

# Docker Setup

--Bhavesh Dhake

- 
1. Enable Virtualization by following steps (2-6) :
  2. Restart the Computer and Press following keys according to your PC Brand to enter BIOS/UEFI :

// **F2** for Dell, Acer, Lenovo

// **Delete** for Asus, MSI

// **Esc , Shift** or **F10** HP

3. Look for a tab or section labeled Advanced, Processor Configuration, or CPU Configuration.

4. The option to enable virtualization may be listed as:

- **Intel Virtualization Technology (VT-x)** for Intel CPUs.
- **AMD-V** for AMD processors.
- **SVM Mode** (for some AMD motherboards).

5. Select the virtualization option and change it to Enabled.
6. Save the changes (usually by pressing F10) and exit the BIOS/UEFI.

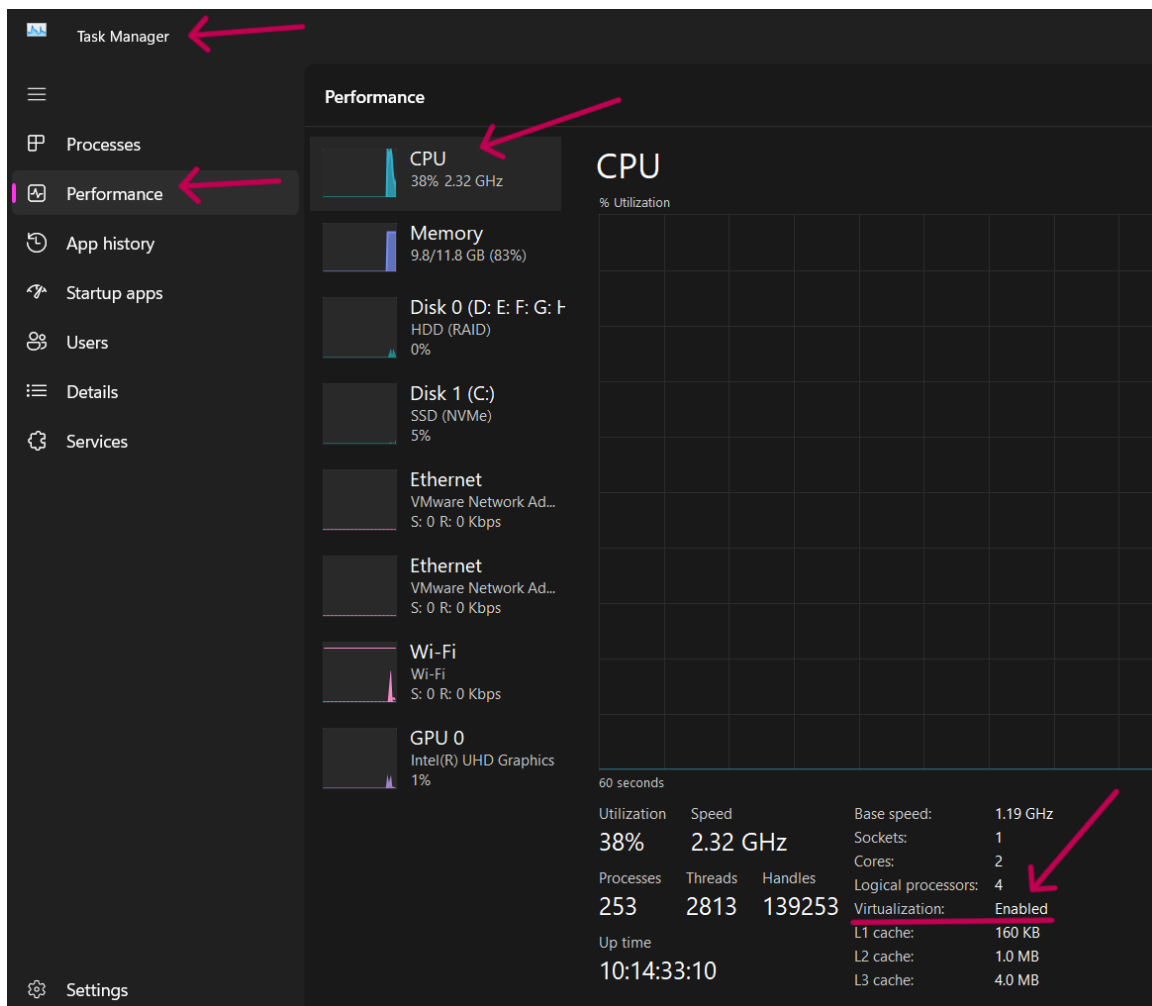
- 
7. To check if changes were made successfully follow steps(8-11) :

