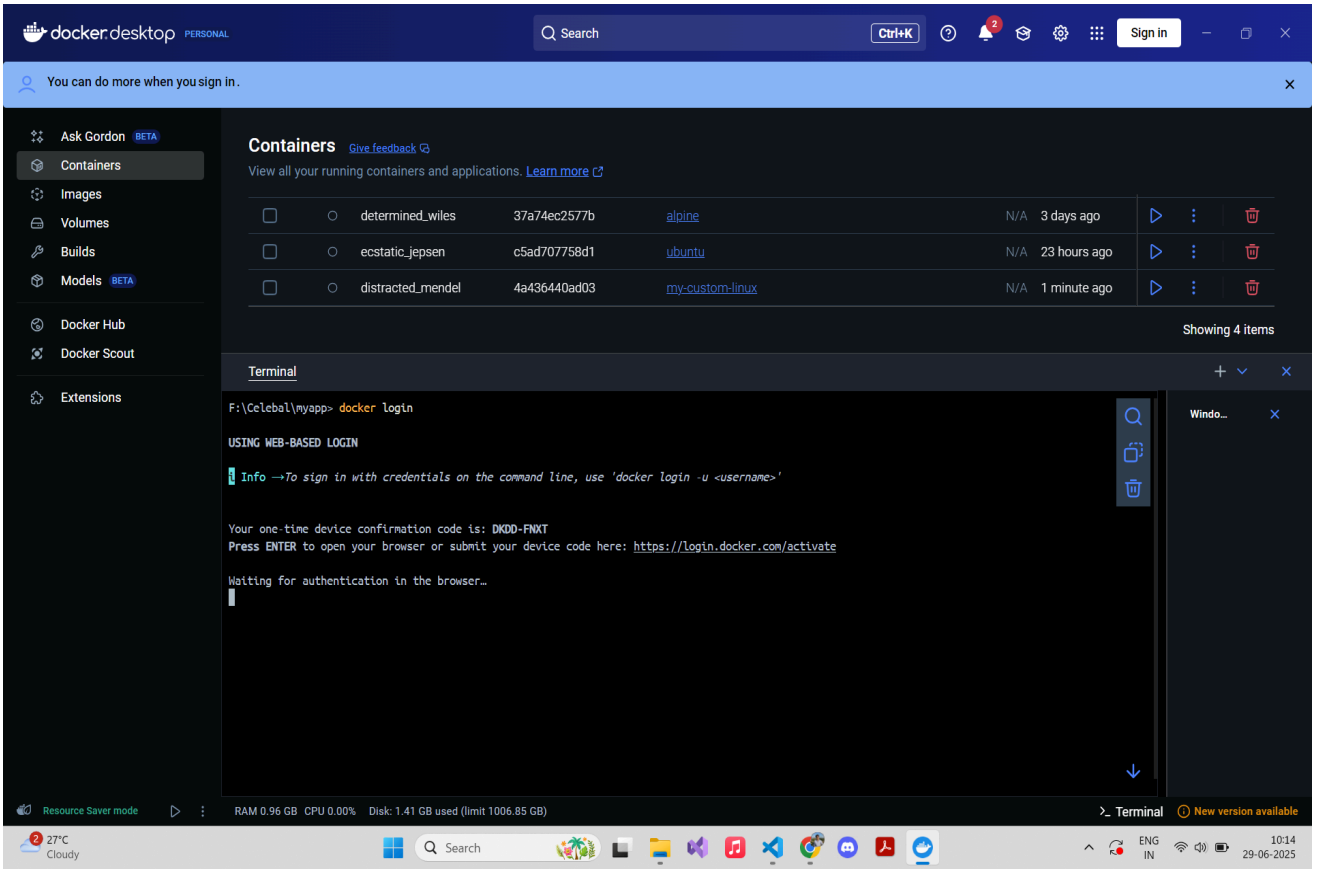
1. Create a Docker Hub account and repository.

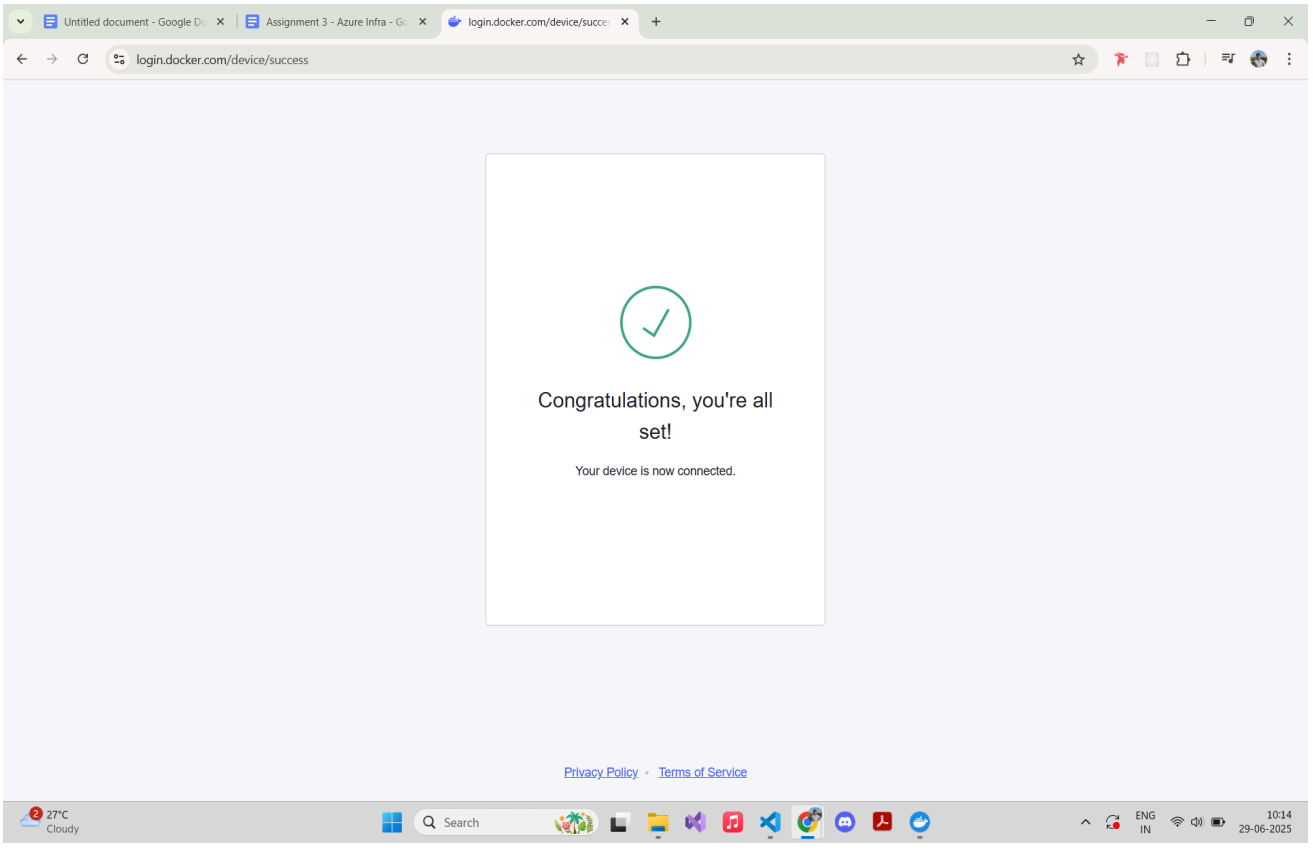
- Click "Create Repository", enter name, visibility (public/private).



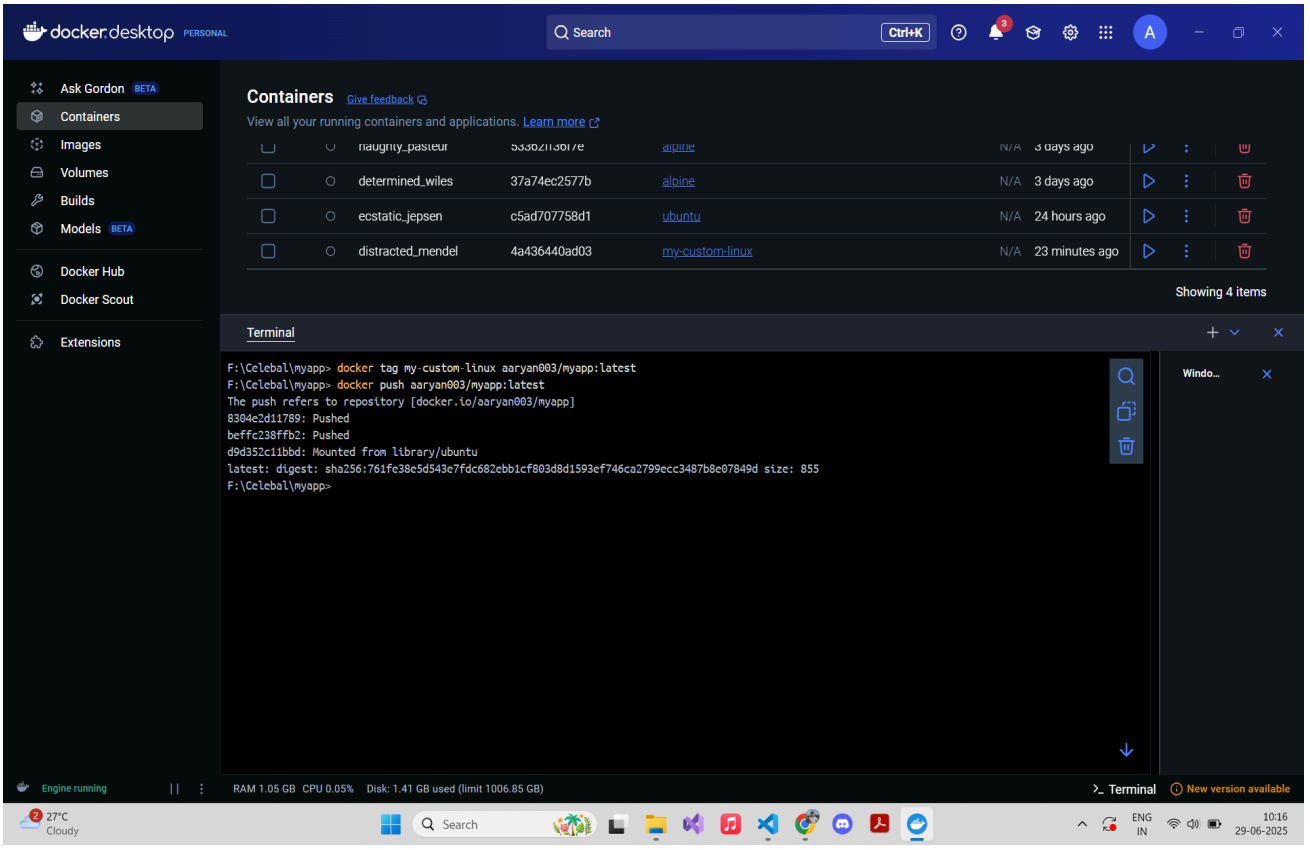


- Login via CLI.

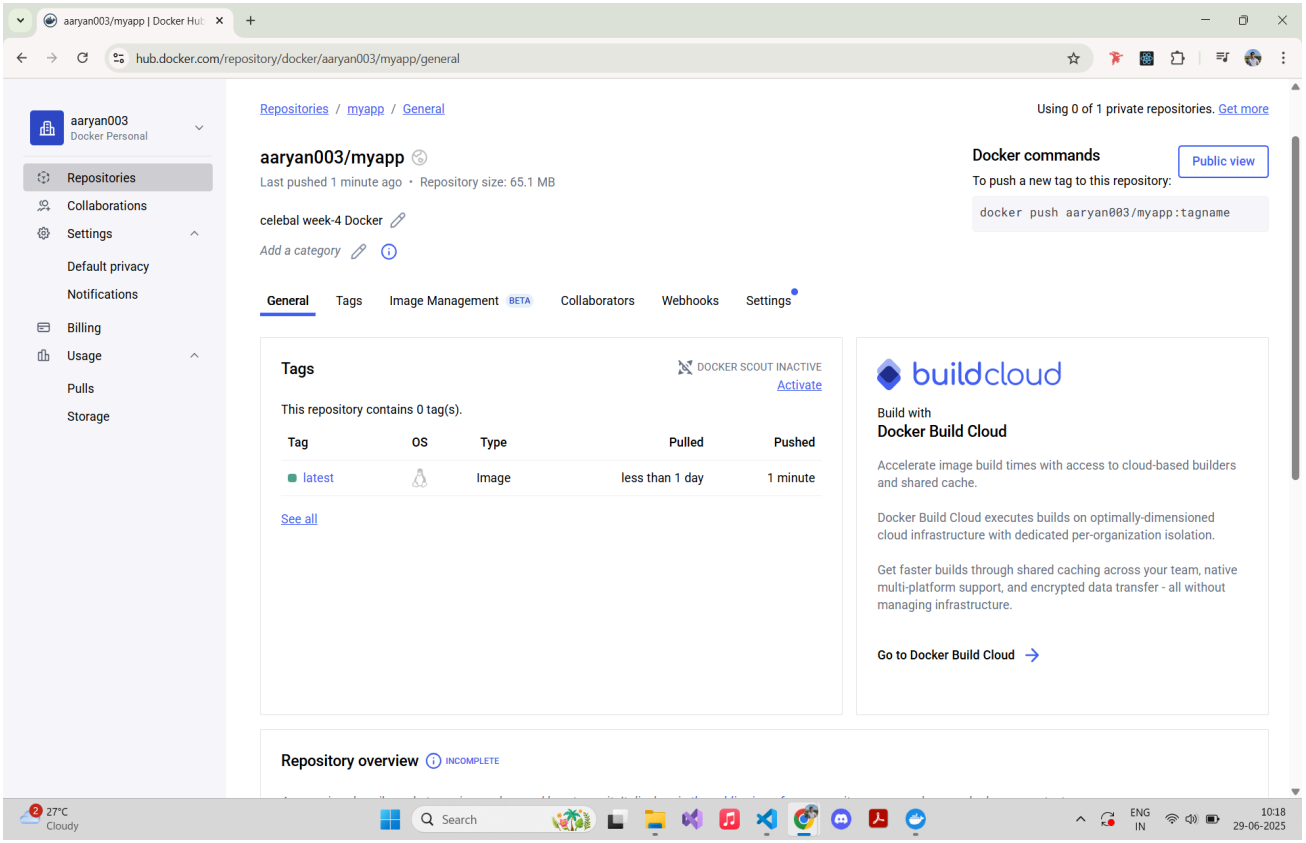




- Tag a local image
- Push image to Docker Hub

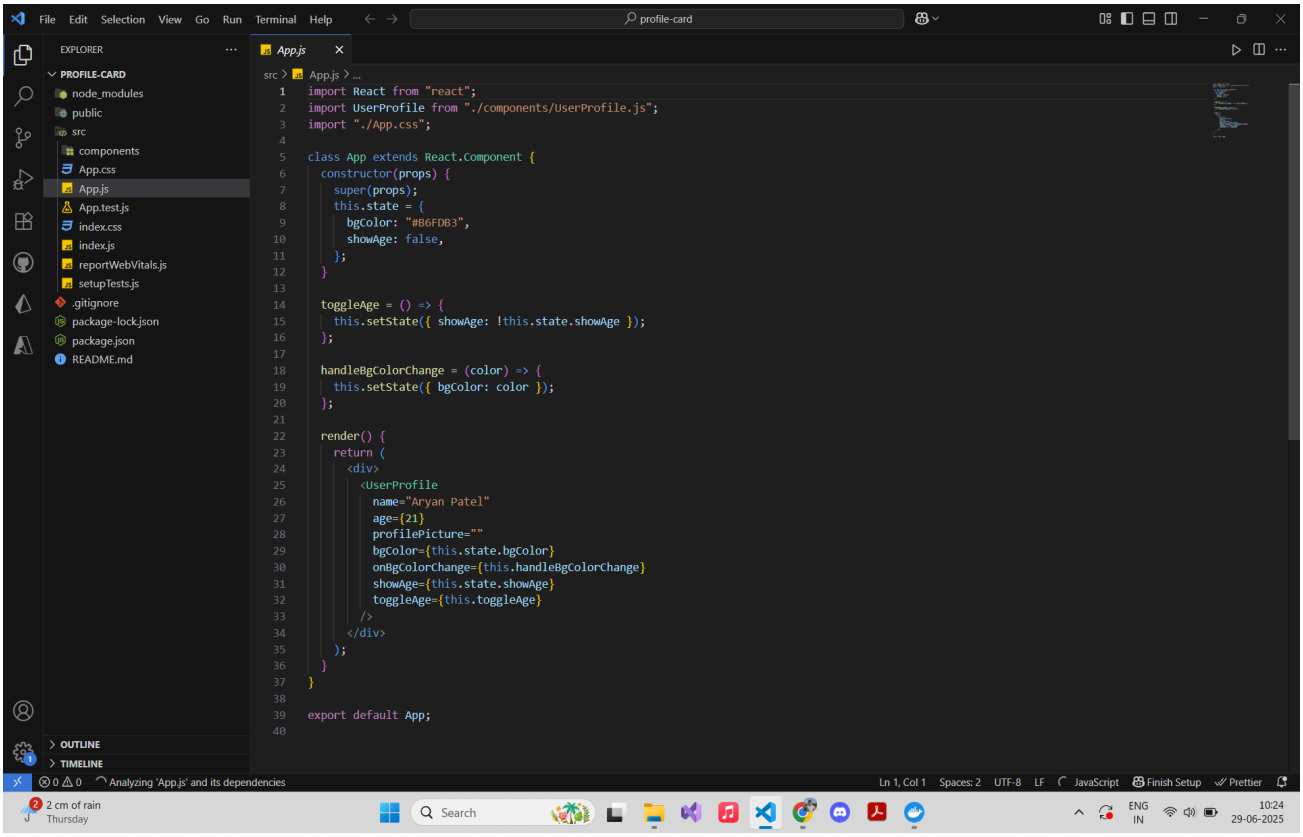


- Verify on Docker Hub



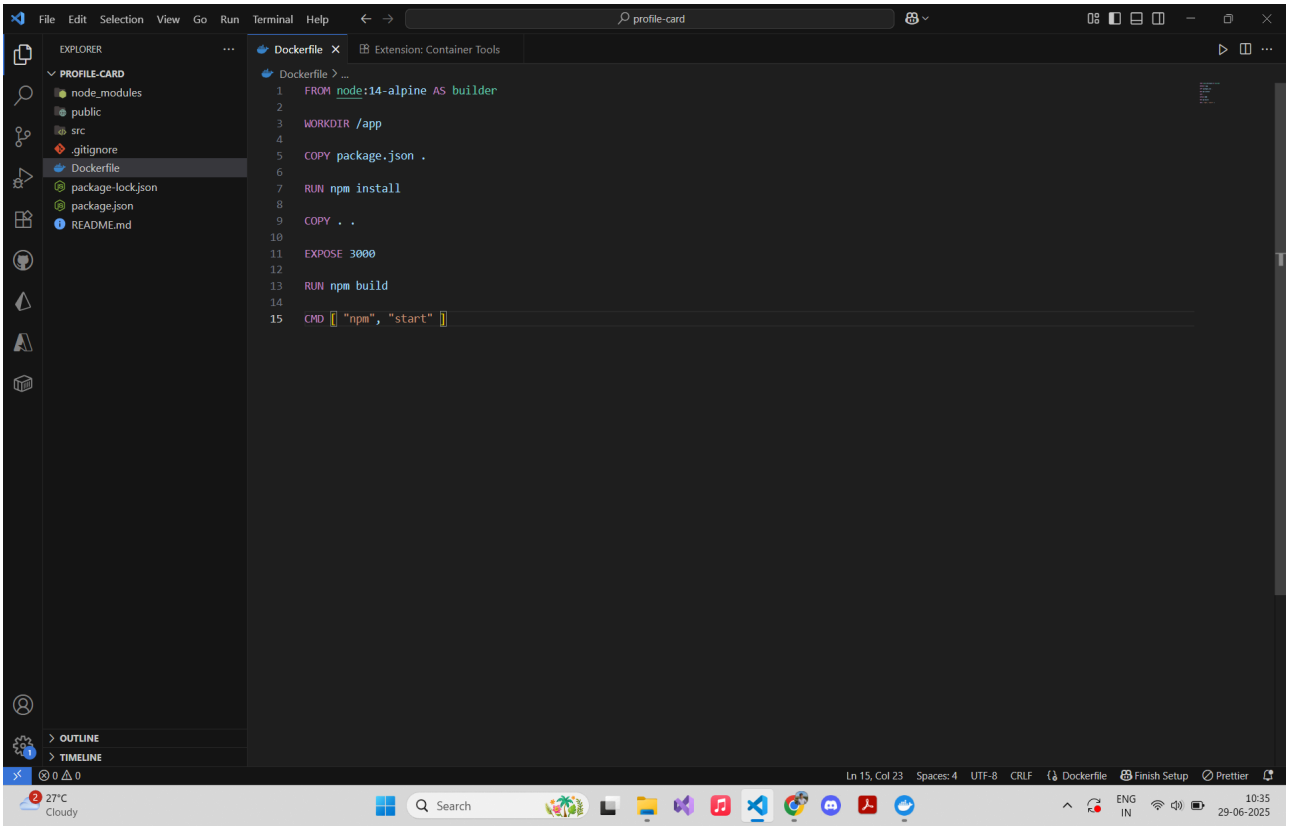
2. Creating a Multi-Stage build.

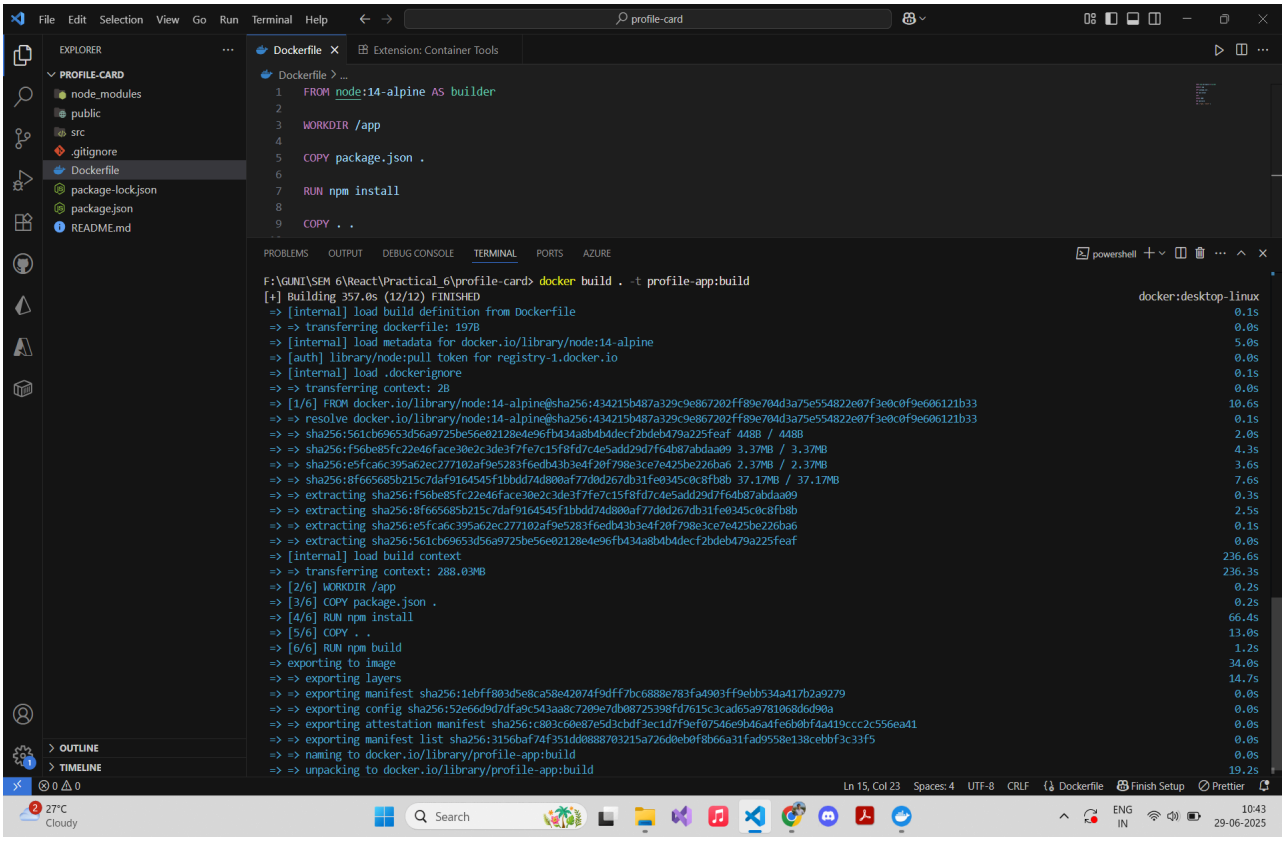
- Create a Sample React App



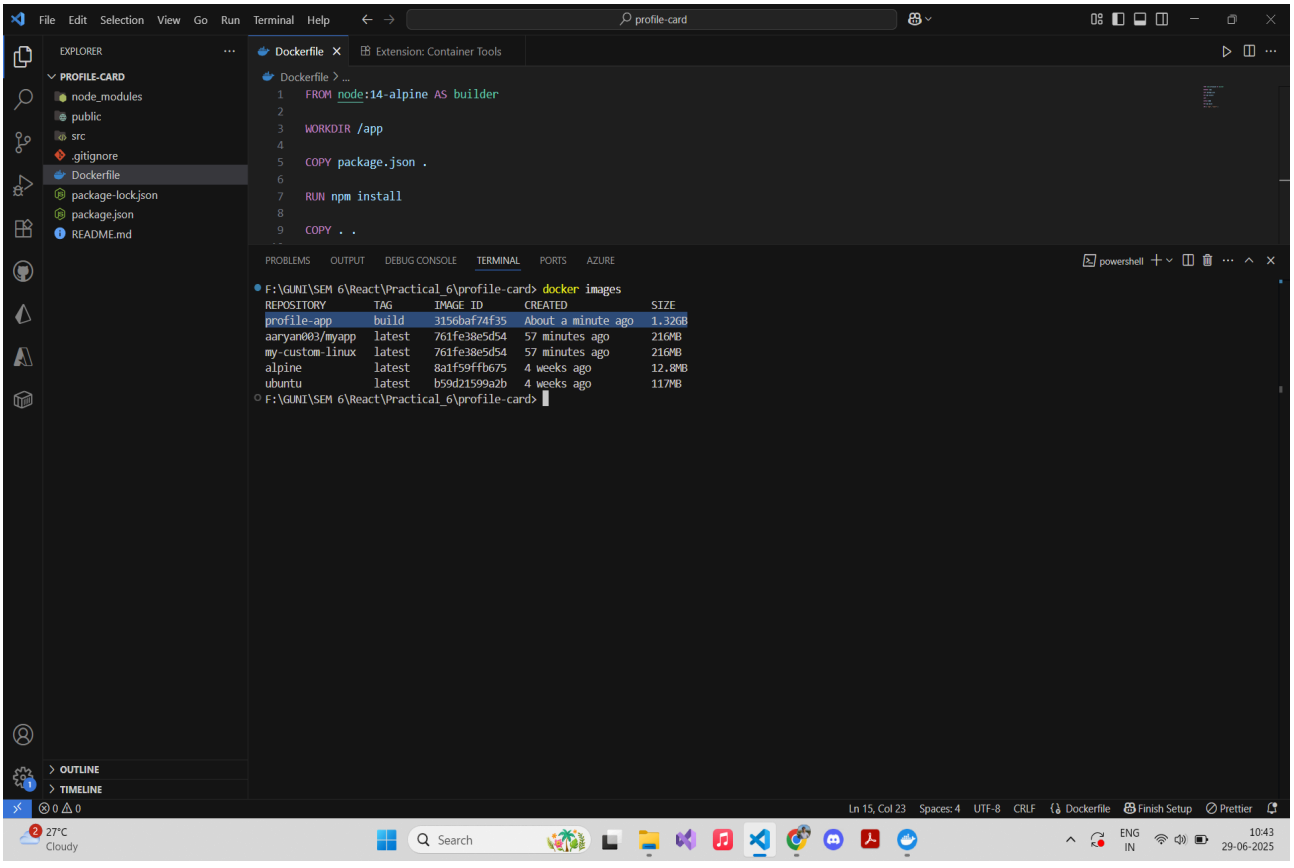
- Before using multi-stage build :

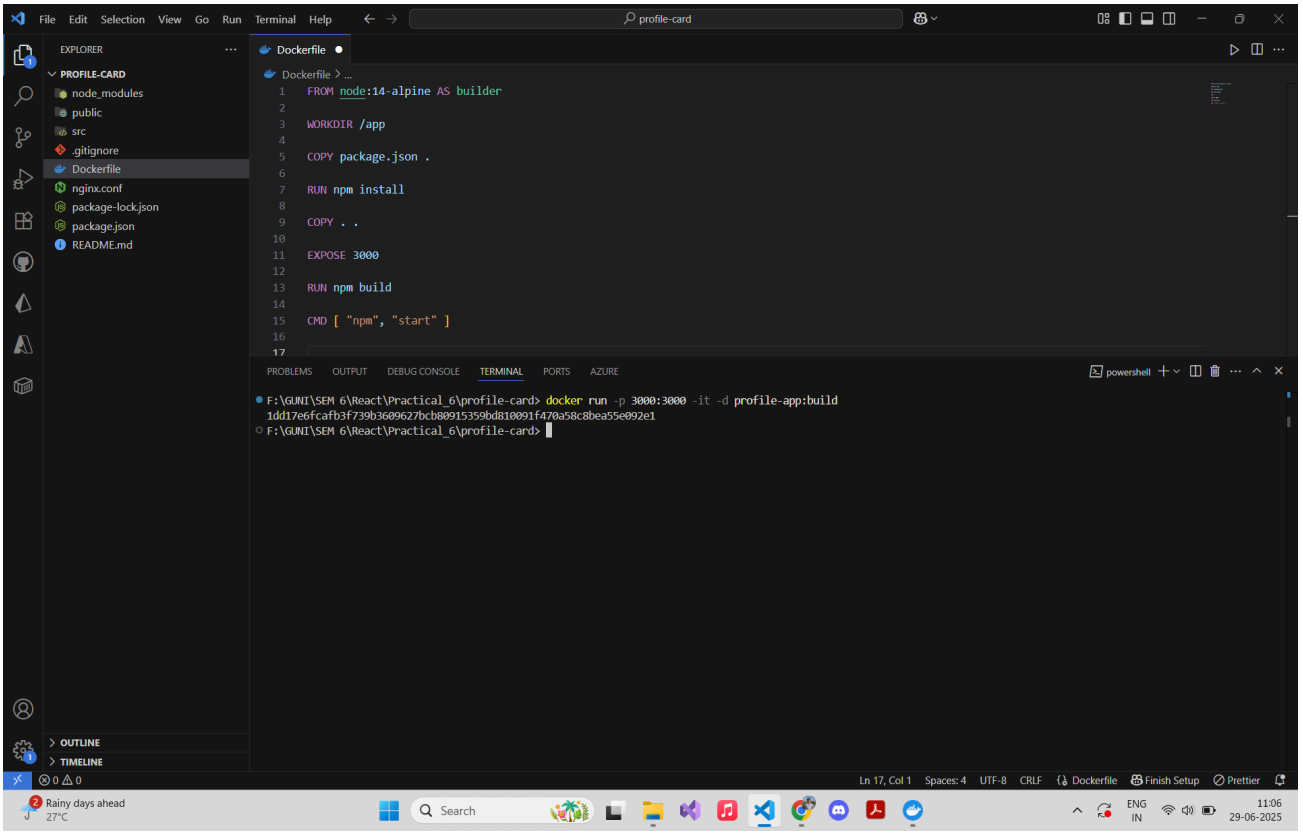
Dockerfile :

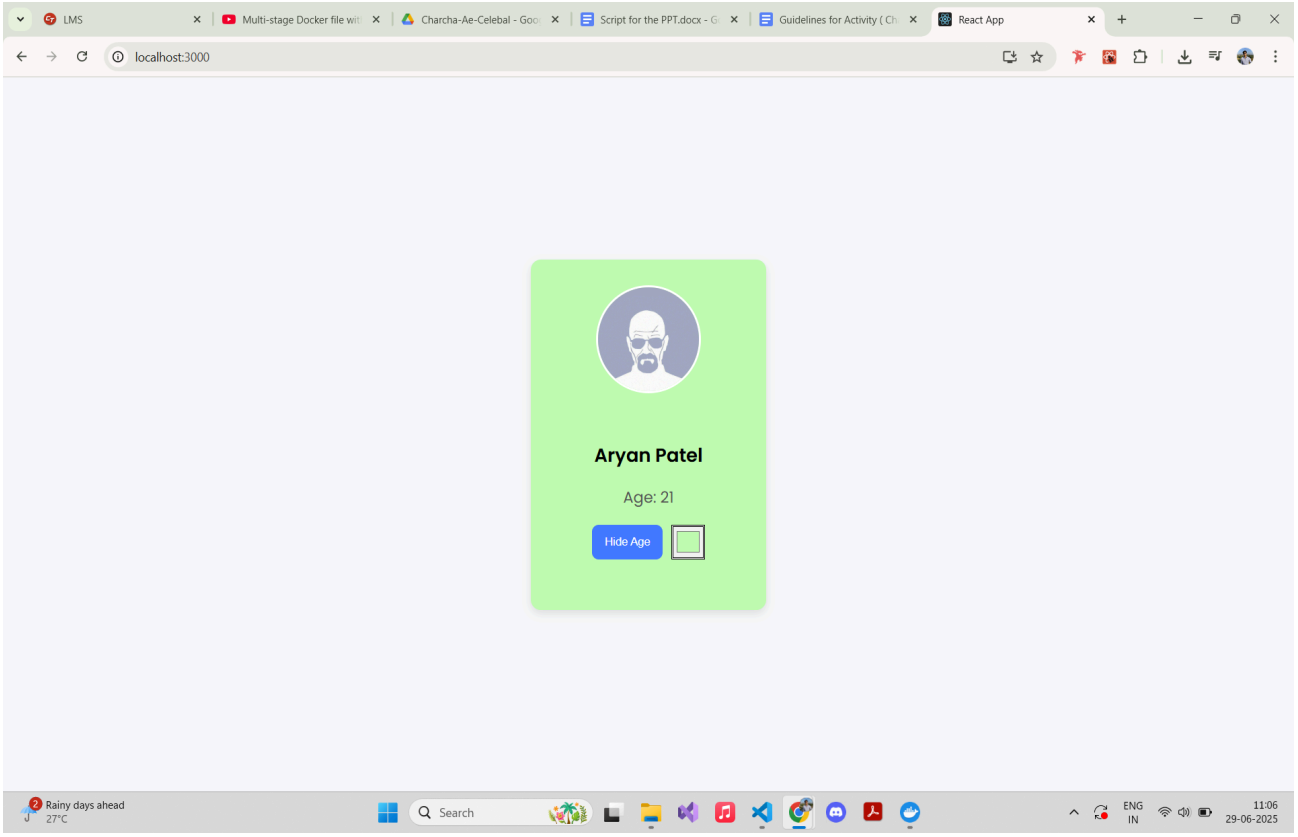




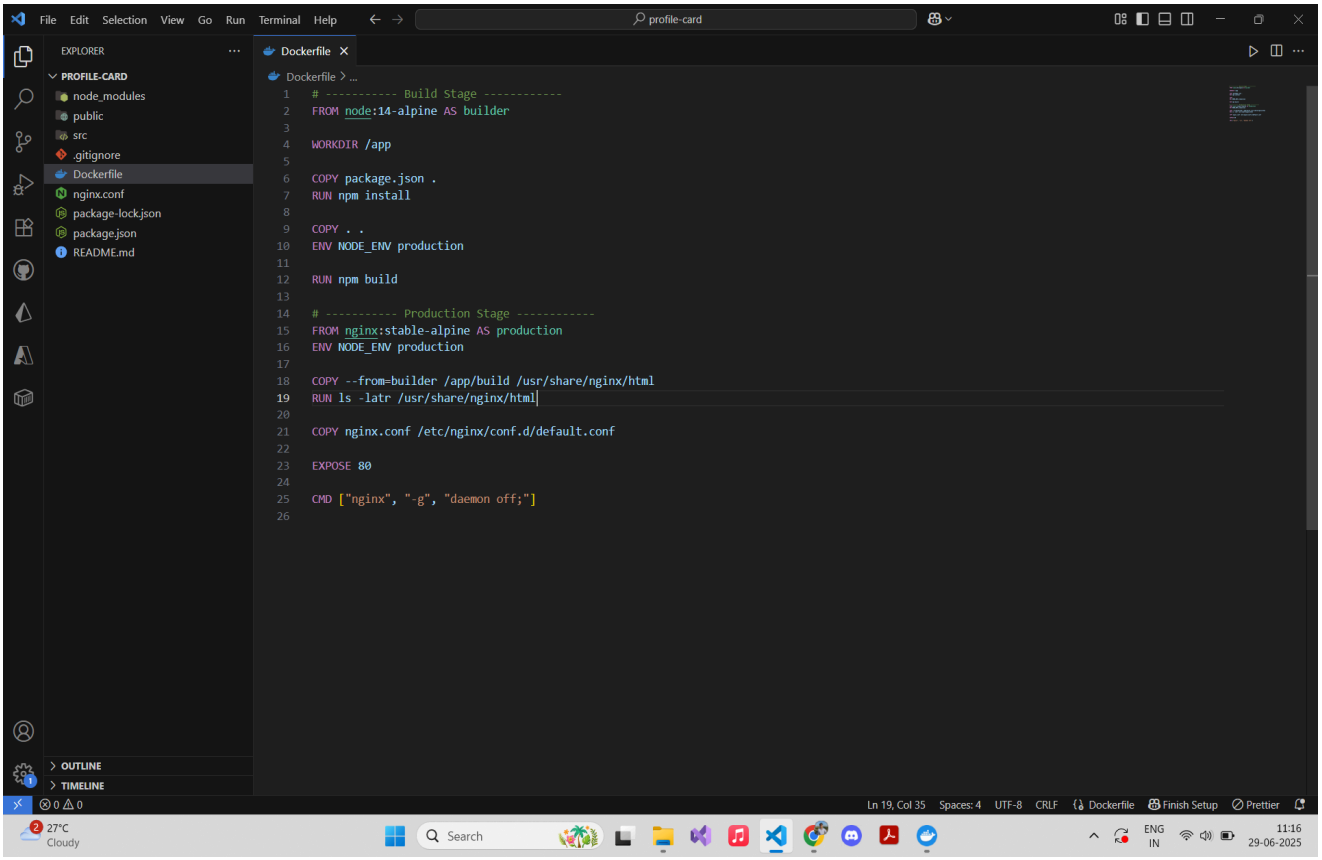
- You can see the size of this docker image “profile-app” which is very large before using multi-stage build.



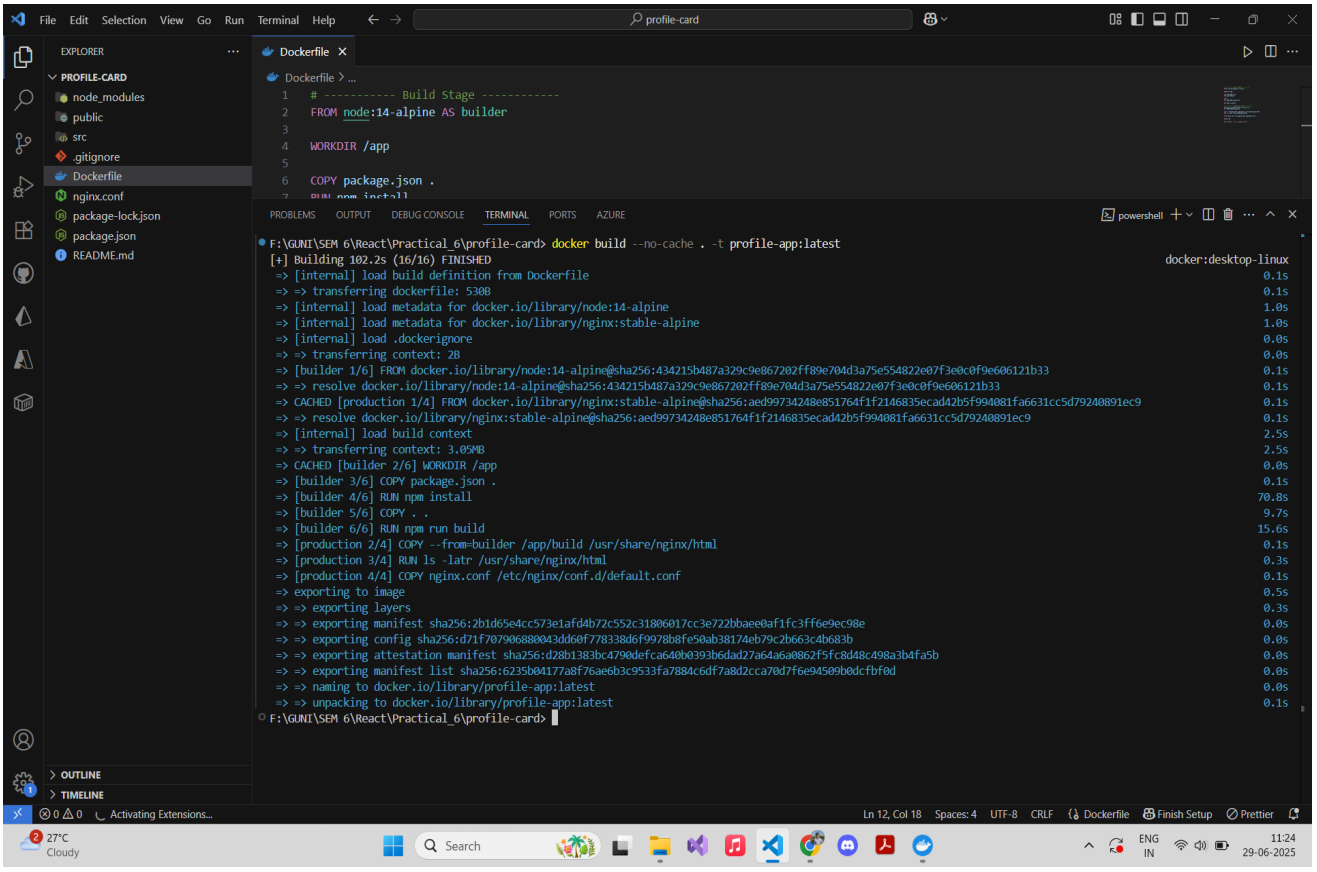

File Edit Selection View Go Run Terminal Help
profile-card
EXPLORER
PROFILE-CARD
node_modules
public
src
.gitignore
Dockerfile
nginx.conf
package-lock.json
package.json
README.md
Dockerfile
Dockerfile > ...
1 FROM node:14-alpine AS builder
3 WORKDIR /app
5 COPY package.json .
7 RUN npm install
9 COPY . .
11 EXPOSE 3000
13 RUN npm build
15 CMD ["npm", "start"]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
powershell
F:\GUNI\SEM 6\React\Practical_6\profile-card> docker run -p 3000:3000 -it -d profile-app:build
1dd17e6fcafb3f739b3609627bcb80915359bd810091f470a58c8bea55e092e1
F:\GUNI\SEM 6\React\Practical_6\profile-card>
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 4 UTF-8 CRLF Dockerfile Finish Setup Prettier
Rainy days ahead
27°C
Search
ENG
IN
11:06
29-06-2025


LMS
Multi-stage Docker file wit
Charcha-Ae-Celebal - Goo
Script for the PPT.docx - G
Guidelines for Activity (Ch
React App
localhost:3000
Aryan Patel
Age: 21
Hide Age
Rainy days ahead
27°C
Search
ENG
IN
11:06
29-06-2025

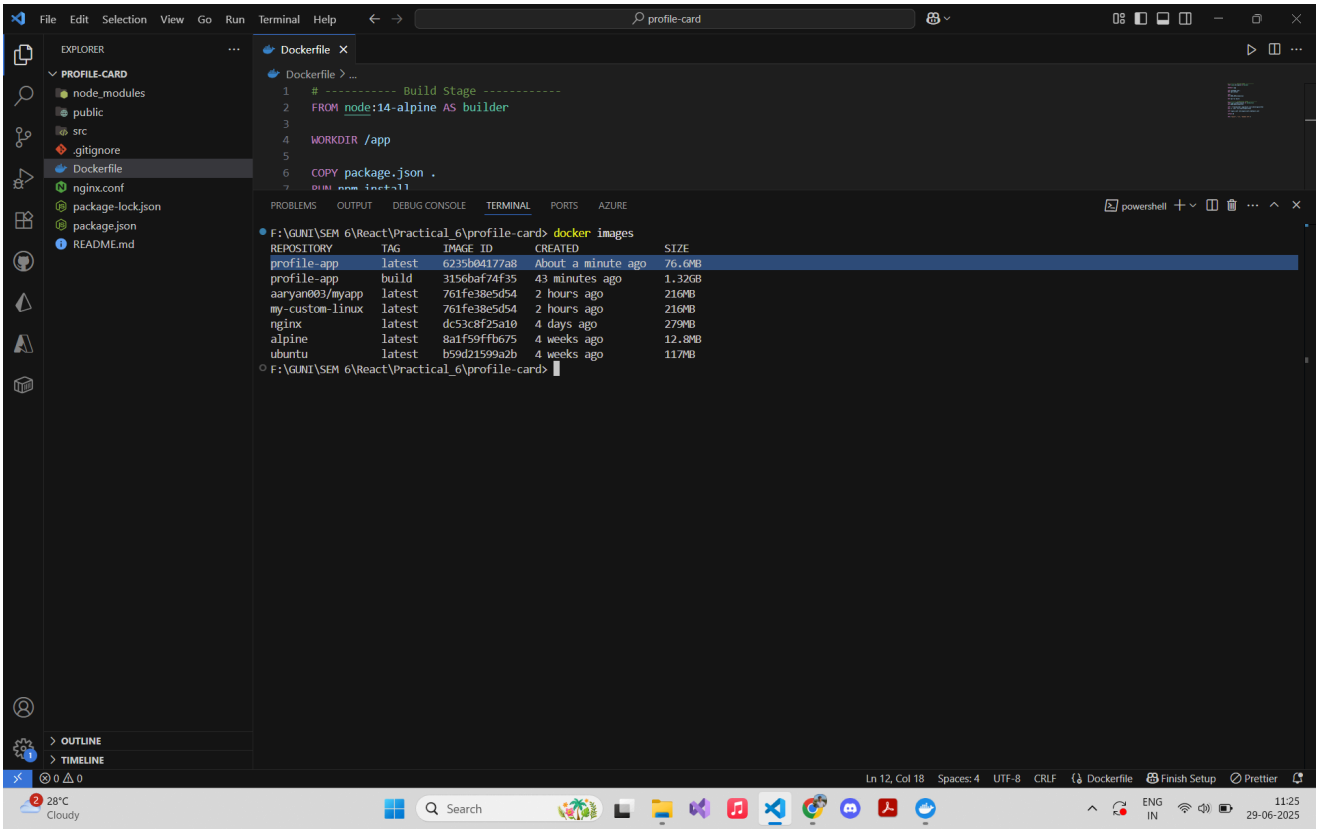
- Now, let’s create a Multi-Stage Dockerfile using React-Nginx.