8. Open Task Manager

9. Go To Performance Tab

10. Go to CPU

11. At the Right Bottom there shall be showing “ Virtualization : Enabled”



---

**12. To setup WSL , follow steps (13-15 ):**

**13. Open “PowerShell” as Administrator**

**14. Enter command : wsl --install**

**15. Enter command : wsl --set-default-version 2**

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Ubuntu is already installed.
Launching Ubuntu...
bro@DESKTOP-SQCOUFF:~$ wsl --set-default-version 2
Command 'wsl' not found, but can be installed with:
sudo apt install wsl
bro@DESKTOP-SQCOUFF:~$ exit
logout
The operation completed successfully.
PS C:\WINDOWS\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
The operation completed successfully.
PS C:\WINDOWS\system32>
```

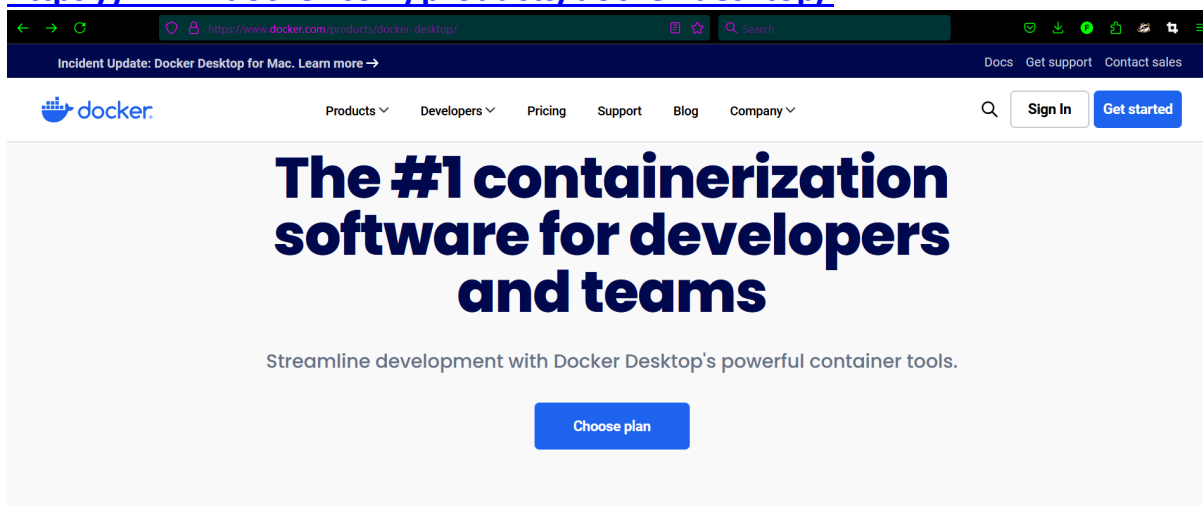
## 16. Restart ur PC

---

### 16.5. Steps to setup Docker ( 17-33) :

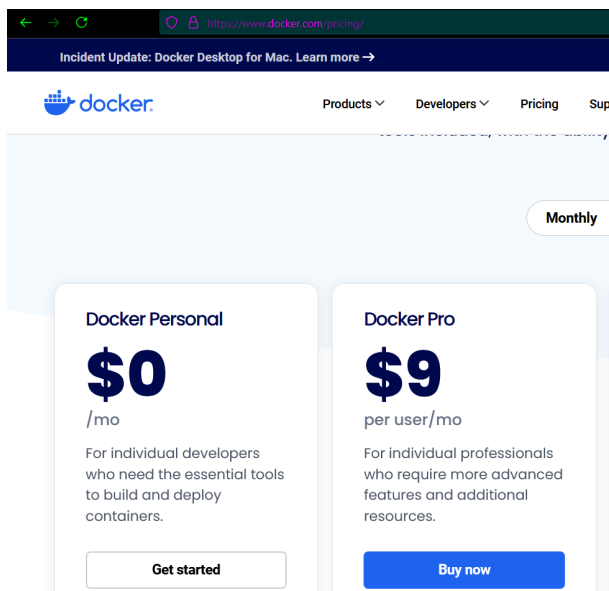
### 17. Go to official website of Docker :

<https://www.docker.com/products/docker-desktop/>

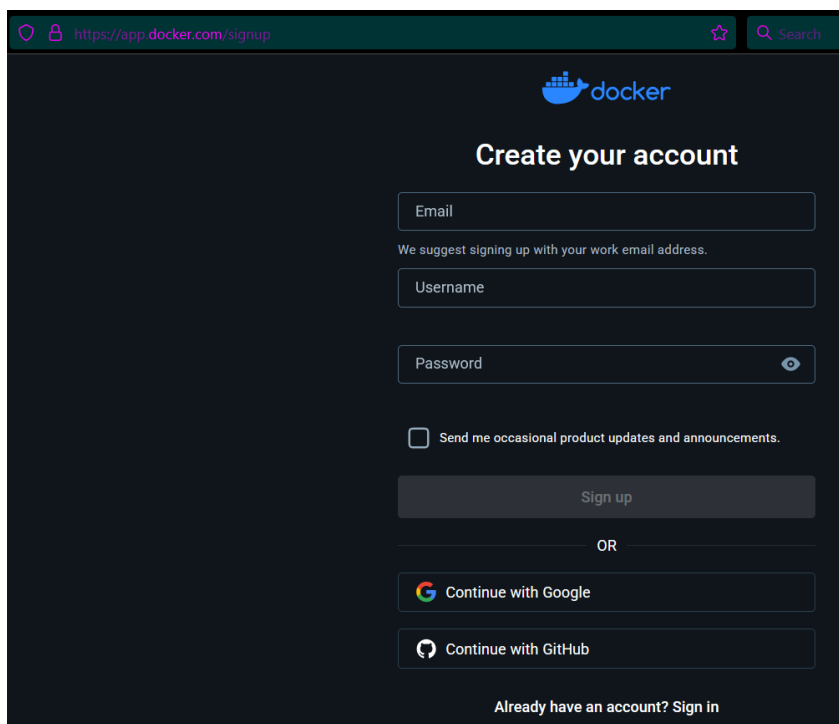


### 18. Click on “Choose plan”

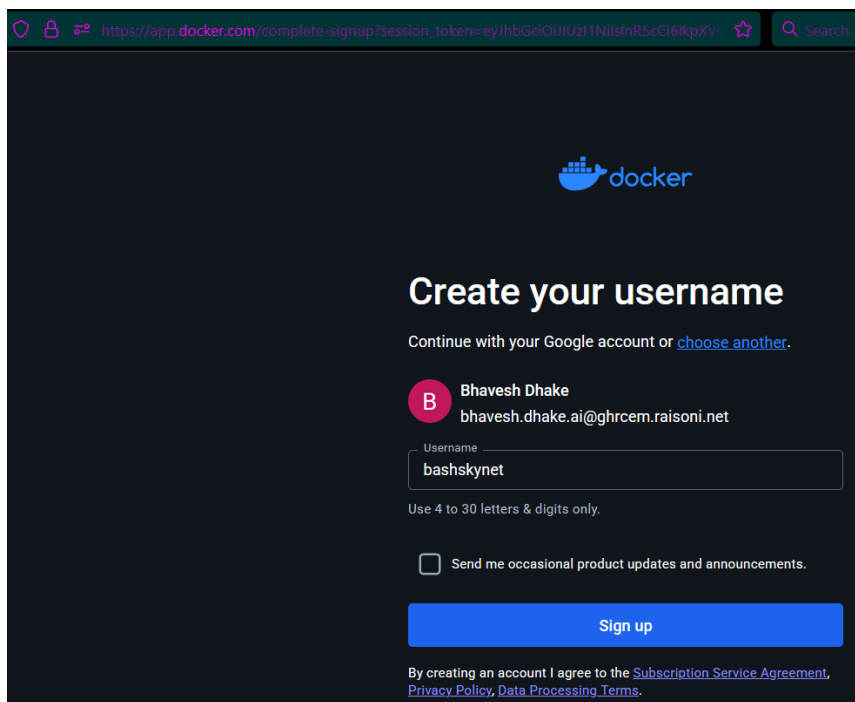
### 19. Select the \$0 plan Docker Personal and click on “ Get Started”



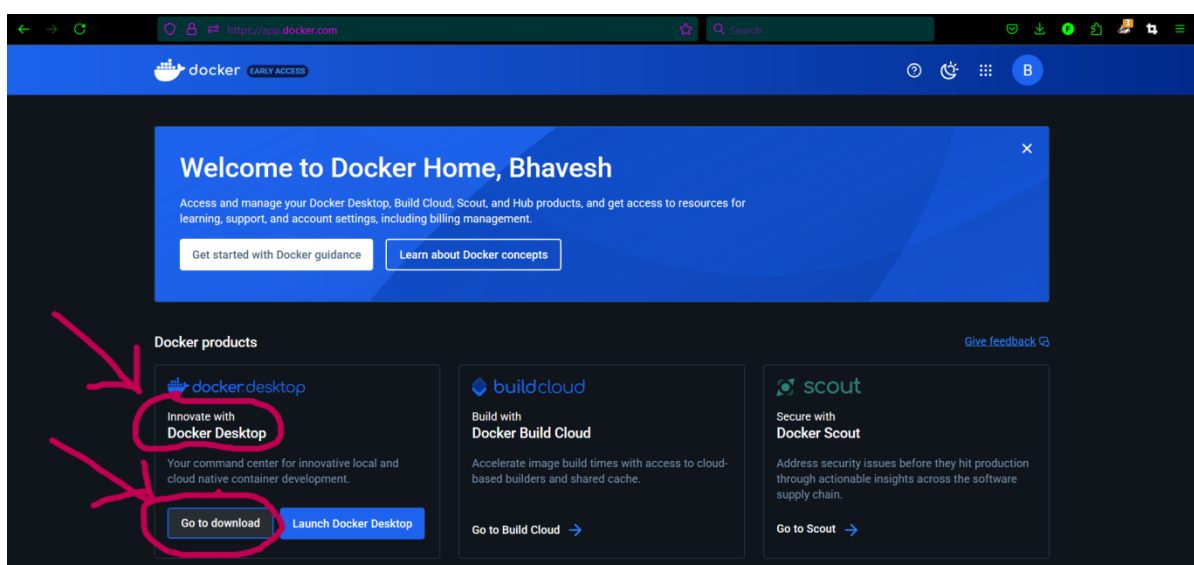
**20. You will be prompted to Sign up – Log in Page , complete the process with either ur gmail id or via google Auth**



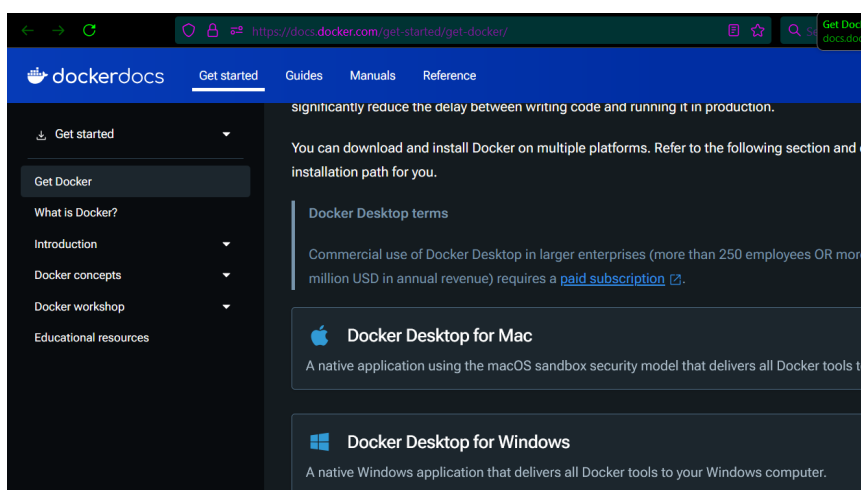
**21. Enter username**



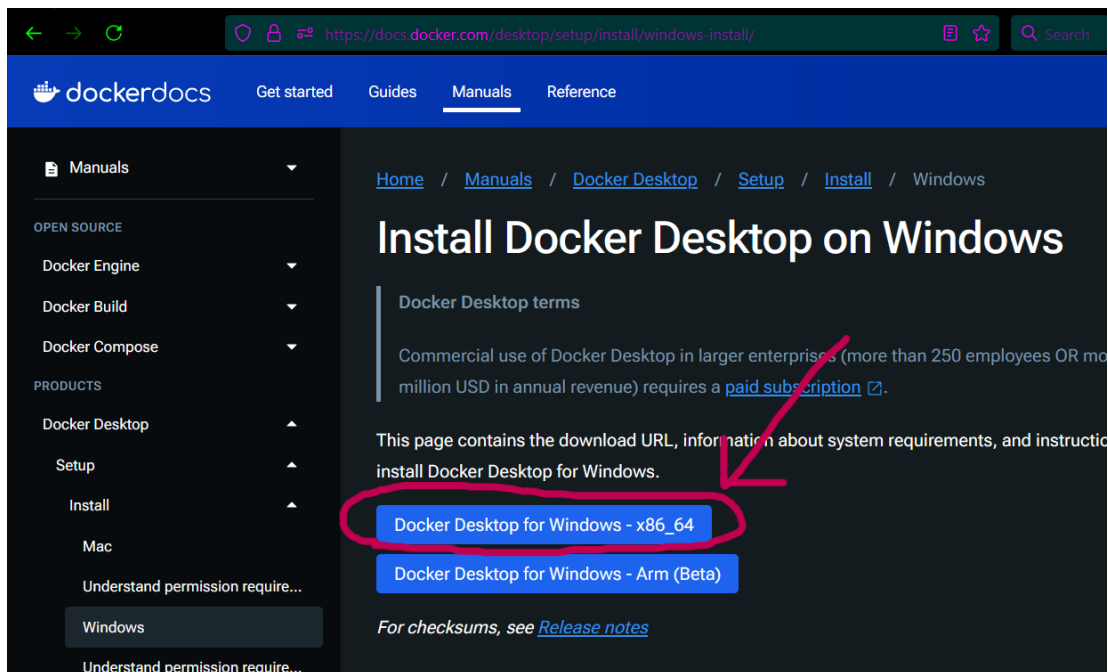