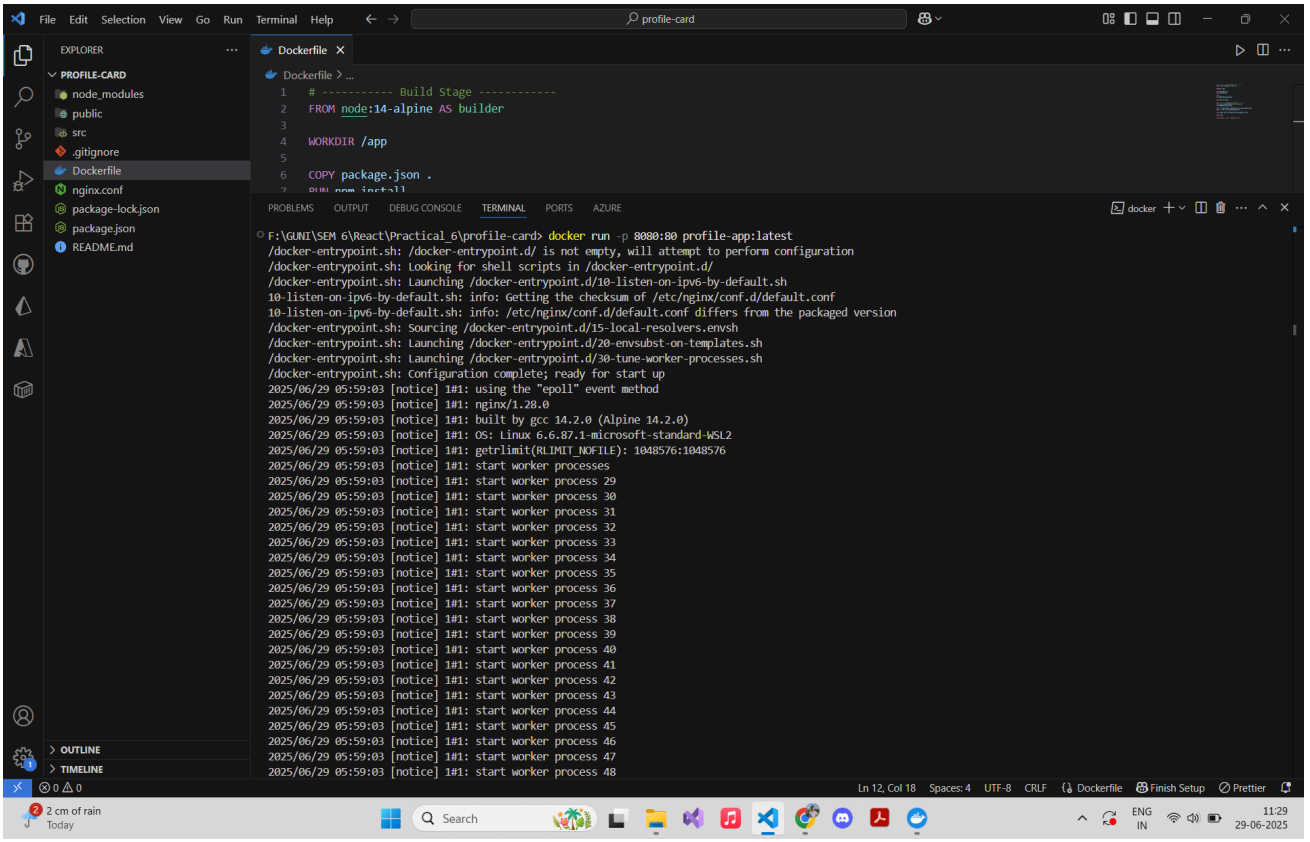
- Build the Docker Image.

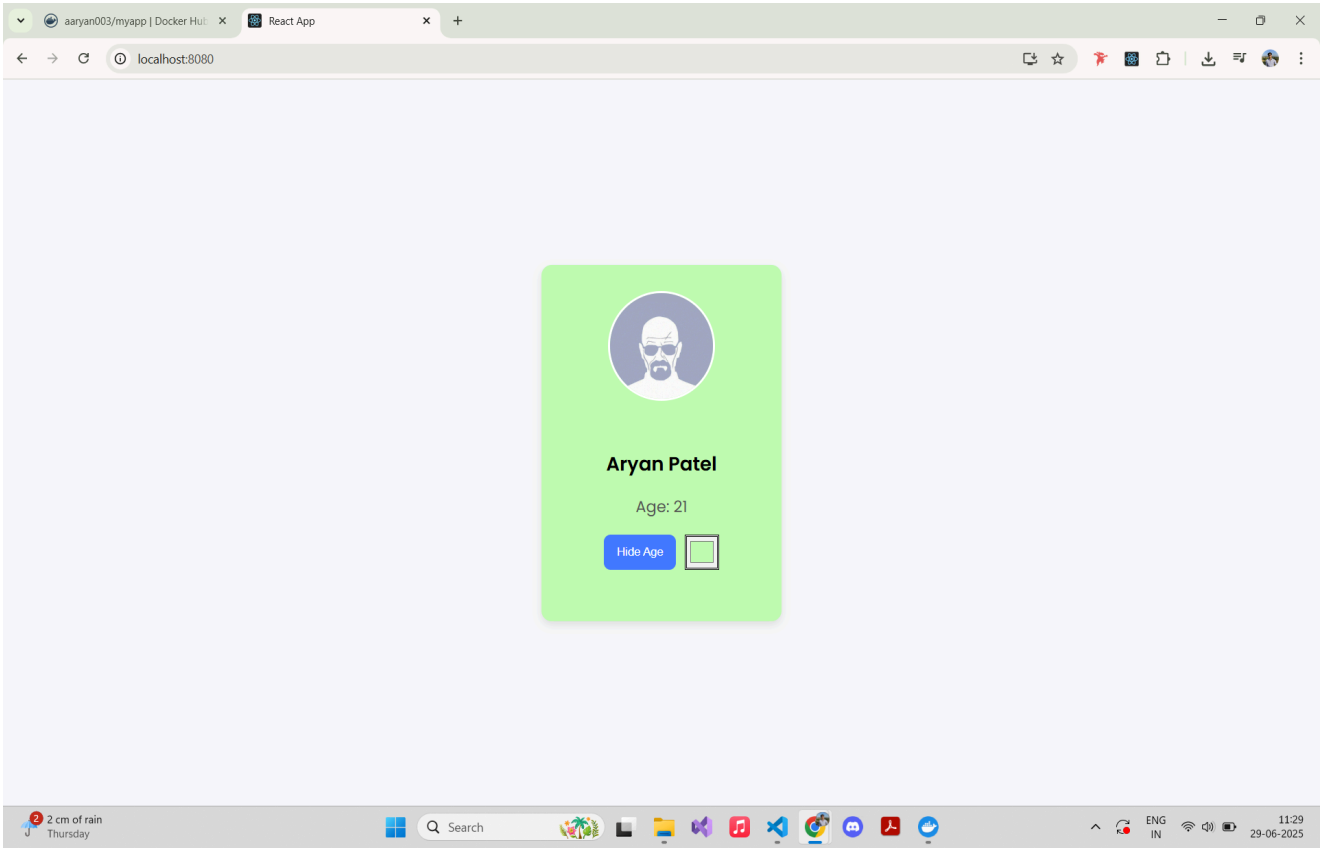


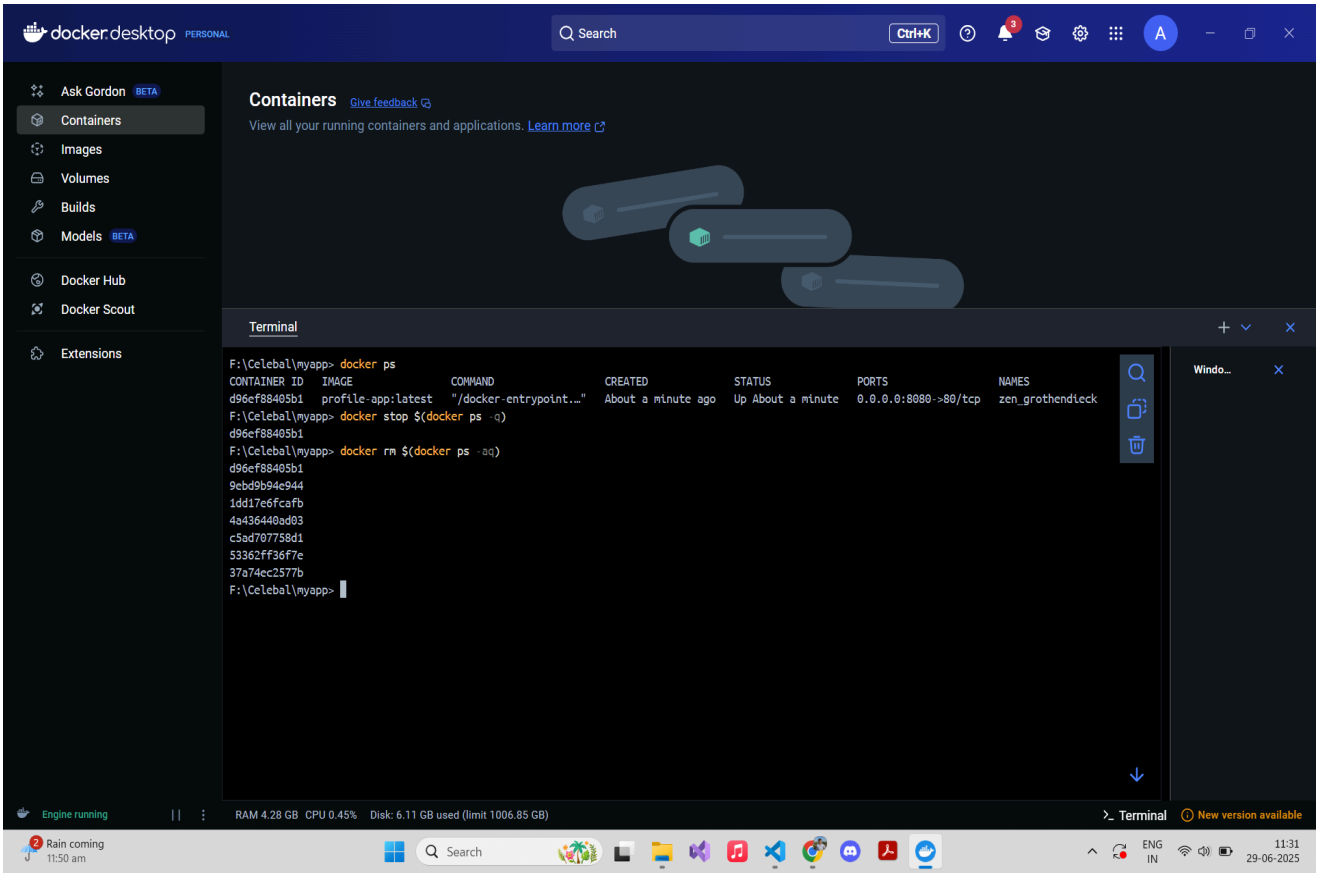
- Now, check the size of the image recently created “profile-app:latest” using ‘docker images’ command.



We can see the image size was reduced to 77MB previously from 1.32GB with the help of **Multi-Stage Build**.

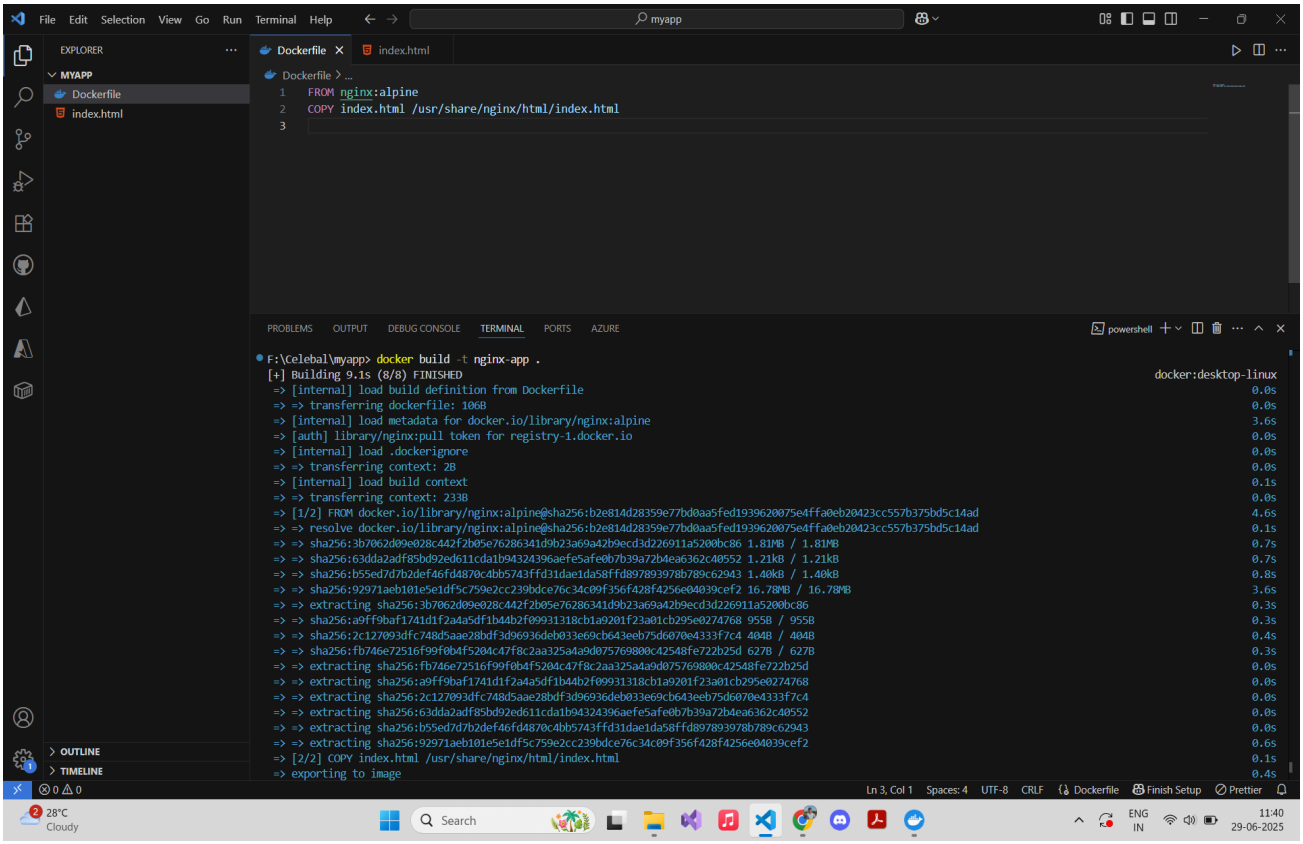



aaryan003/myapp | Docker Hub
React App
localhost:8080
Aryan Patel
Age: 21
Hide Age
2 cm of rain
Thursday
Search
ENG
IN
11:29
29-06-2025

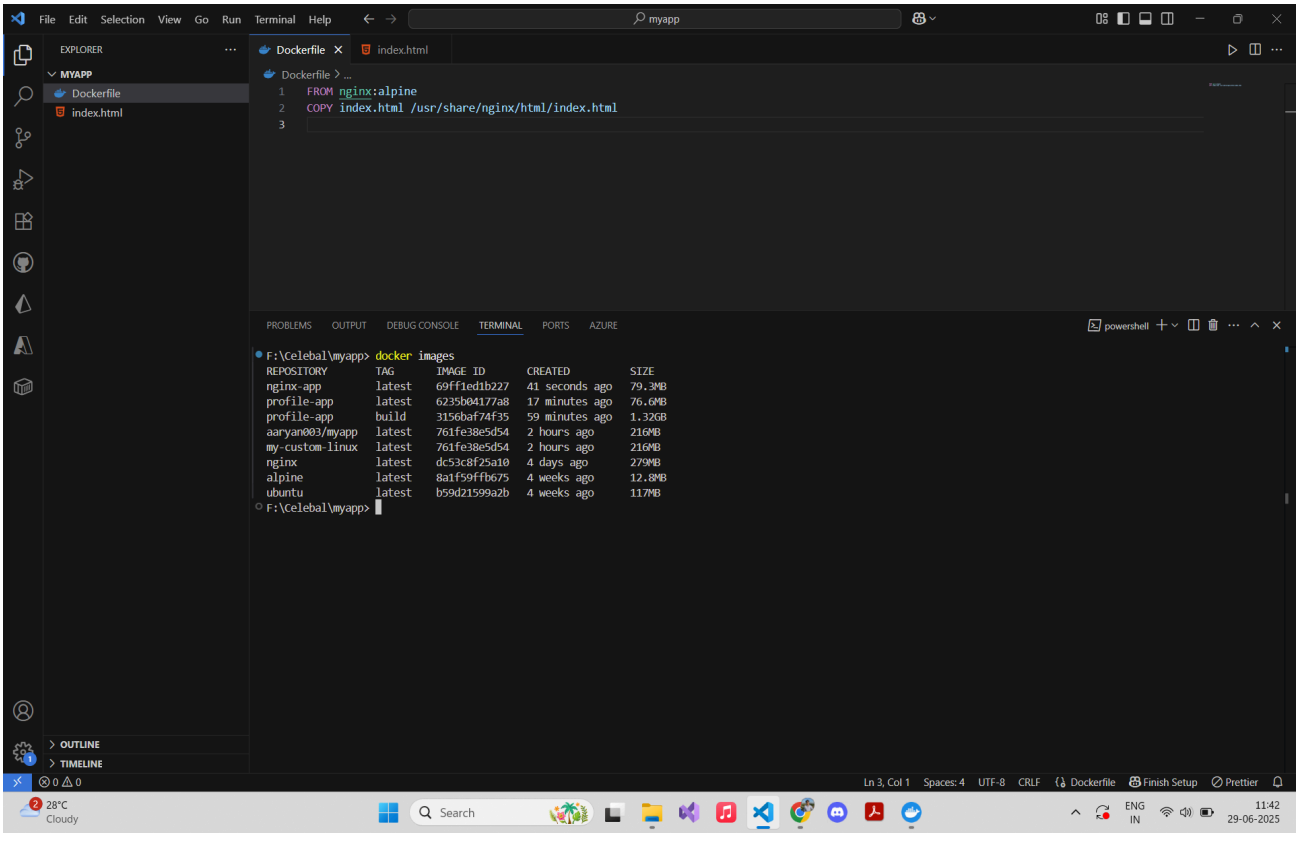

docker desktop PERSONAL
Search
Ctrl+K
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
Terminal
F:\Celebal\myapp> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d96ef88405b1 profile-app:latest "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8080->80/tcp zen_grothendieck
F:\Celebal\myapp> docker stop $(docker ps -q)
d96ef88405b1
F:\Celebal\myapp> docker rm $(docker ps -aq)
d96ef88405b1
9ebd9b94c944
1dd17e6fcafb
4a436440ad03
c5ad707758d1
53362ff36f7e
37a74ec2577b
F:\Celebal\myapp>
Engine running
RAM 4.28 GB CPU 0.45% Disk: 6.11 GB used (limit 1006.85 GB)
Terminal
New version available
Rain coming
11:50 am
Search
ENG
IN
11:31
29-06-2025

4. Create a docker image from multiple methods like Dockerfile, running containers.

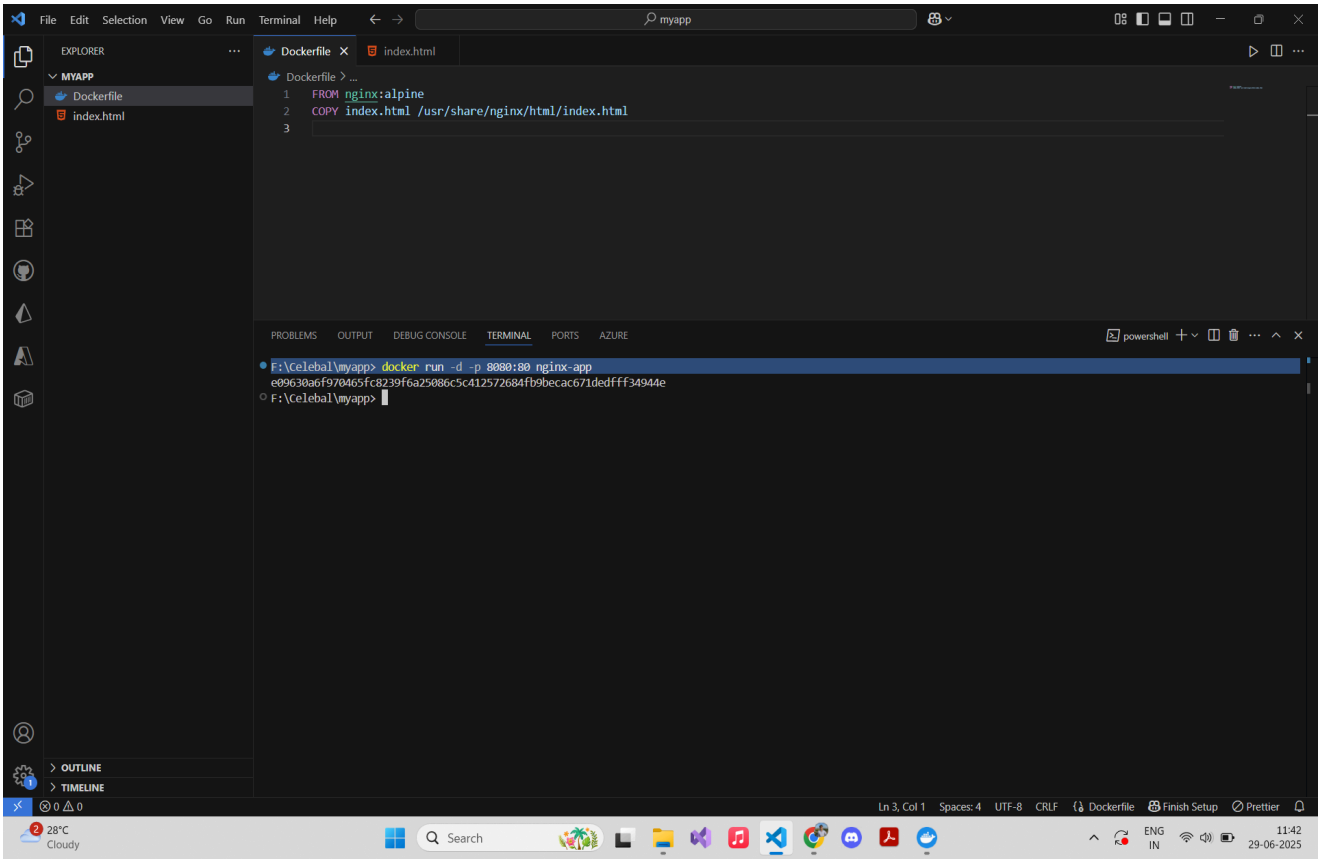
Method 1: Create Image from Dockerfile

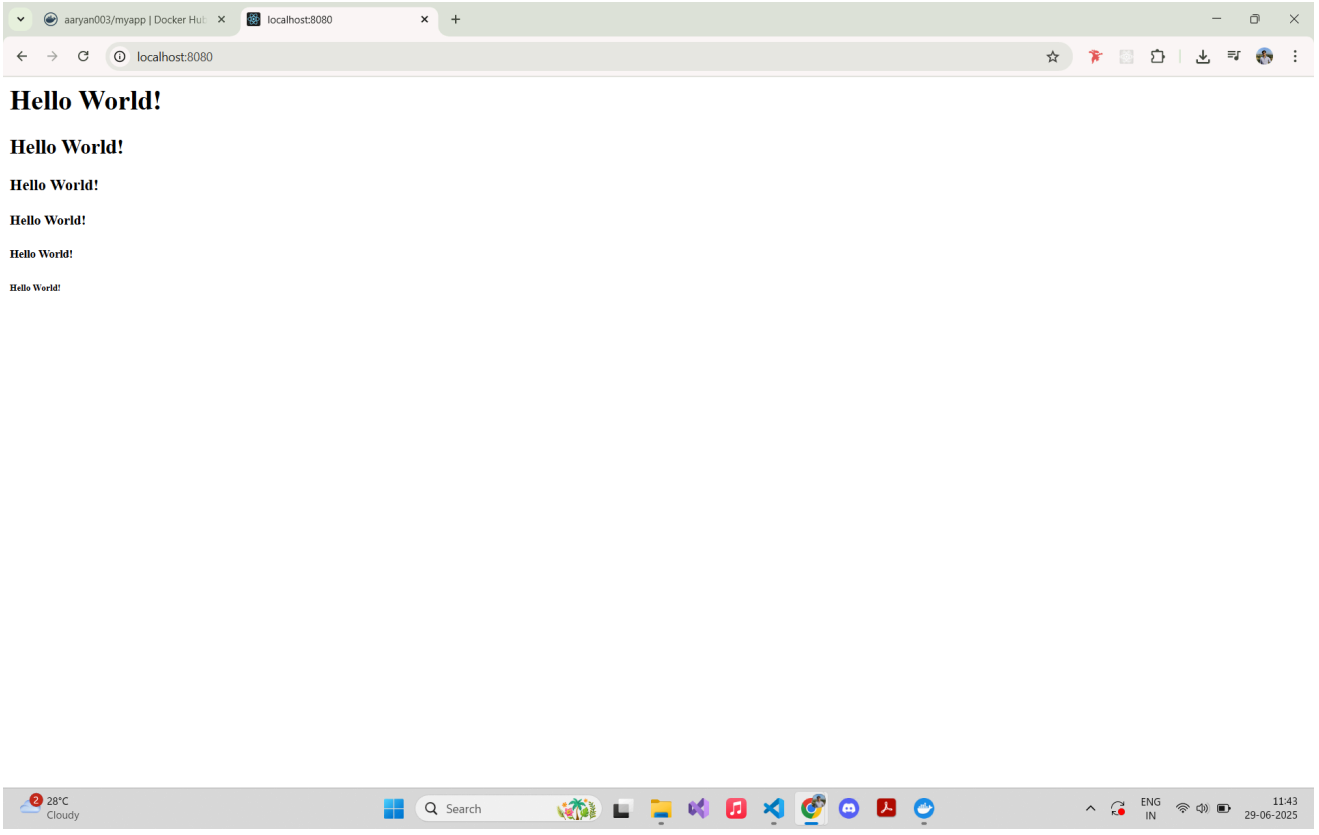


- Verify the Image Exists using ‘docker images’.



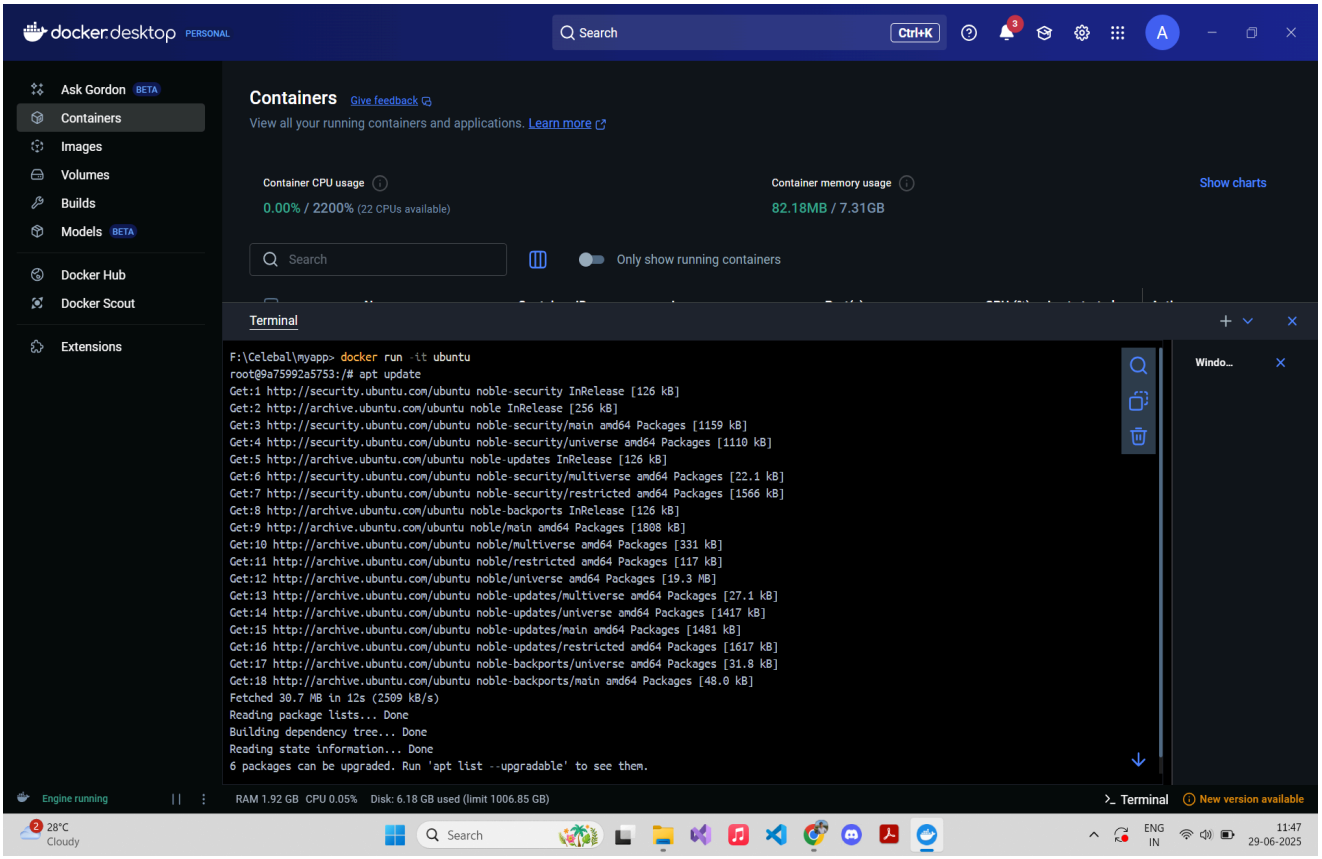
- Run and Test the Image.

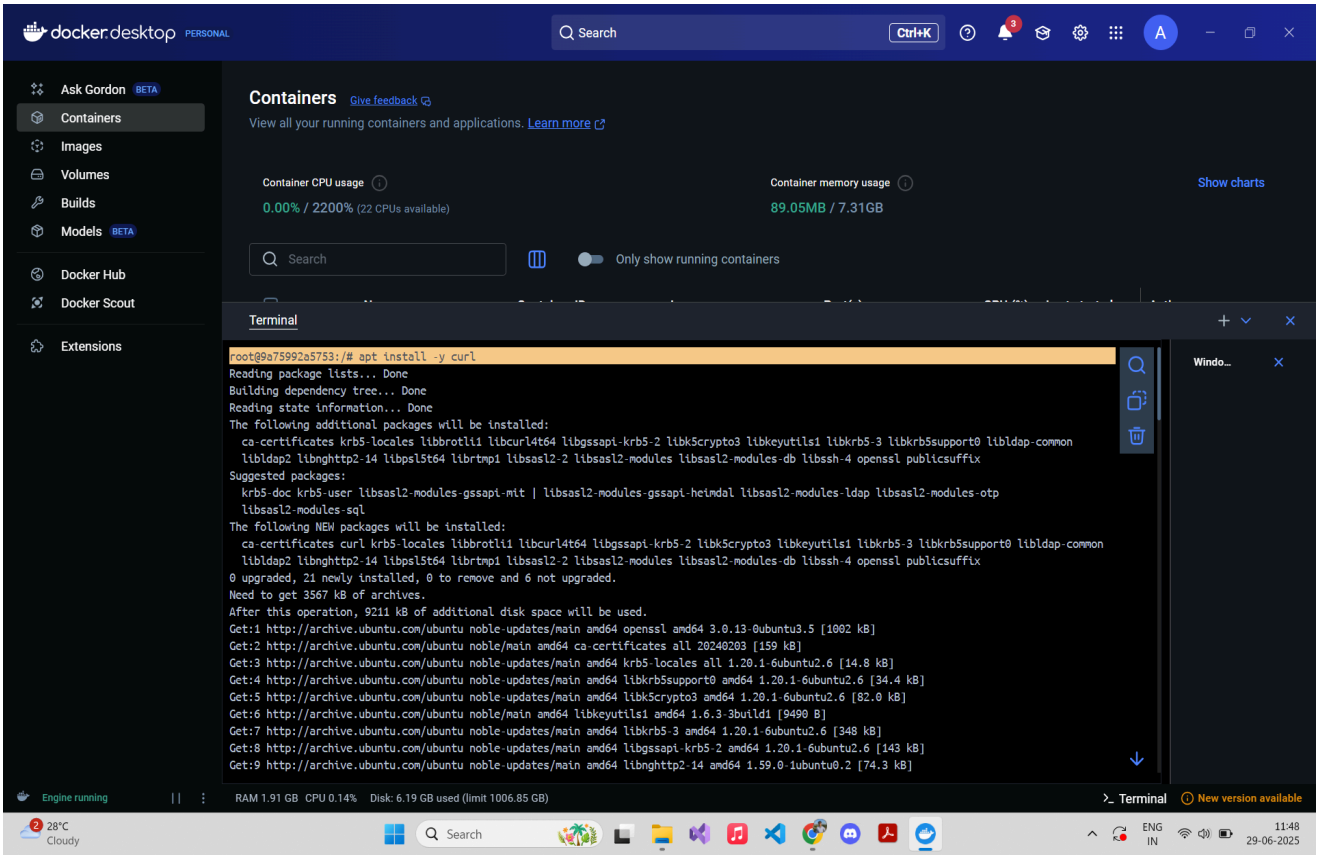




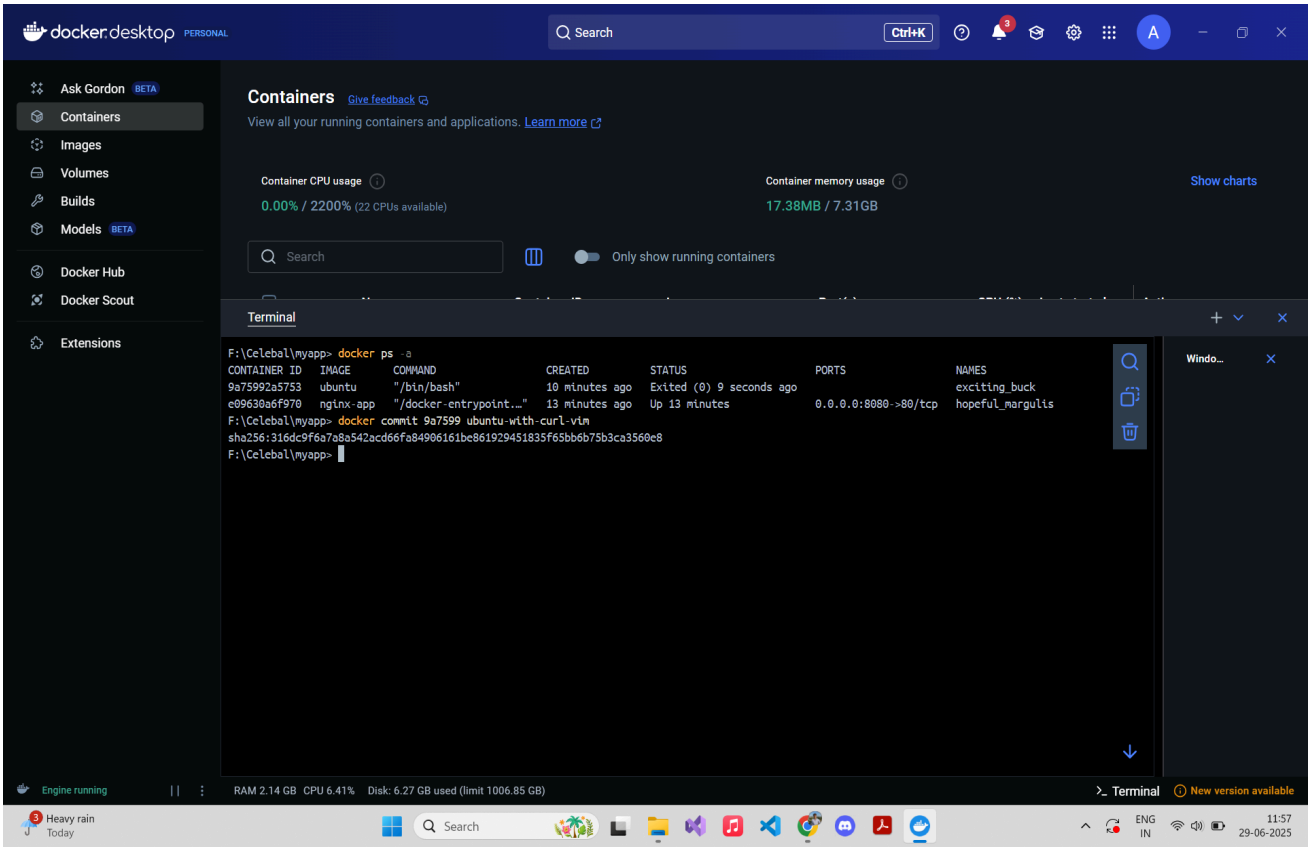
Method 2: Create Image from a Running Container

- Run a Base Container and Modify It.

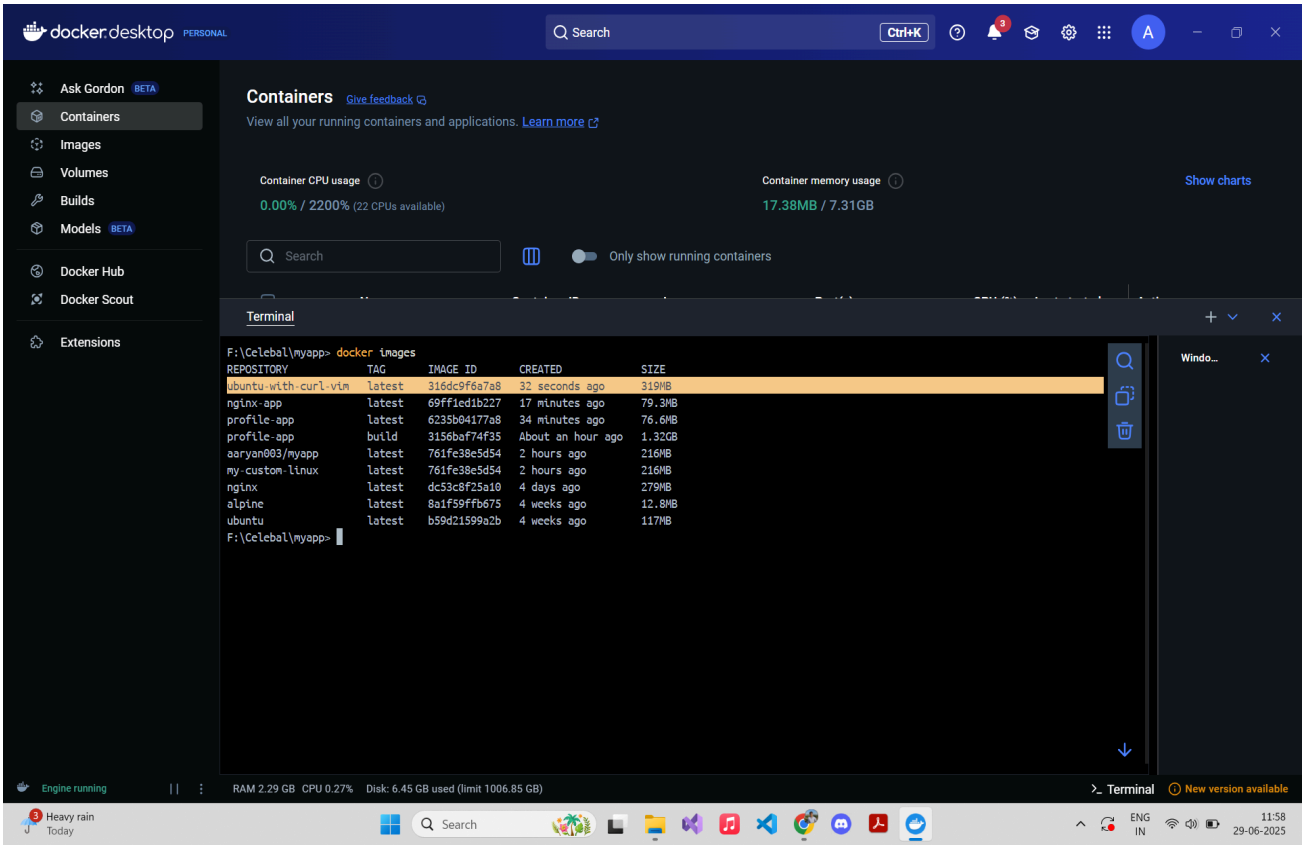




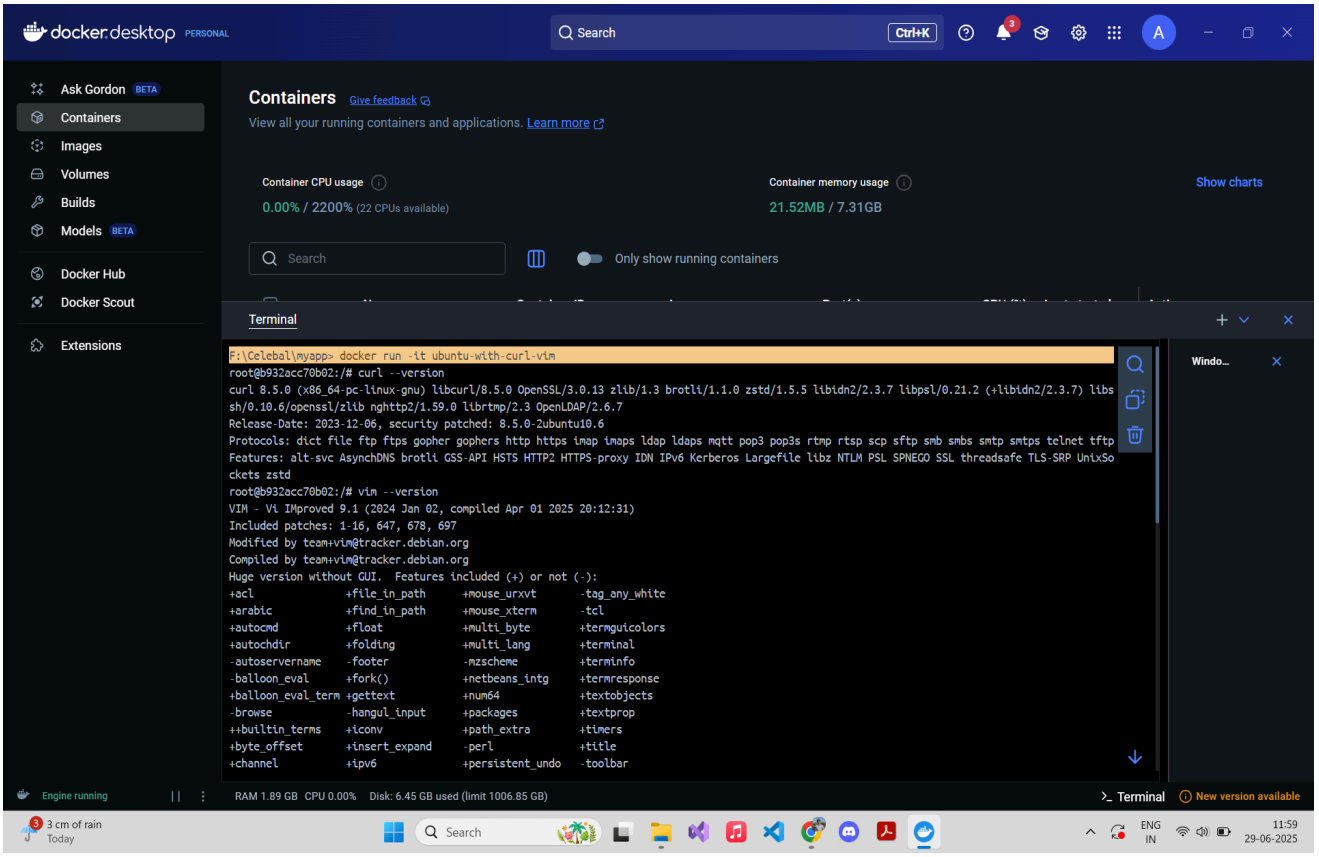
- Commit the Container to a New Image.