22. u will be prompted Home Page , Click on “Go to Download” under “Docker Desktop”



23. Scroll down to see Windows option , click on it :

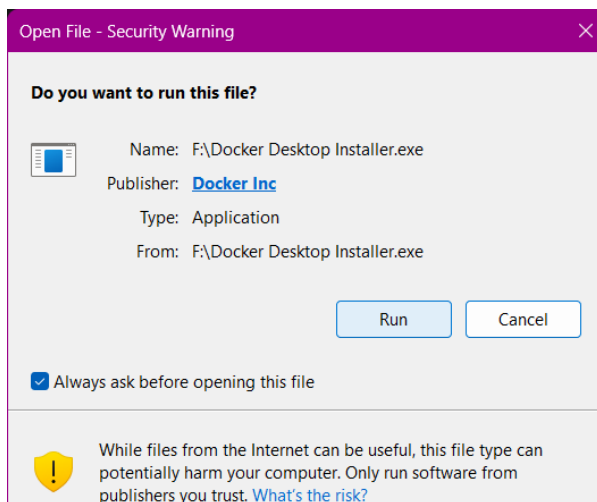


## 24. Select X86-64

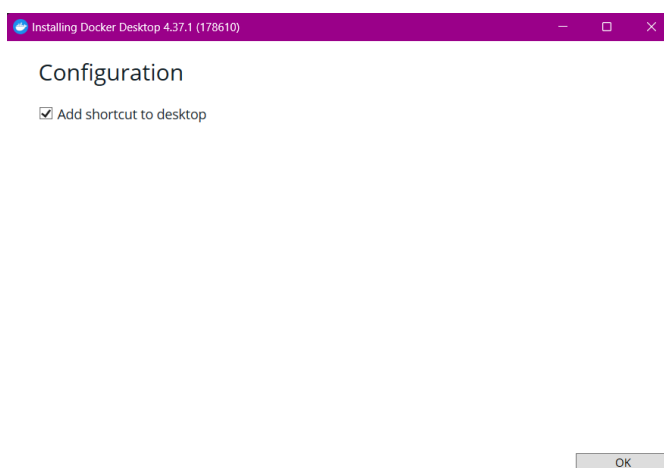


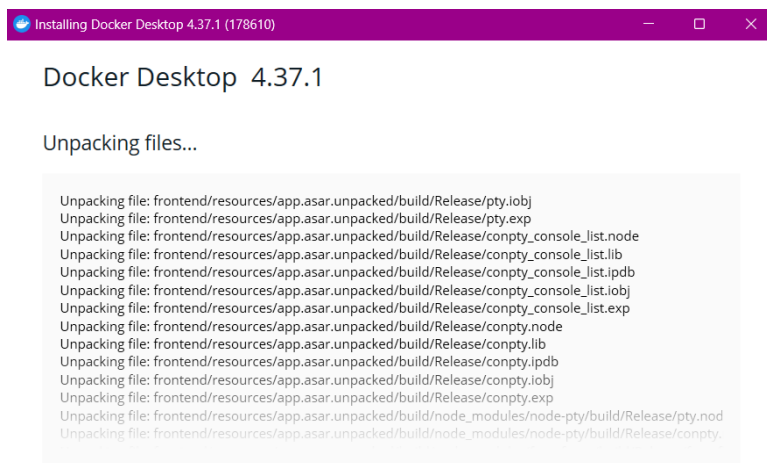
## 25. Locate the Downloaded file : “Docker Desktop Installer.exe” and launch

## 26. Click run :

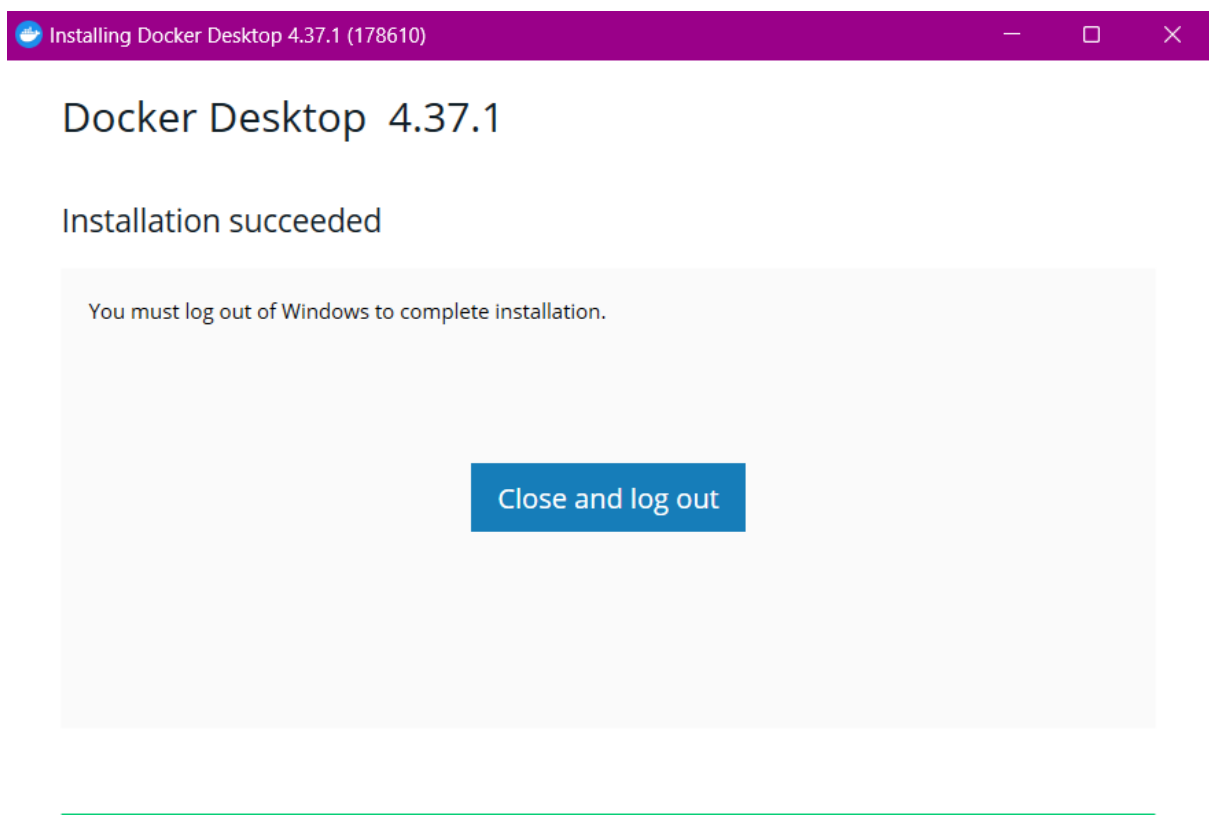


## 27. Click ok

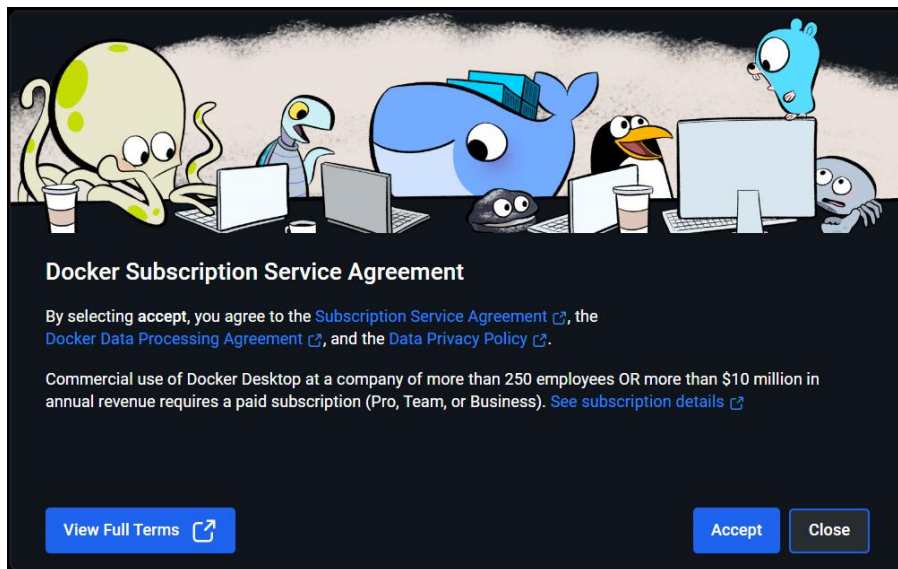




## 28. Click Close and log out and PC will restart



## 29. Upon restarting this window will appear , click accept



30. If an error dialogue box appears saying wsl error due to pre existing other os wsl , then simple remove the non required ones by usinf command “wsl -l -v” to view all existing ones and use command “wsl --unregister <name>” to remove it , and then click on “ skip <other> distro.