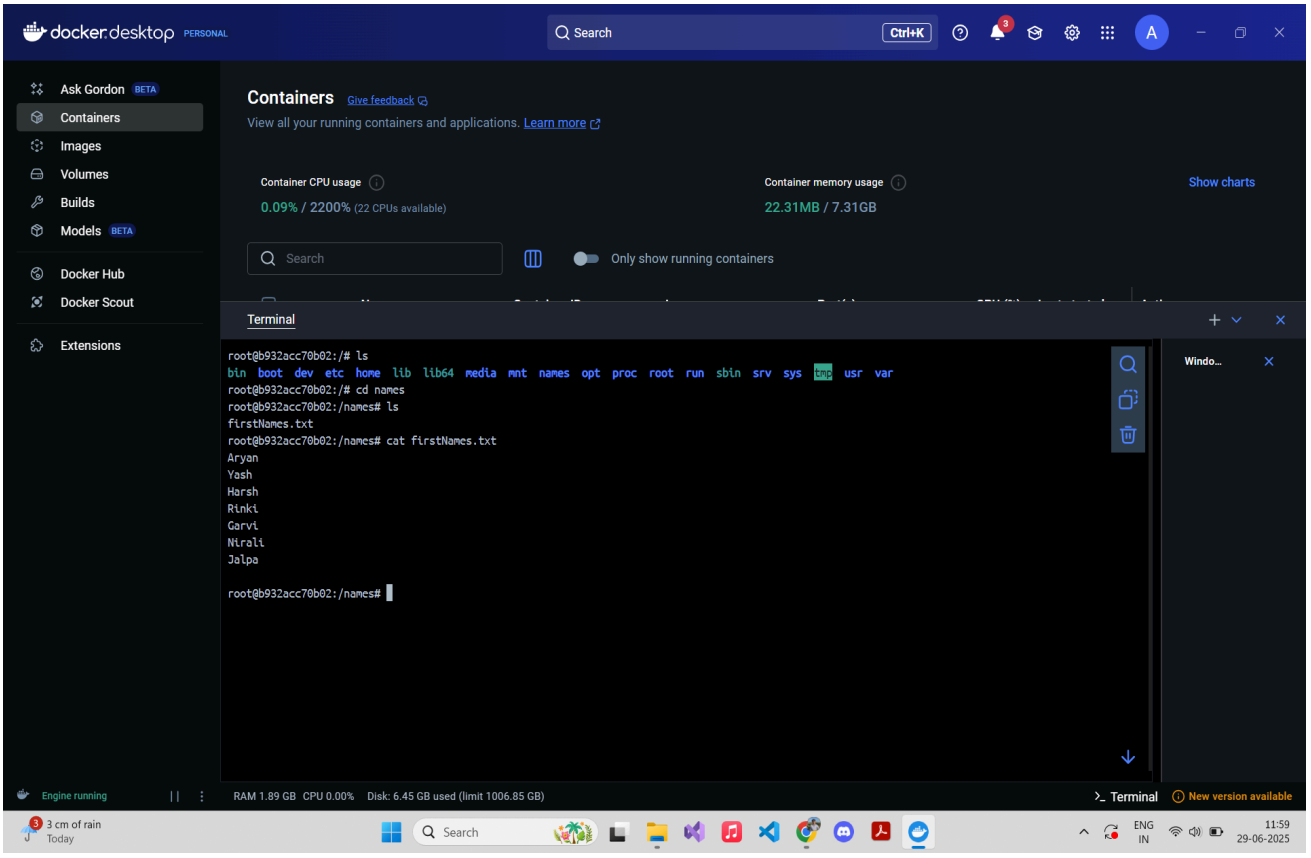


- Verify Image Was Created using “docker images”.



- Test New Image using “docker run” command.



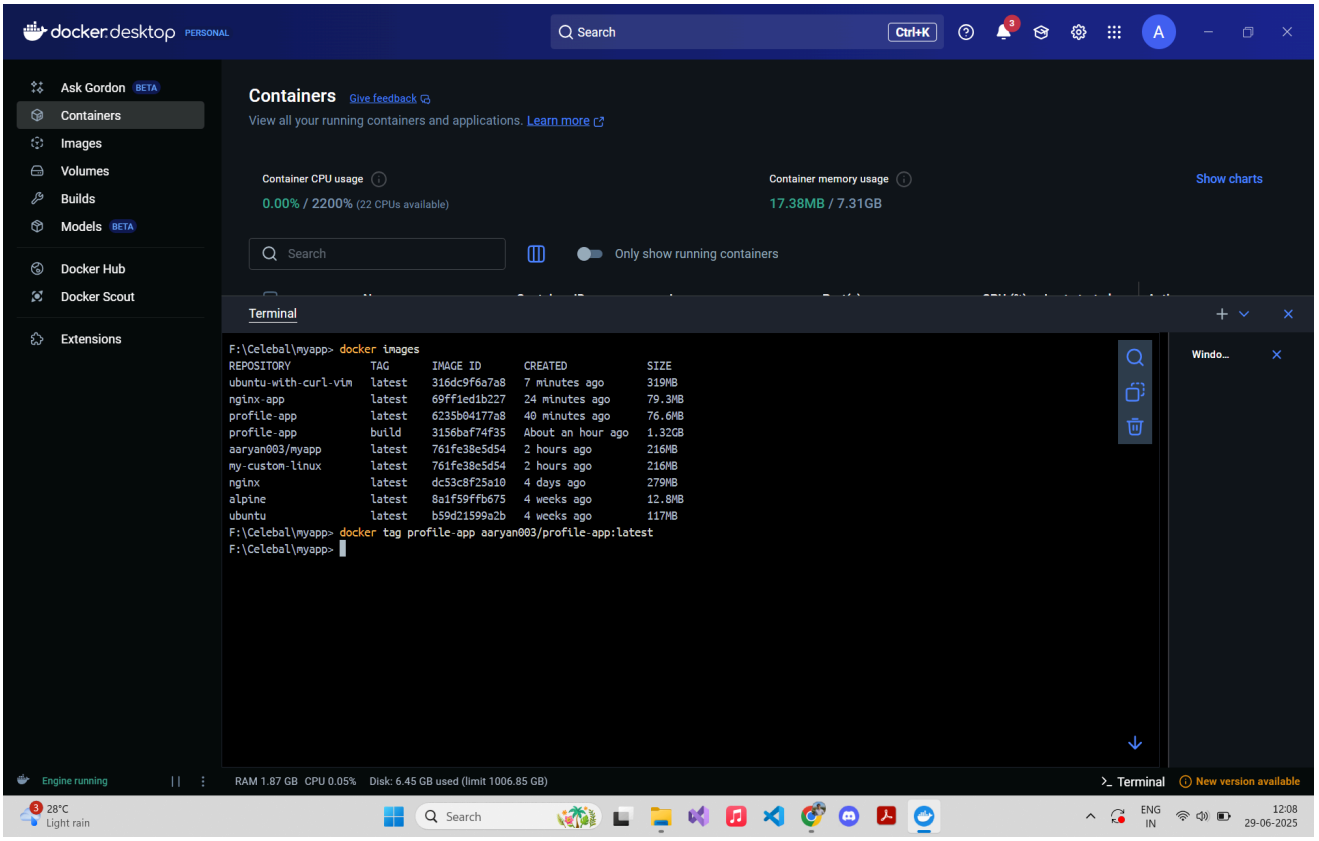


5. Push and pull image to Docker hub and ACR

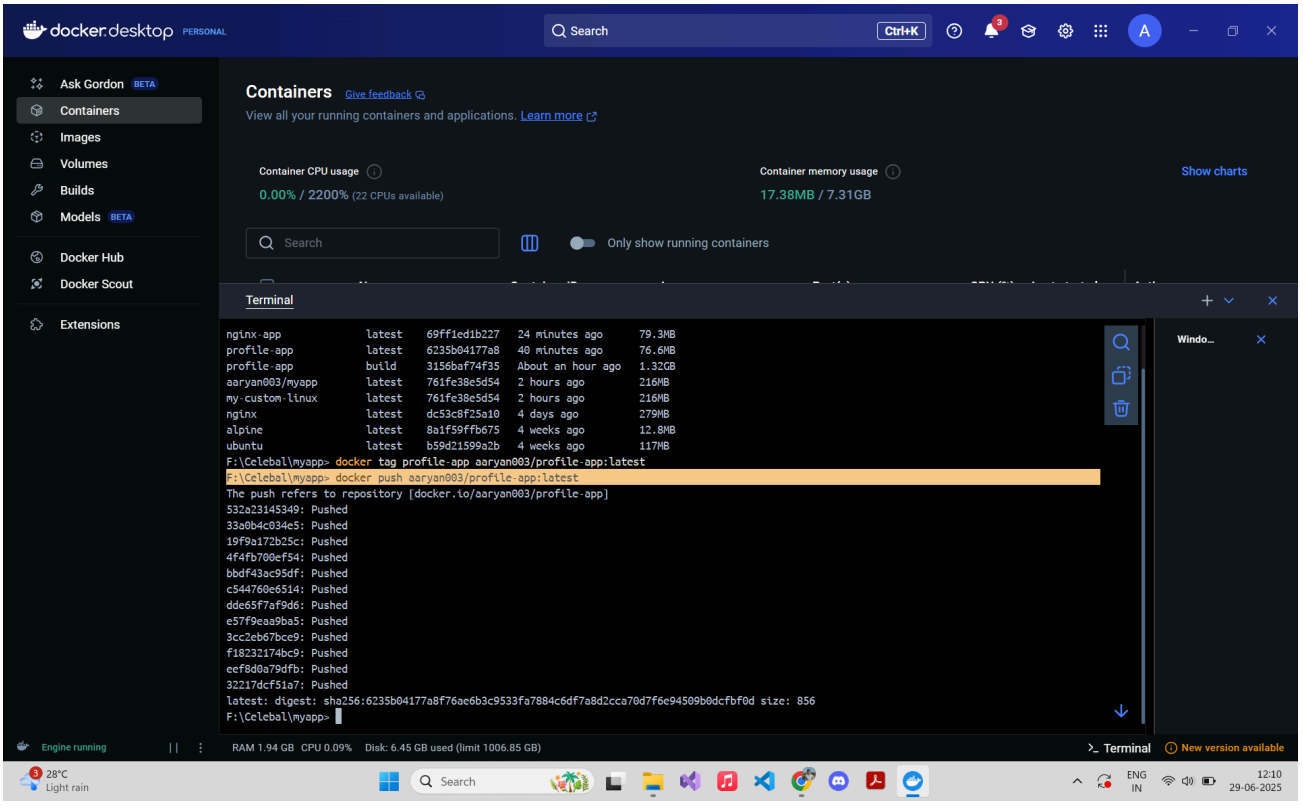
Steps :

1. Push & Pull Docker Image to/from Docker Hub

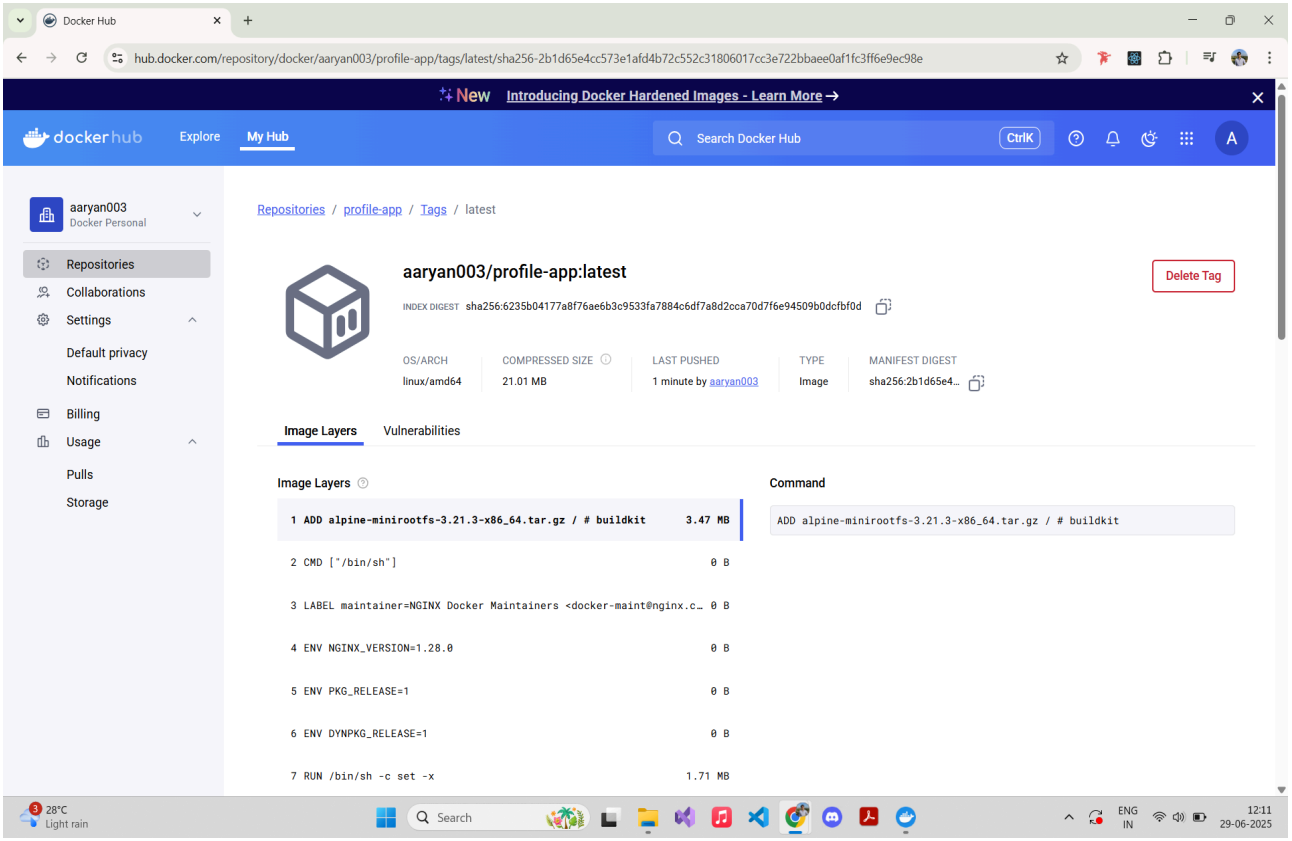
- Log in to Docker Hub using the "docker login" command and then tag any of your local images (eg : profile-app:latest).



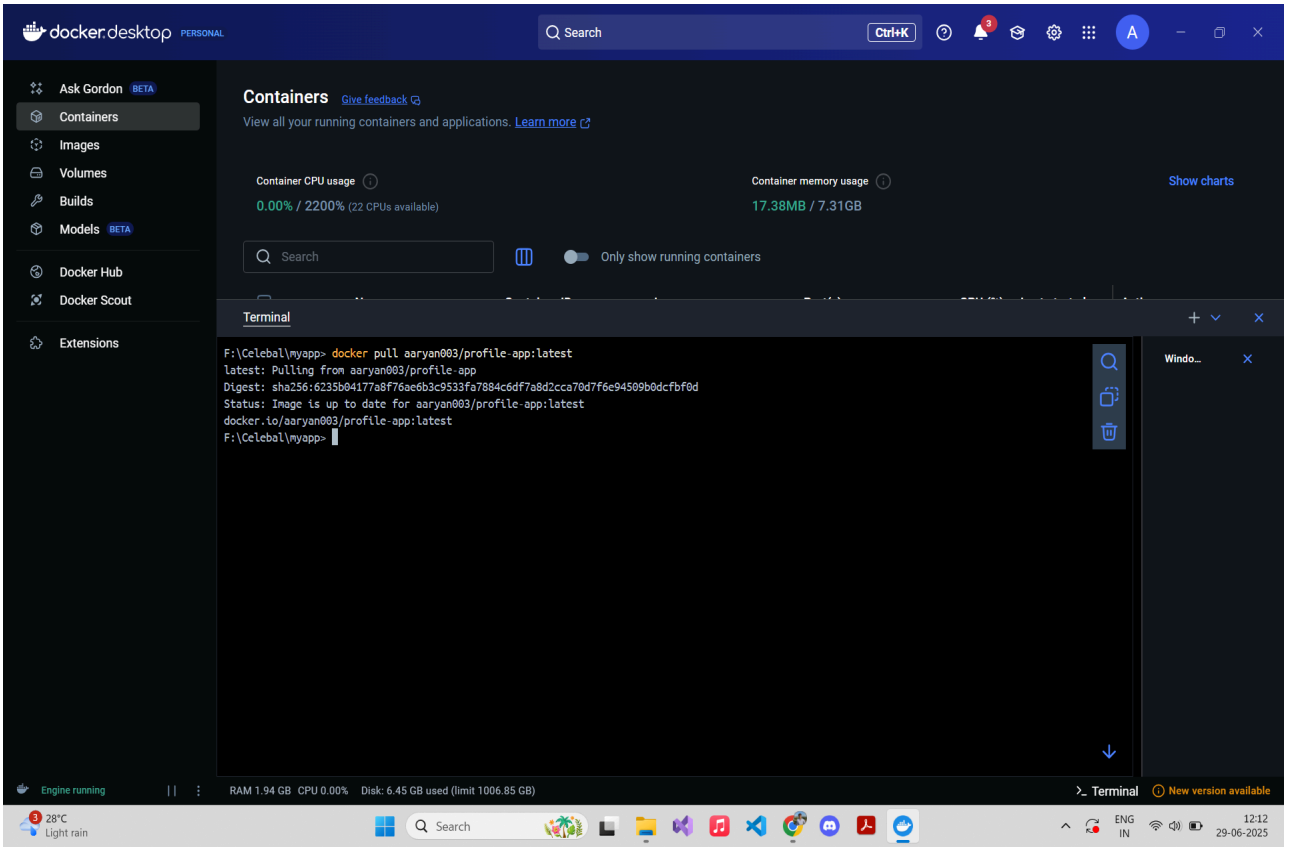
- Push Image to Docker Hub.



- Verify on Docker Hub Website.

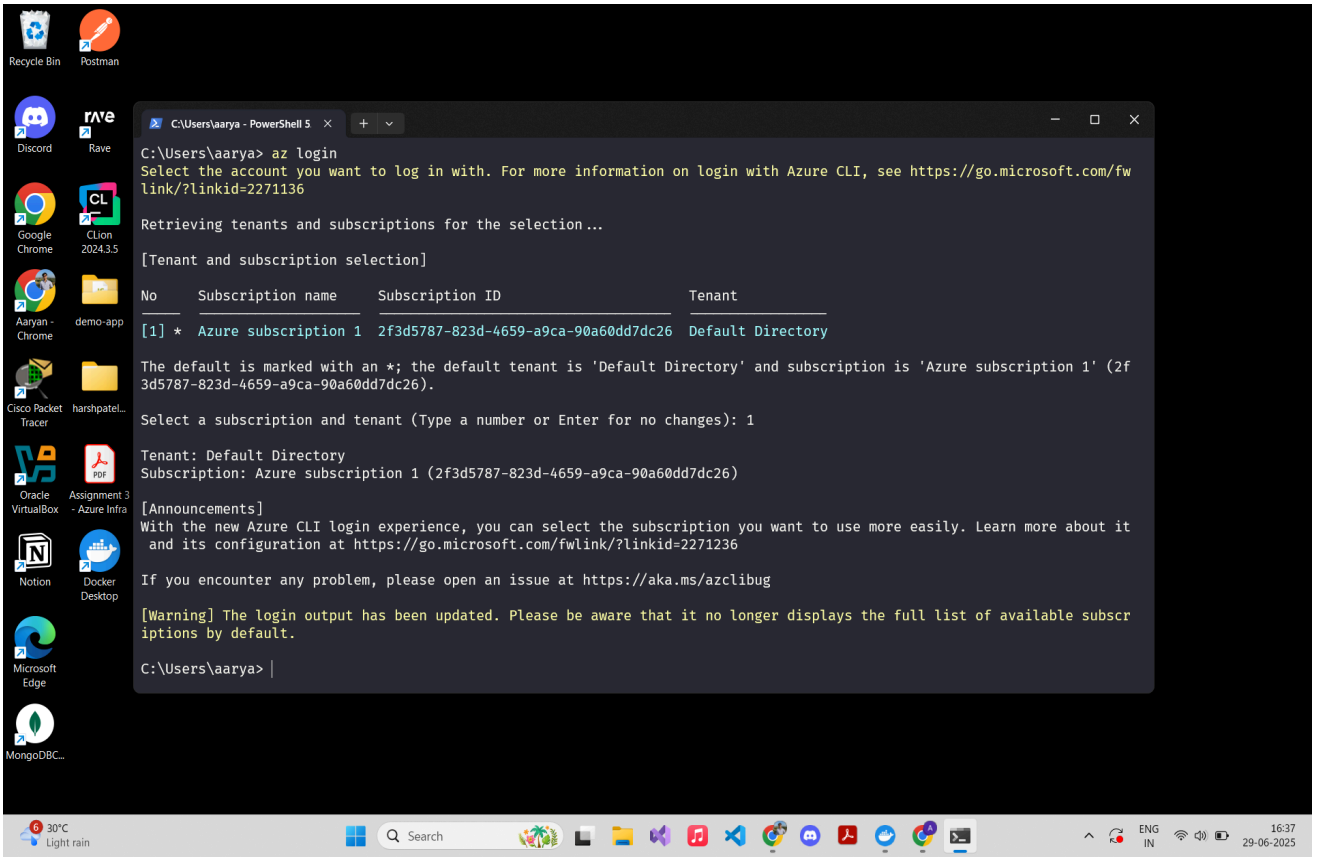


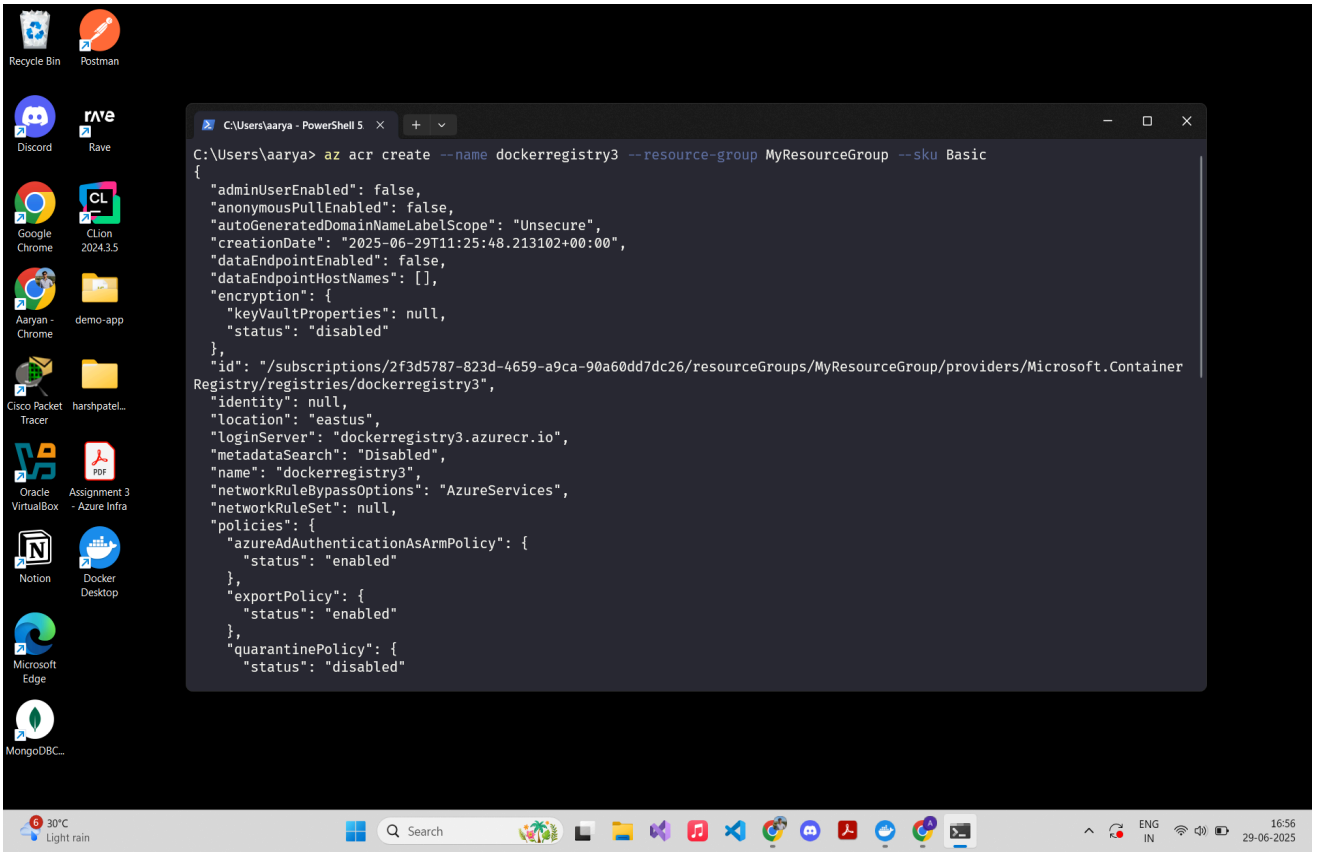
- Pull from Docker Hub on Any Machine.



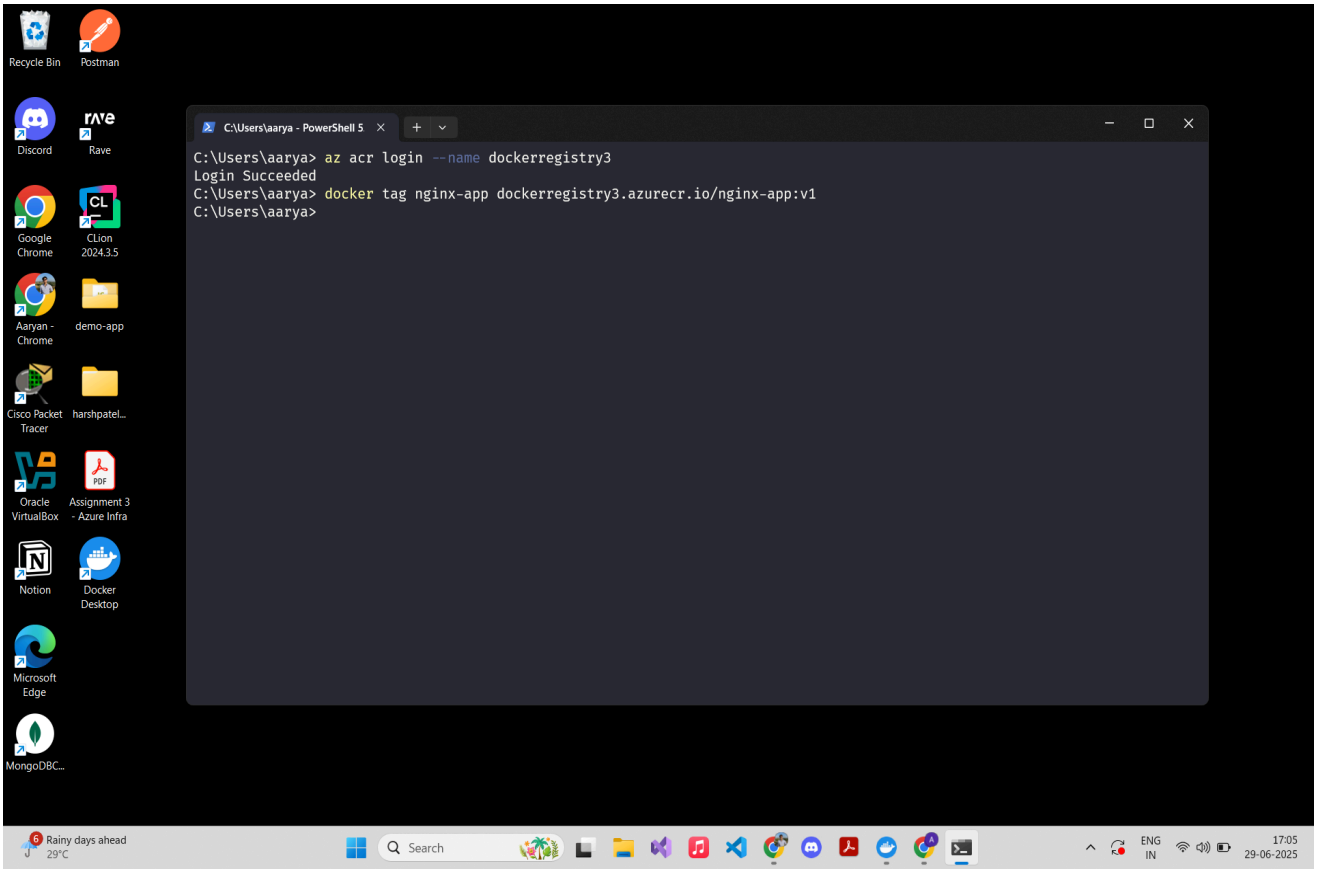
2. Push & Pull Docker Image to/from Azure Container Registry (ACR).

- Create Azure Container Registry.

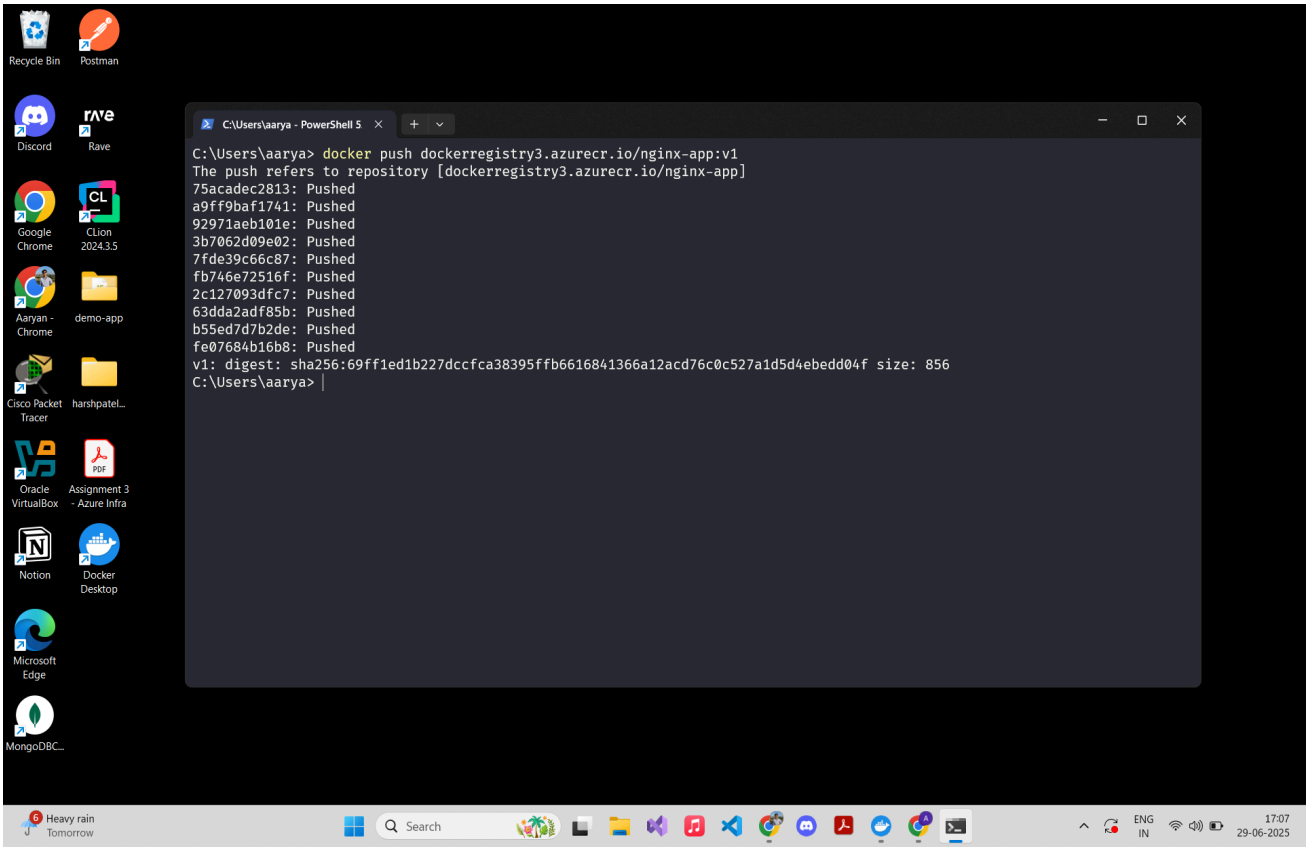




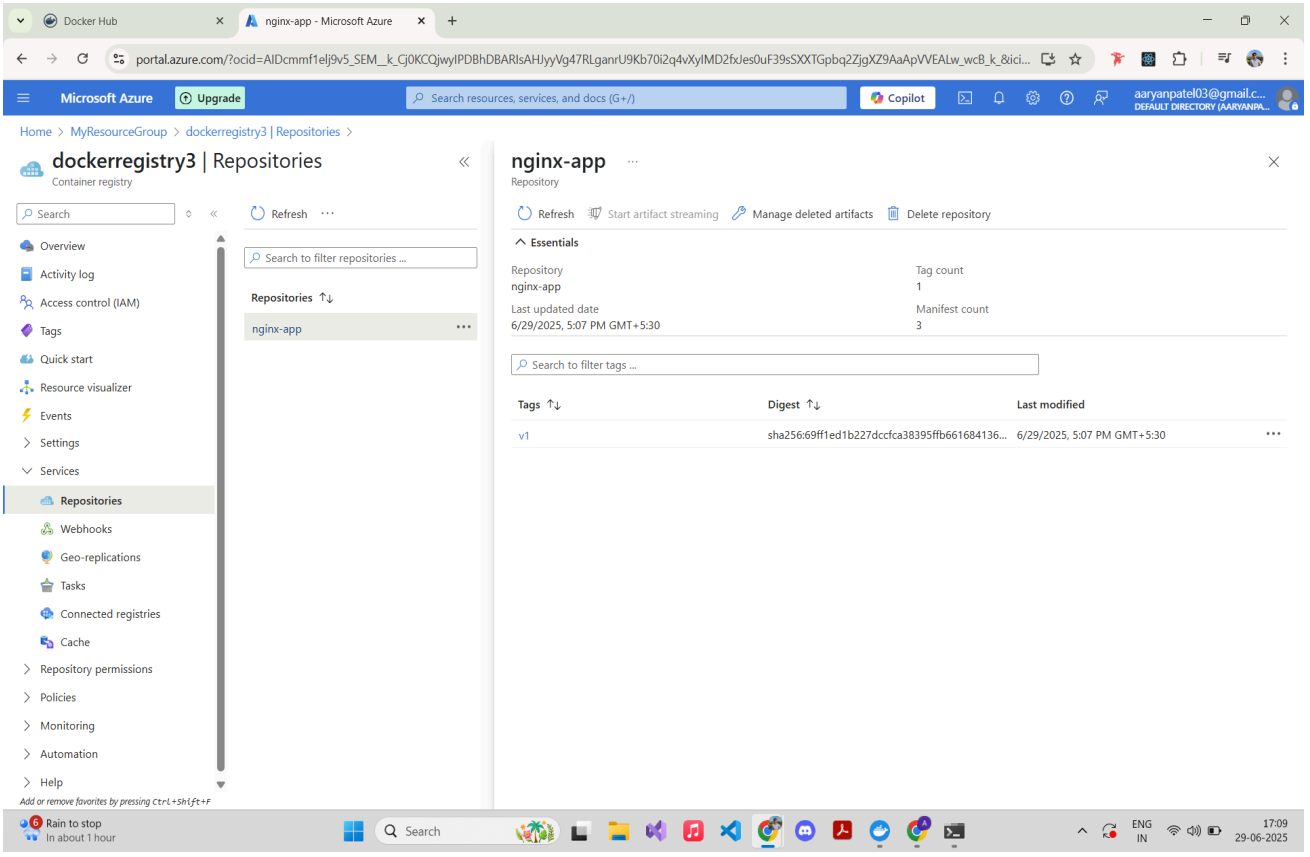
- Log in to ACR
- Tag Image with ACR Format