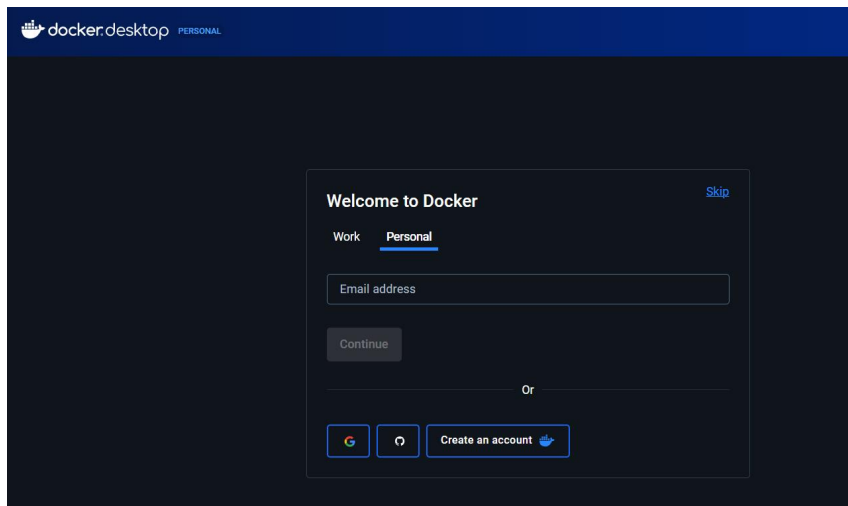
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl -l -v
  NAME                STATE      VERSION
*  kali-linux          Stopped    2
   Ubuntu              Stopped    2
   docker-desktop      Running    2
PS C:\WINDOWS\system32> wsl --set-version docker-desktop 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
Conversion in progress, this may take a few minutes.
The distribution is already the requested version.
Error code: Wsl/Service/WSL_E_VM_MODE_INVALID_STATE
PS C:\WINDOWS\system32> wsl --unregister kali-linux
Unregistering.
The operation completed successfully.
PS C:\WINDOWS\system32> _
```

31. On home of Docker opened earlier after restart select “personal” and sign in with google (use same gmail account used for WSL)





**32. u will be redirected to docker webpage of ur default browser and then back to application , complete/skip the survey**

**33. To ensure docker has been successfully setup open Command Prompt and run Following commands : “docker --version” and “docker run hello-world”**

```
Command Prompt

C:\Users\admin>docker --version
Docker version 27.4.0, build bde2b89

C:\Users\admin>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:d715f14f9eca81473d9112df50457893aa4d099adeb4729f679006bf5ea12407
Status: Downloaded newer image for hello-world:latest

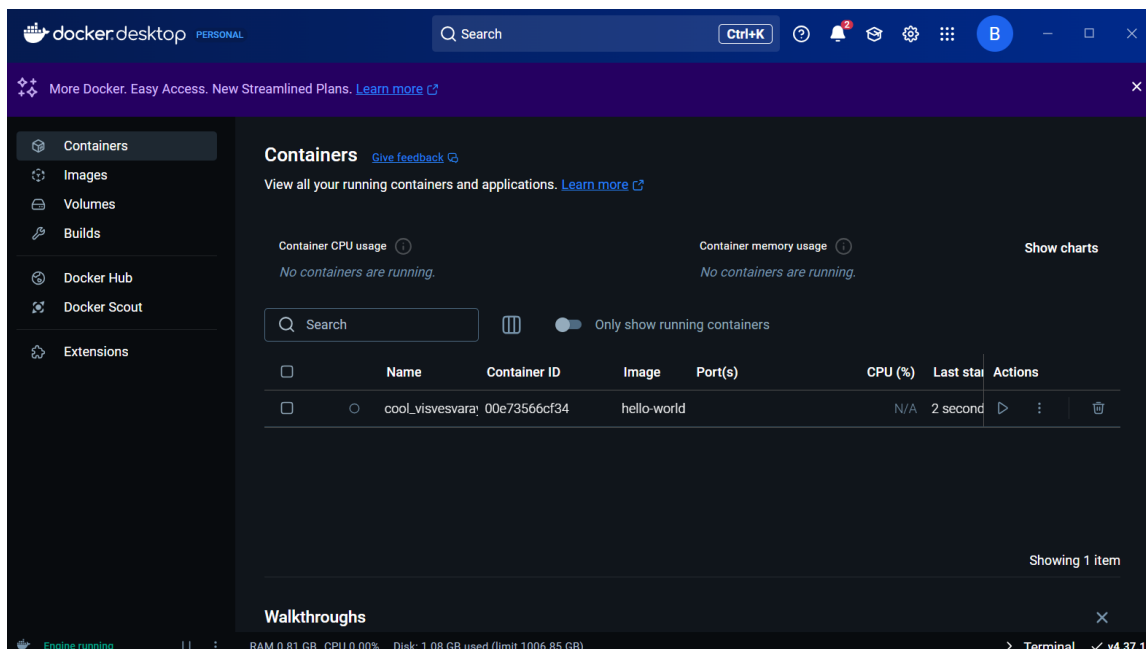
Hello from Docker!
This message shows that your installation appears to be working correctly.

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

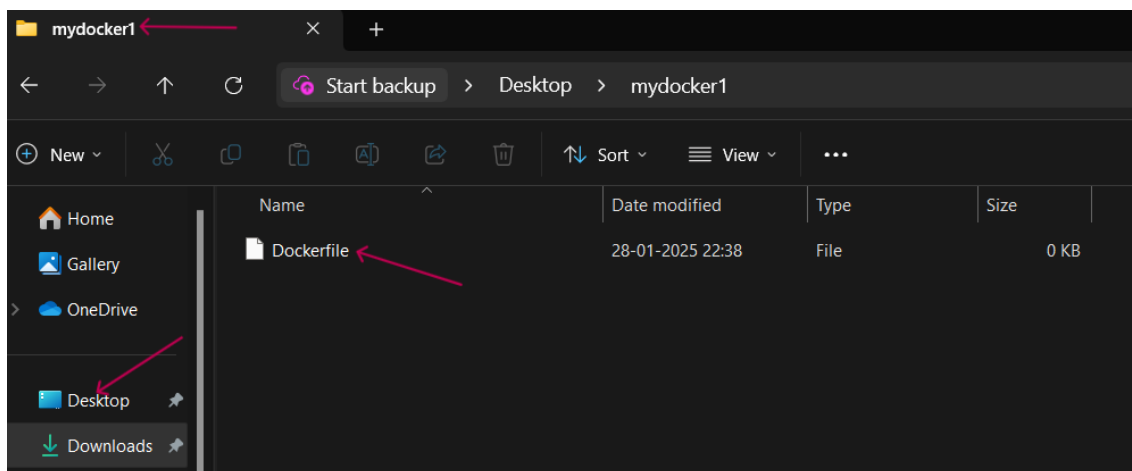
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
```