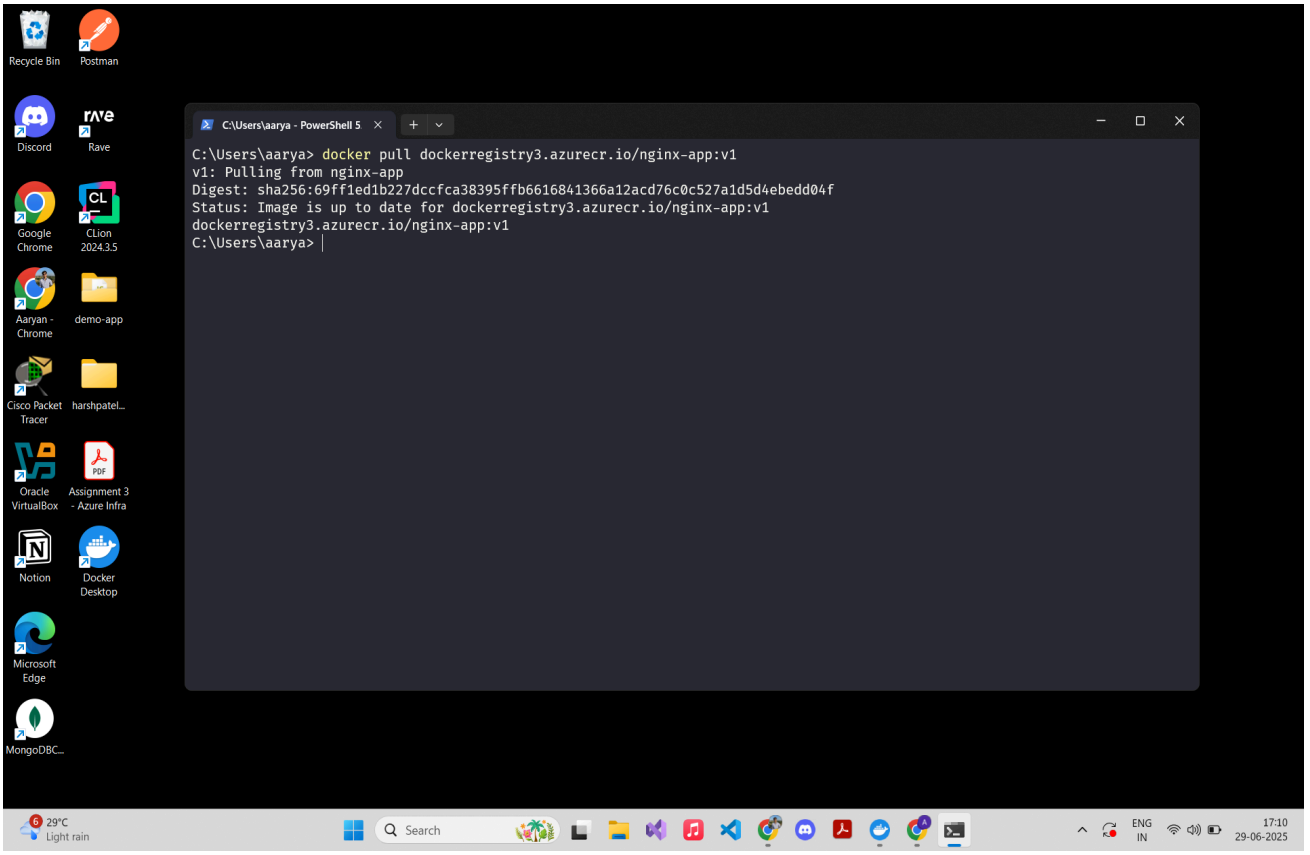
- Push to ACR



- Verify in Azure Portal



- Pull from ACR on Any Machine

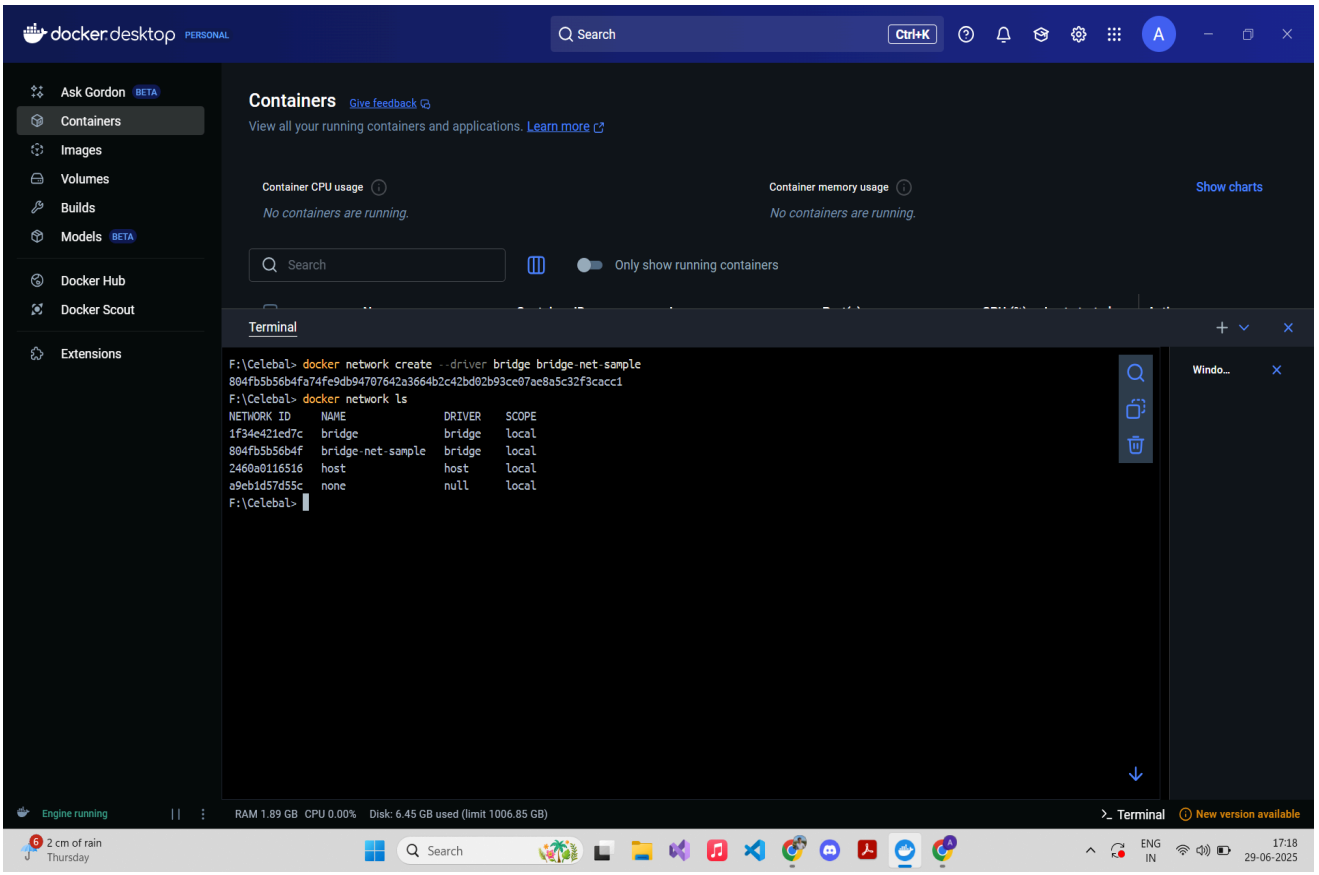


6. Create a Custom Docker Bridge Network

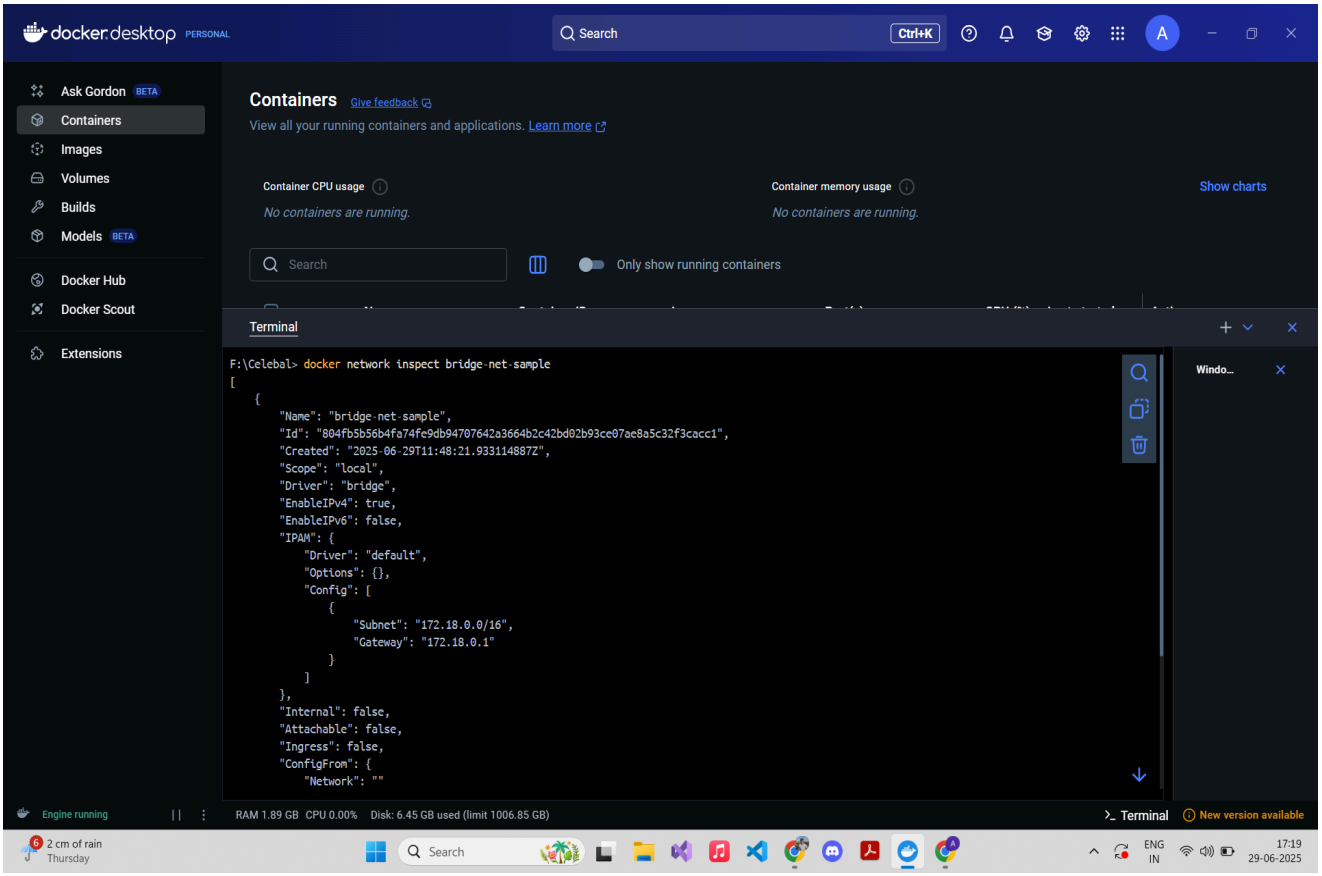
Steps :

1. Create and Use a Custom Docker Bridge Network.

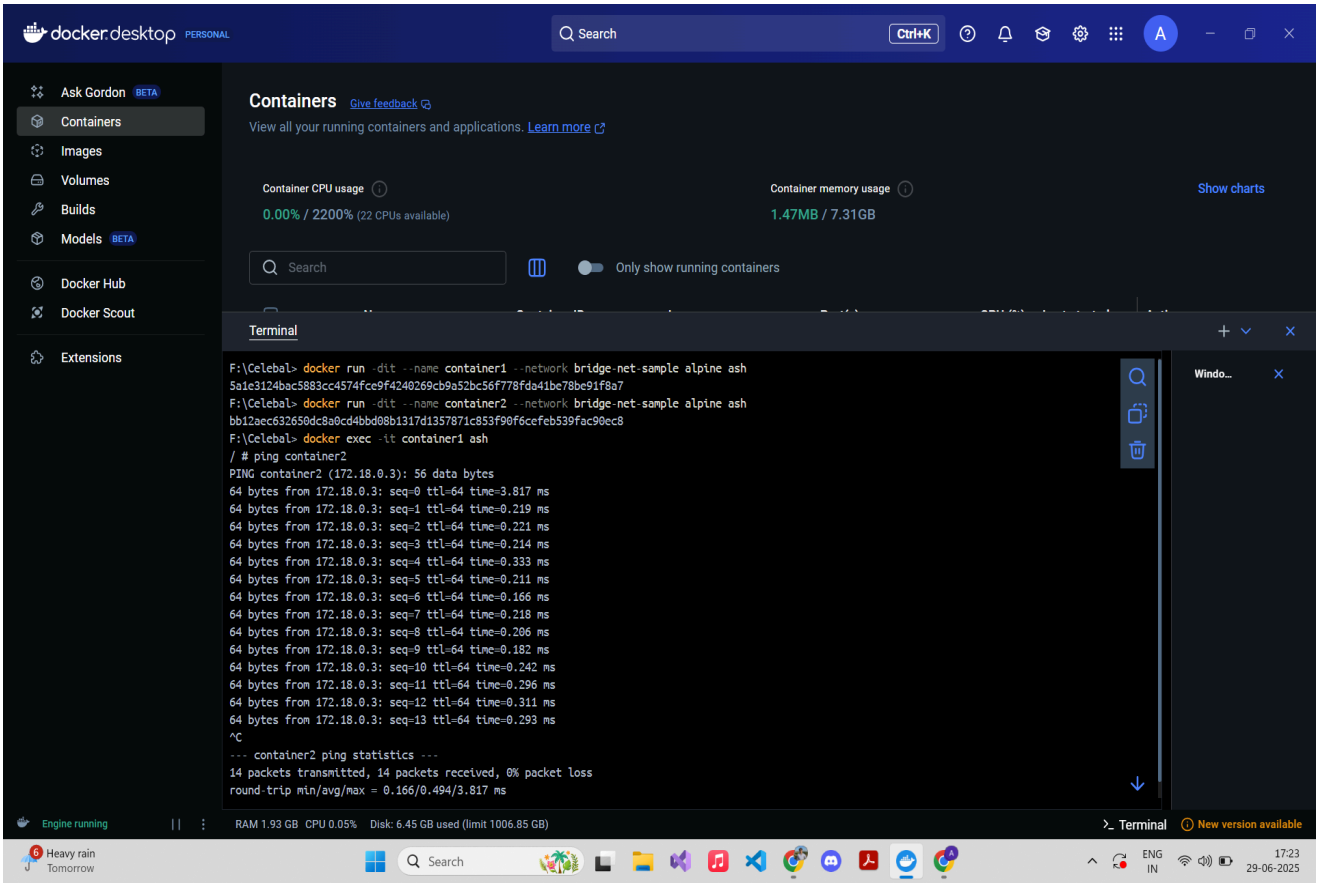
- Create a Custom Bridge Network.
- Verify the Network Exists.

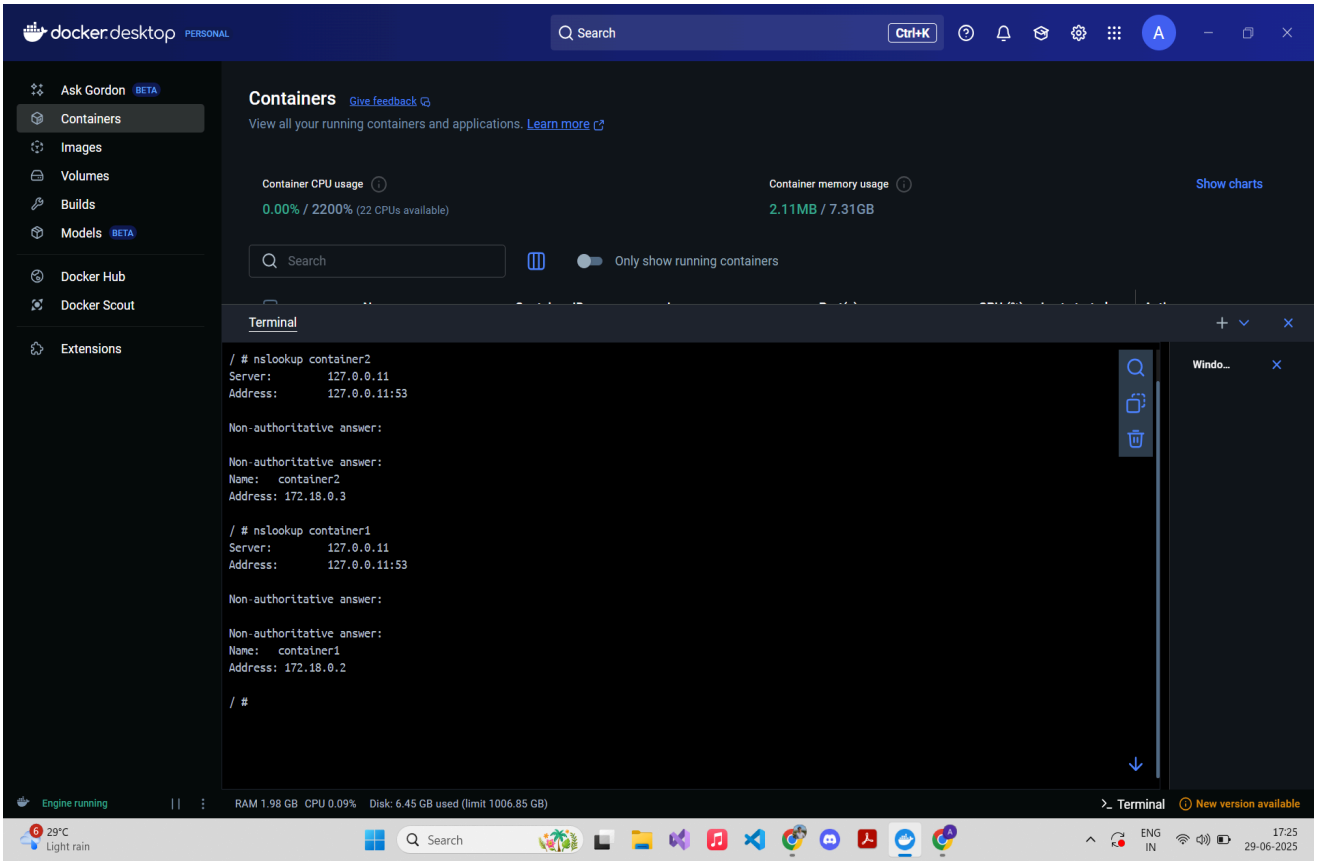


- Inspect the Network



- Run Two Containers in the Same Network.
- Test Inter-Container Communication.



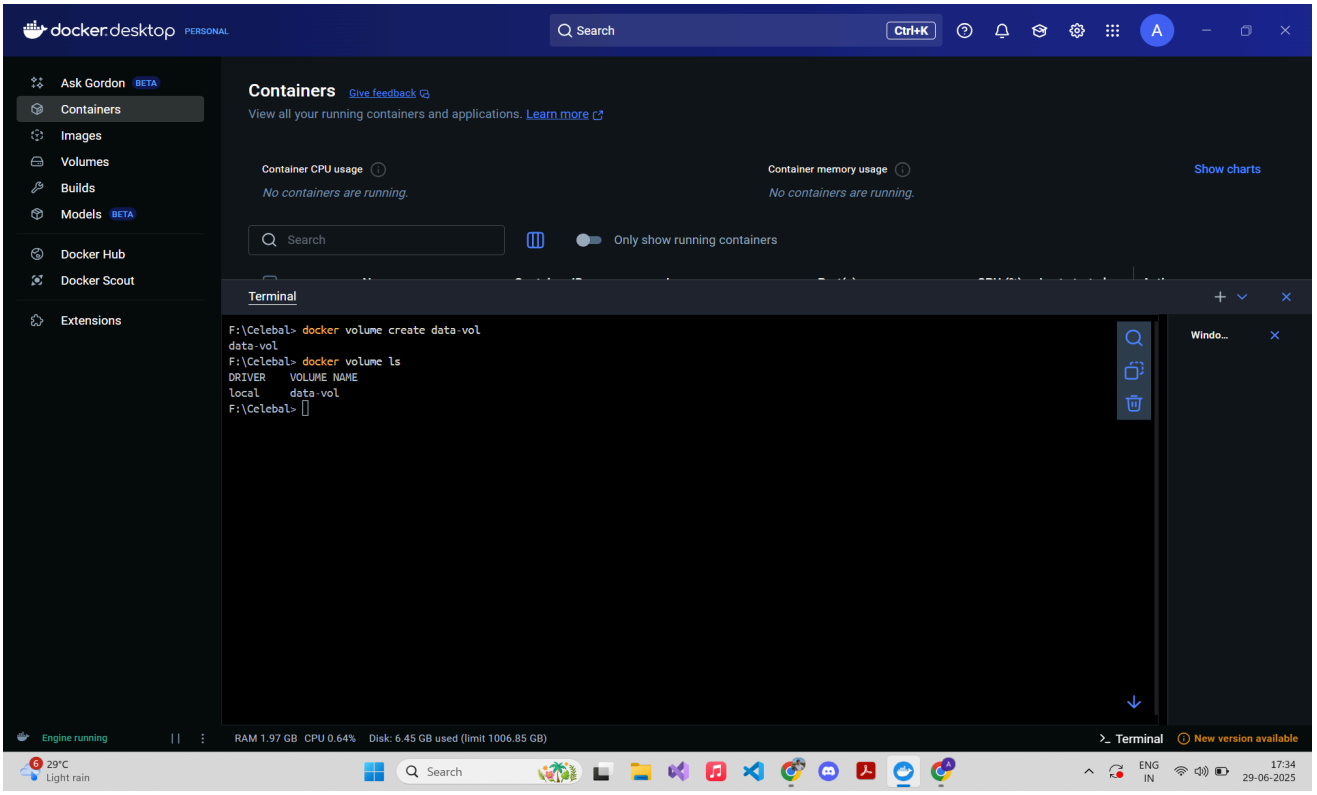

docker desktop PERSONAL
Search Ctrl+K
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
Container CPU usage
0.00% / 2200% (22 CPUs available)
Container memory usage
2.11MB / 7.31GB
Show charts
Search
Only show running containers
Terminal
/ # nslookup container2
Server: 127.0.0.11
Address: 127.0.0.11:53
Non-authoritative answer:
Non-authoritative answer:
Name: container2
Address: 172.18.0.3
/ # nslookup container1
Server: 127.0.0.11
Address: 127.0.0.11:53
Non-authoritative answer:
Non-authoritative answer:
Name: container1
Address: 172.18.0.2
/ #
Windo...
Engine running
RAM 1.98 GB CPU 0.09% Disk: 6.45 GB used (limit 1006.85 GB)
Terminal
New version available
29°C
Light rain
Search
ENG
IN
17:25
29-06-2025

7. Create a Docker volume and mount it to a container.

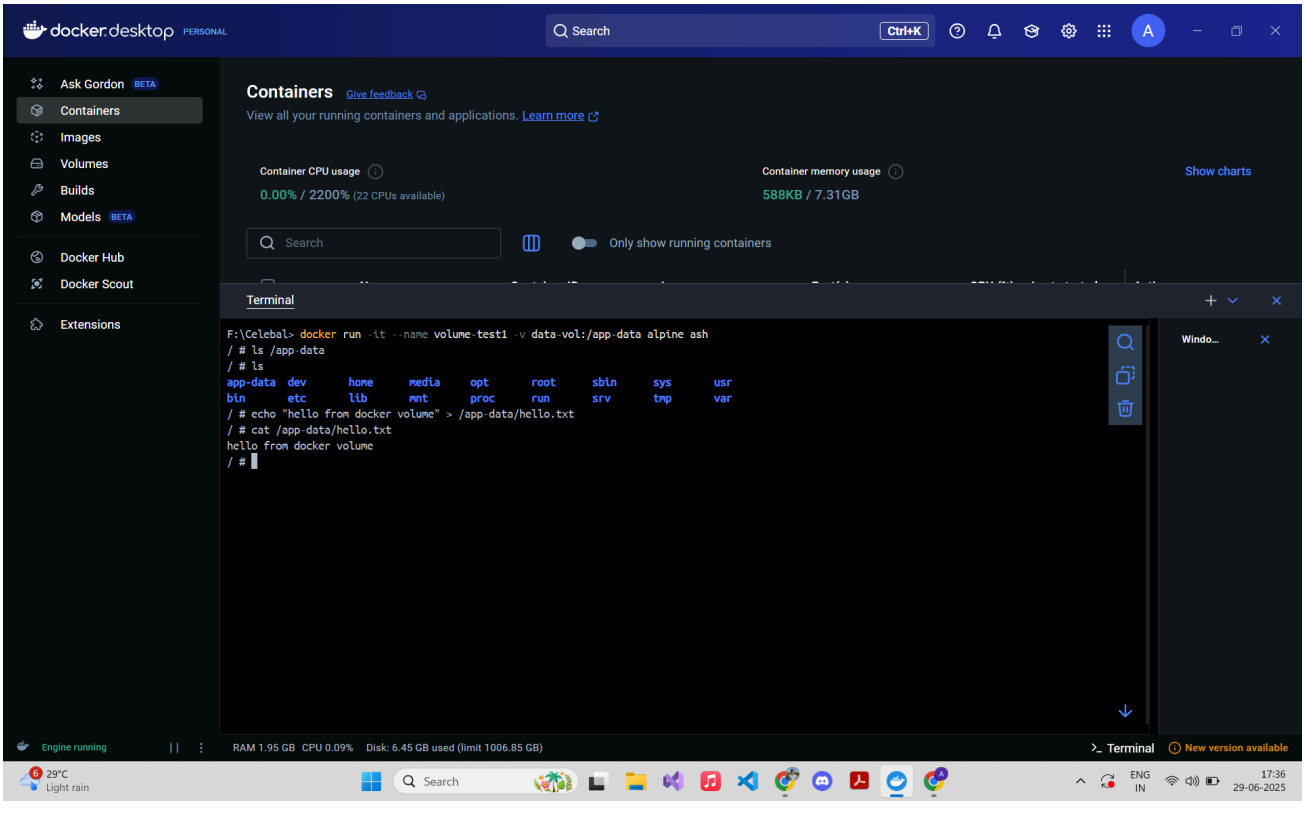
Steps :

1. Create and Use a Docker Volume.

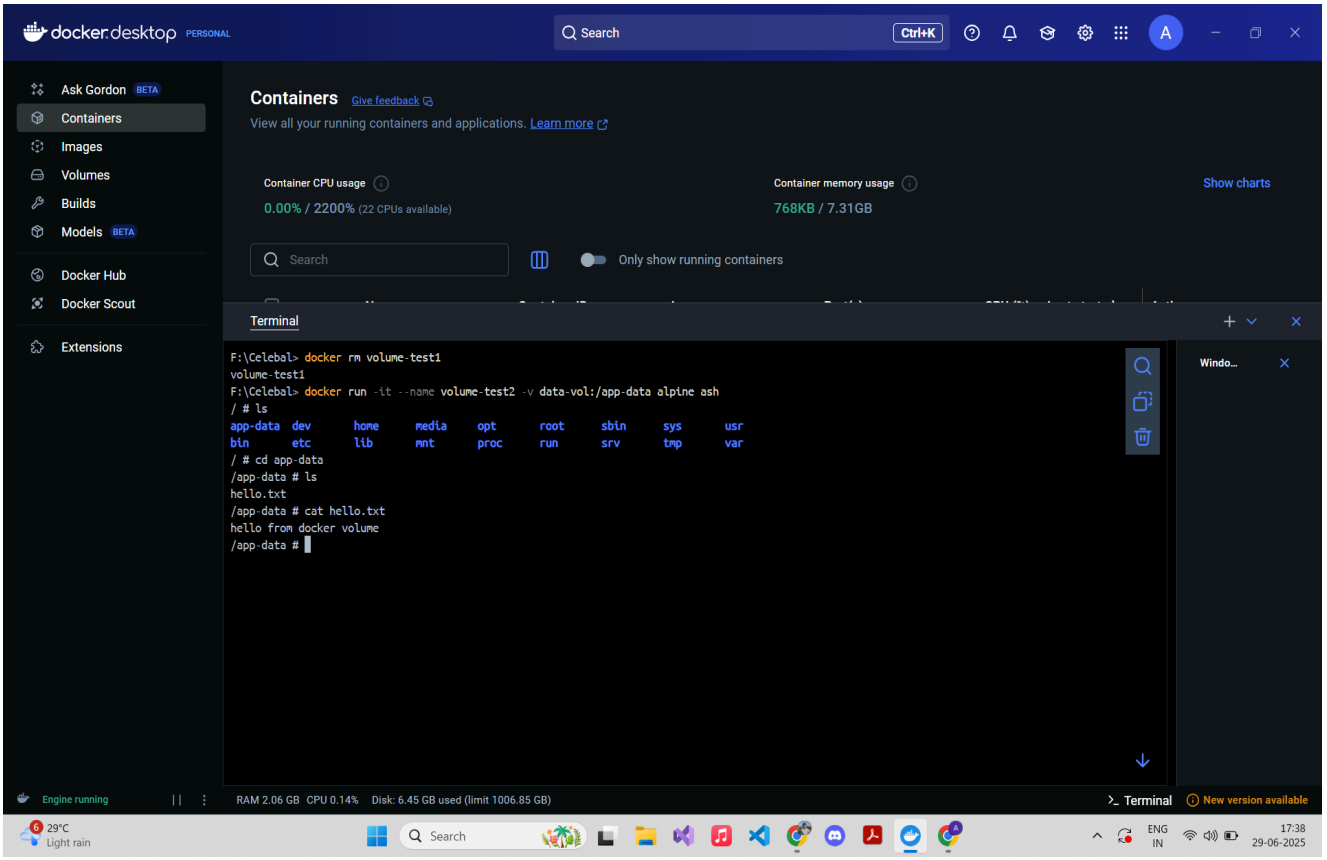
- Create a Docker Volume.
- Verify Volume Creation.