34. Steps to run first Docker image ( 35- ) :

35. Go to Desktop and create new folder named “mydocker1” and in that folder create new text document and name it “Dockerfile” and remove the .txt extension while saving (no extension)



36. Right click on “Dockerfile” and select open with notepad and write following code and save and close :

FROM python:3.9-slim

WORKDIR /app

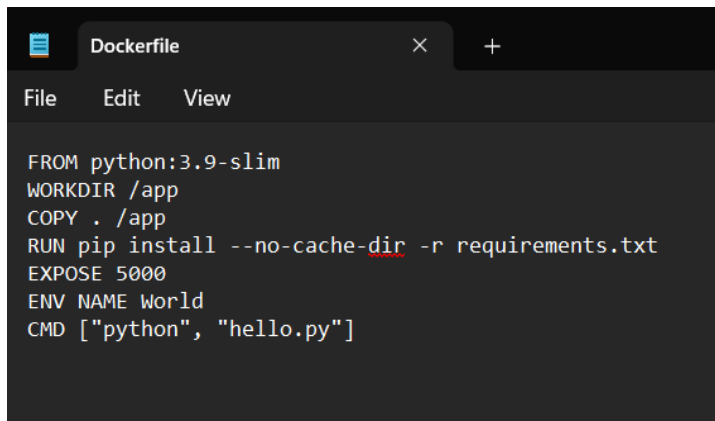
COPY . /app

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 5000

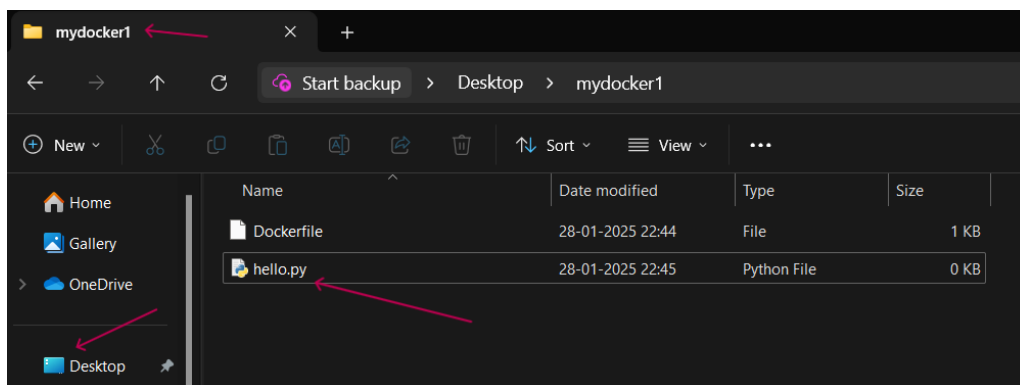
ENV NAME World

CMD ["python", "hello.py"]



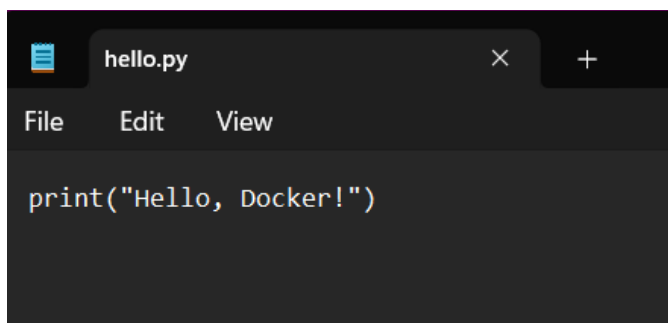
```
FROM python:3.9-slim
WORKDIR /app
COPY . /app
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 5000
ENV NAME World
CMD ["python", "hello.py"]
```