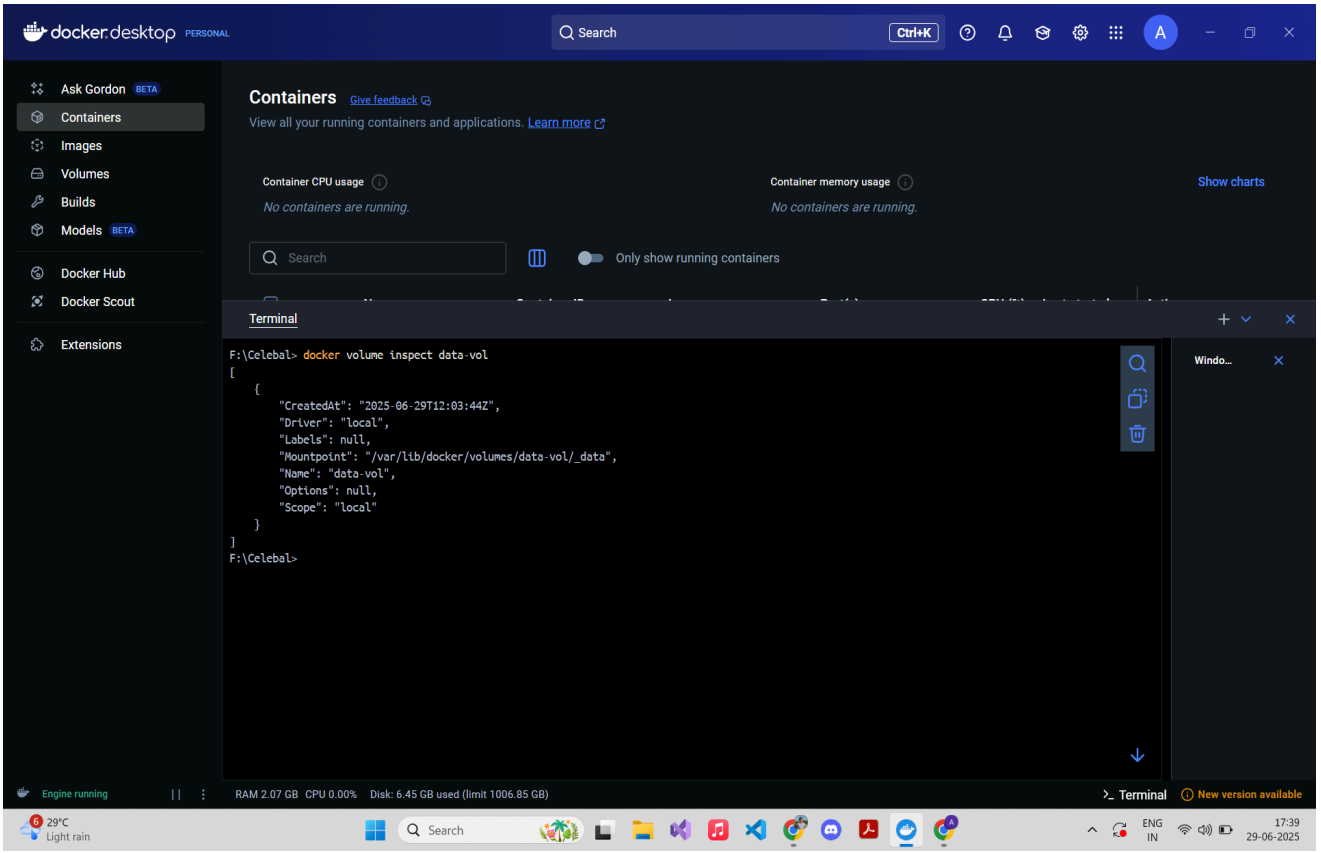
- Run a Container and Mount the Volume.
- Write Some Data into the Volume.



- Remove the Container.
- Run a New Container with the Same Volume.



- Inspect Volume Path on Host.

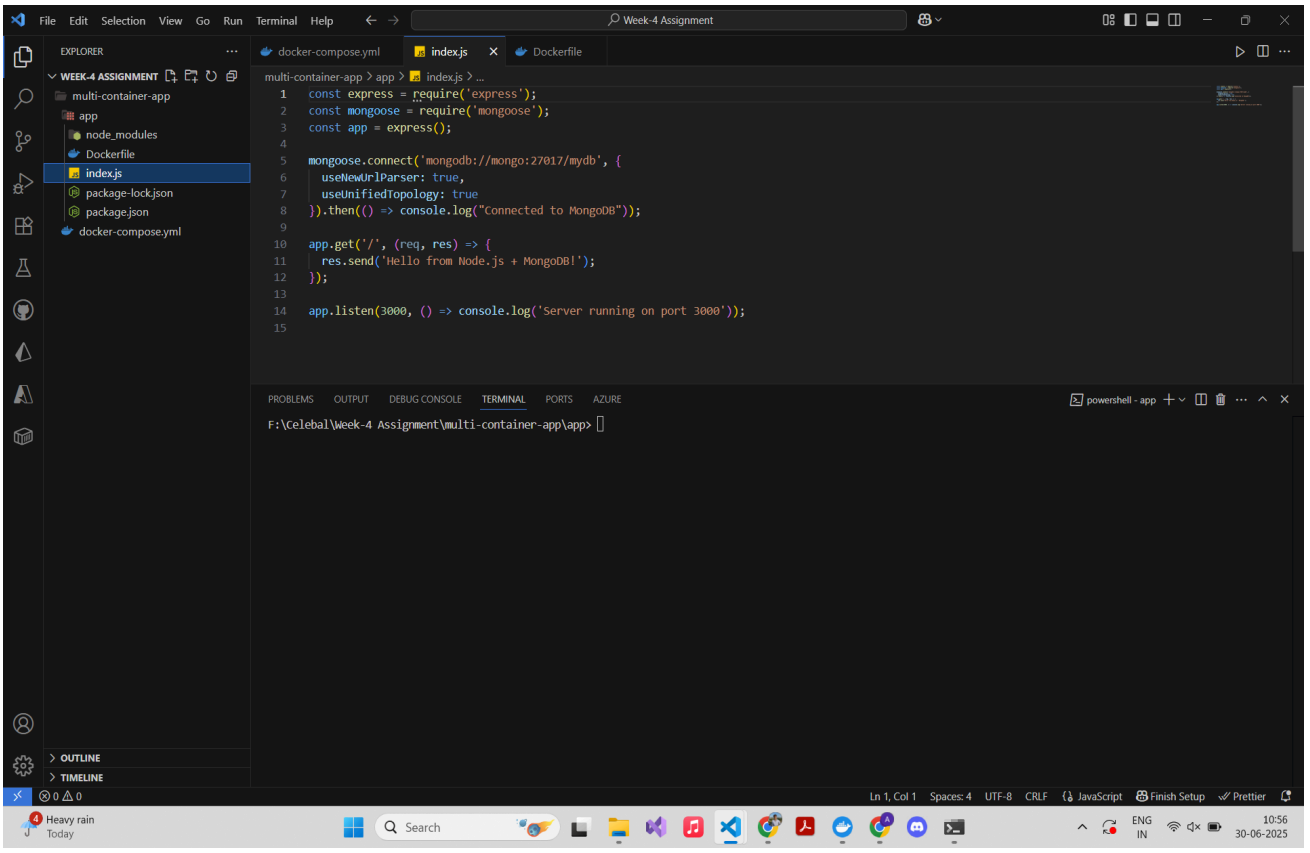


8. Docker Compose for multi-container applications, Docker security best practices

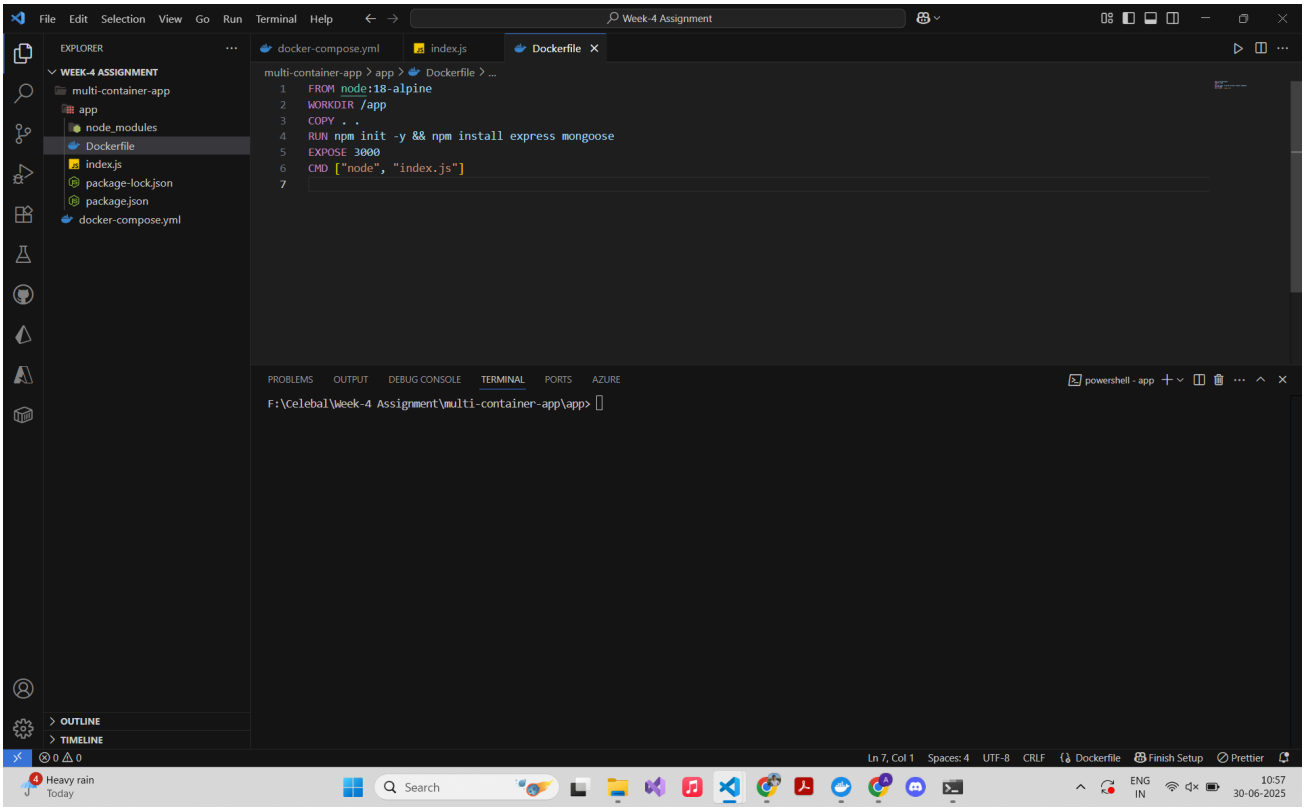
Steps :

Part 1: Docker Compose – Multi-Container Applications

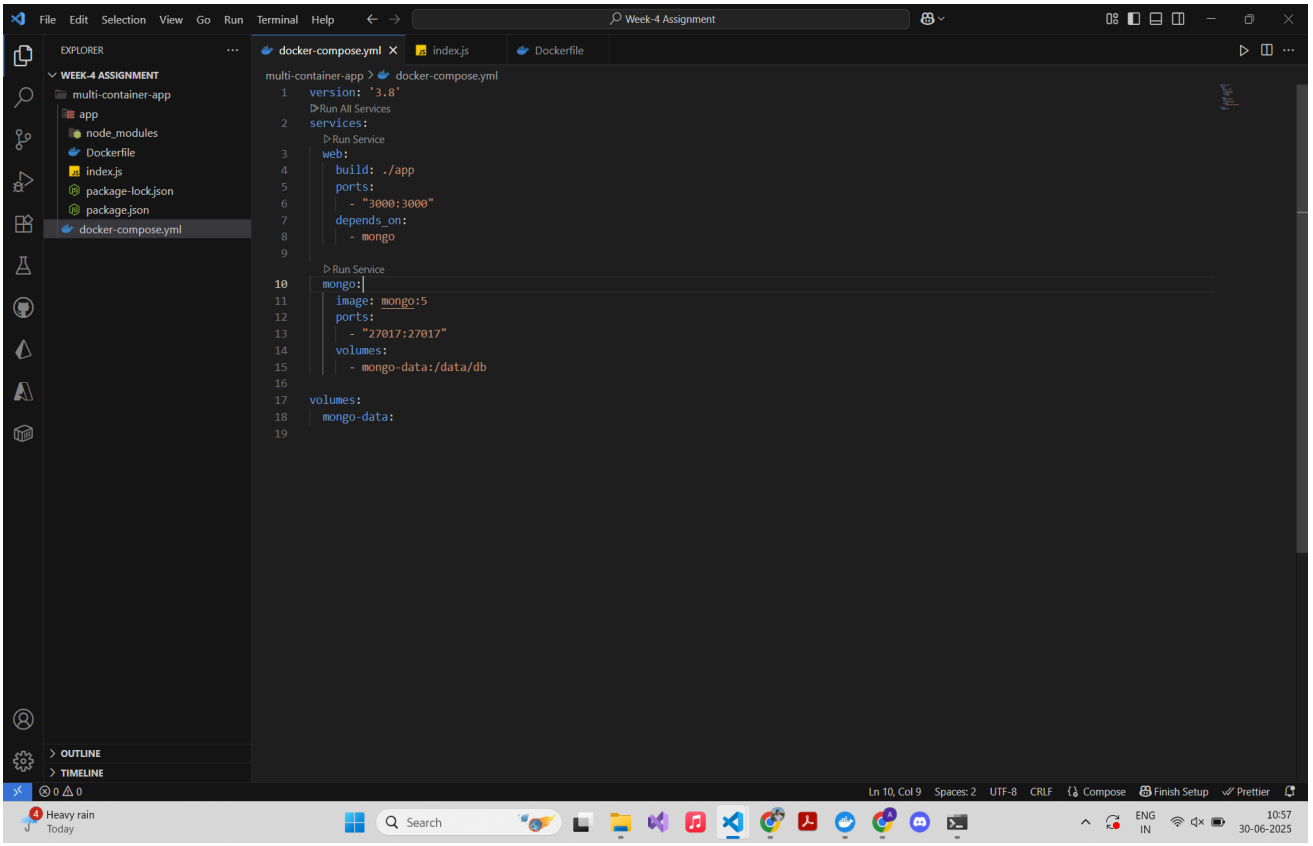
- Create a Simple `Node.js + MongoDB` App.



- Create Dockerfile for Node.js

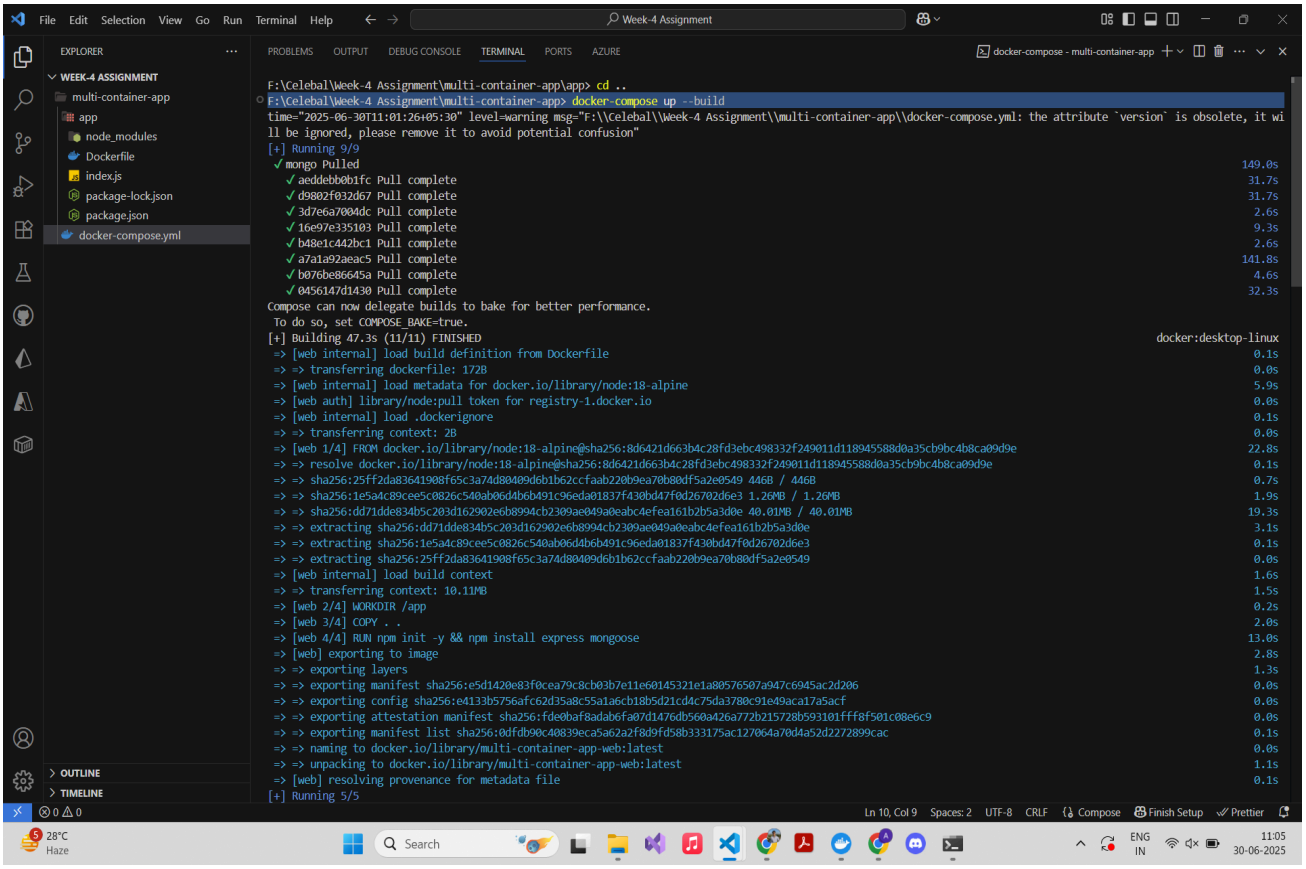


- Create docker-compose.yml



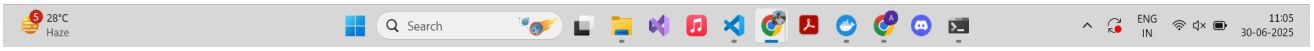

```yaml
version: '3.8'
services:
  web:
    build: ./app
    ports:
      - "3000:3000"
    depends_on:
      - mongo

  mongo:
    image: mongo:5
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db

volumes:
  mongo-data:
```

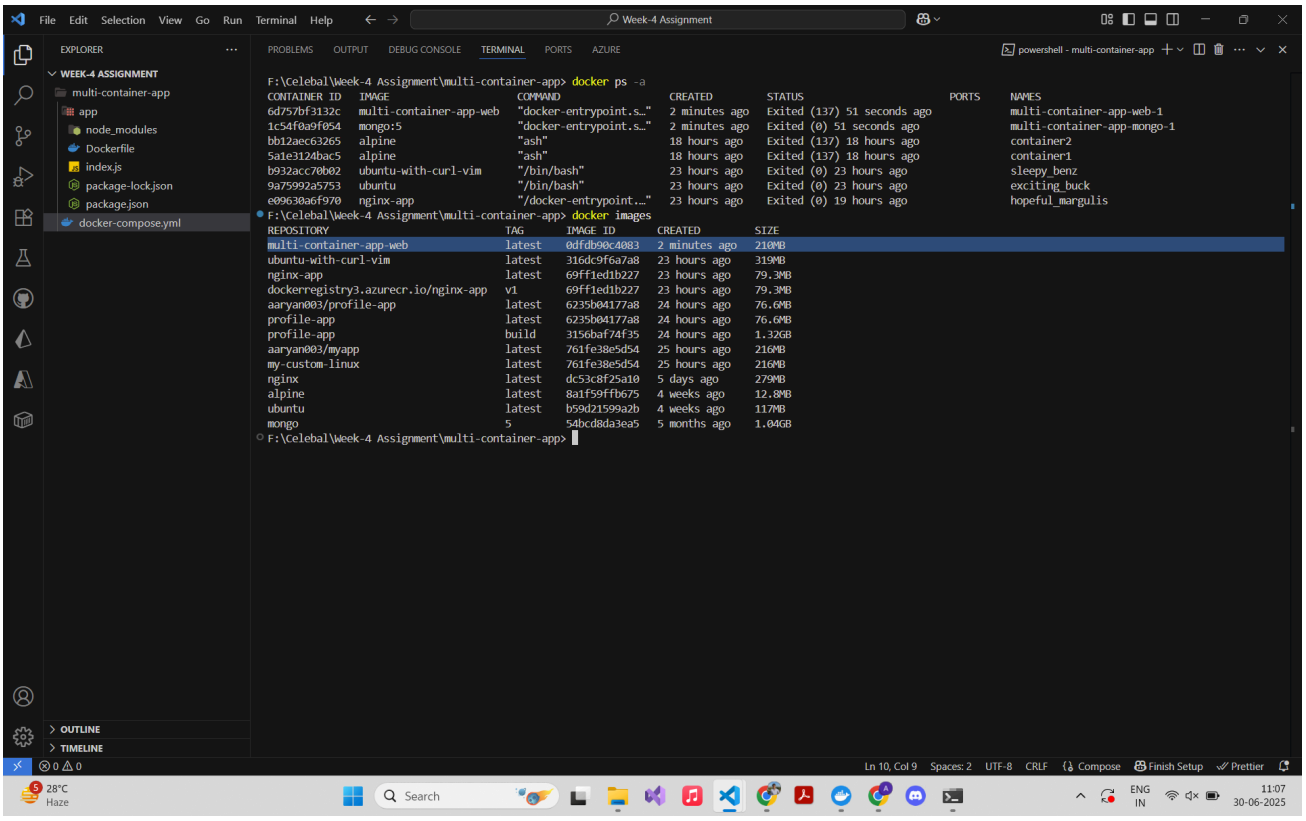
- Run the Compose Application



- Verify the output in the browser.

Hello from Node.js + MongoDB!





2. Docker Security Best Practices.

- Use Minimal Base Images
- Scan Images for Vulnerabilities