**37. Again go into the “mydocker1” directory and make new text file with name “hello.py”**



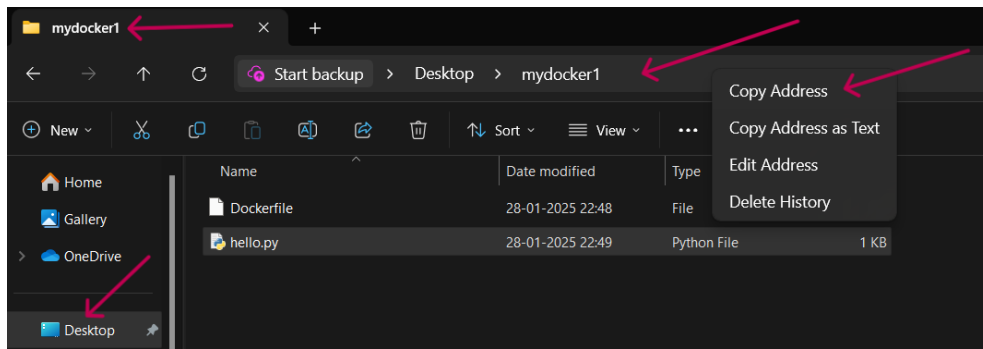
**38. right click hello.py and open with notepad and write following code in it and save and close :**

**print("Hello, Docker!")**



```
print("Hello, Docker!")
```

**39. Go in directory “mydocker1” and right click on path and click “copy address”**



**40. Open Command Prompt and enter the following command :**

**cd <paste the path u copied>**

**in my case it's : cd C:\Users\admin\Desktop\mydocker1**

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>cd C:\Users\admin\Desktop\mydocker1

C:\Users\admin\Desktop\mydocker1>_
```

**41. Build the Docker image using this command :**

**docker build -t my-first-image .**

**42. If u encounter an error like this for the line to install required dependancies , follow steps (43- 44 ) :**

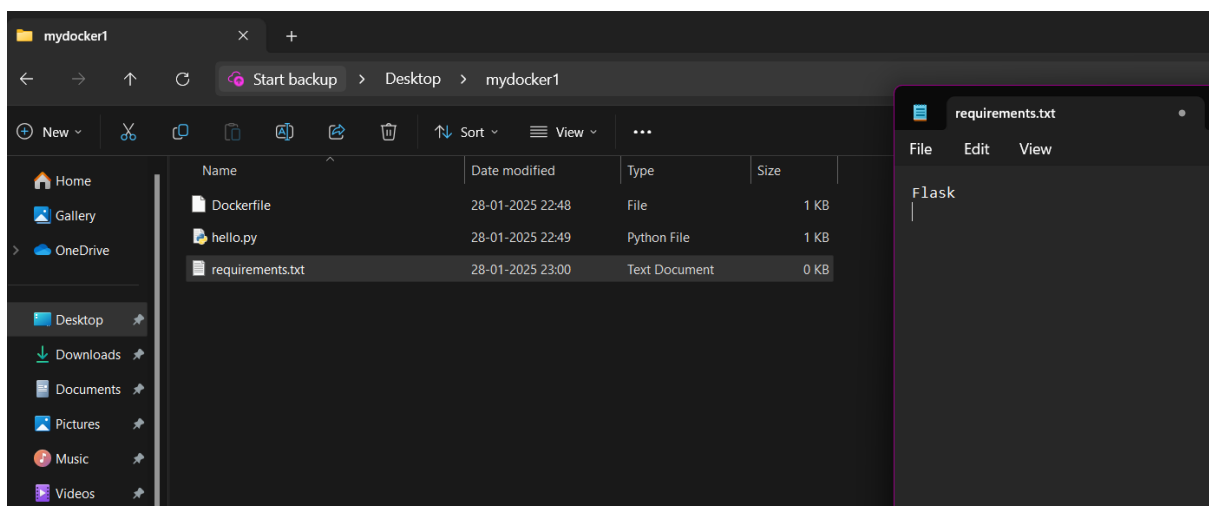
```
Command Prompt
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>cd C:\Users\admin\Desktop\mydocker1

C:\Users\admin\Desktop\mydocker1>docker build -t my-first-image .
[*] Building 16.6s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 197B                               0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 2.8s
=> [auth] library/python:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                     0.0s
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:bb8009c87ab69e751a1dd2c6c7f8abaae3d9fce8e072802d4a23c955 10.6s
=> => resolve docker.io/library/python:3.9-slim@sha256:bb8009c87ab69e751a1dd2c6c7f8abaae3d9fce8e072802d4a23c9559 0.0s
=> => sha256:1da0723265ec311debef6bec17d4fae5f1e5f7809fca4378aac265cdef238f1c 3.51MB / 3.51MB 4.0s
=> => sha256:4f4cb1a24c66f1a92f204ba0bbd6d2a7c941a853c83161ffa38bbfa121448861 14.93MB / 14.93MB 7.8s
=> => sha256:bb8009c87ab69e751a1dd2c6c7f8abaae3d9fce8e072802d4a23c95594d16d84 10.41kB / 10.41kB 0.0s
=> => sha256:ddb56f2e39ec00c79c1207f182c60e03ddfd417525b56ef467e1519706792cd 1.75kB / 1.75kB 0.0s
=> => sha256:453d3342b002fa5f904ba0cd72a07accb5121641d20776f3e64339842f275d38 5.28kB / 5.28kB 0.0s
=> => sha256:af302e5c37e9dc1dbe2eadc8f5059d82a914066b541b0d1a6daa91d0cc55057d 28.21MB / 28.21MB 6.6s
=> => sha256:c876ae22765e4a125855eb121718c3f8f07bd8b0dae0ad4e68e716571961f37 249B / 249B 4.5s
=> => extracting sha256:af302e5c37e9dc1dbe2eadc8f5059d82a914066b541b0d1a6daa91d0cc55057d 2.2s
=> => extracting sha256:1da0723265ec311debef6bec17d4fae5f1e5f7809fca4378aac265cdef238f1c 0.2s
=> => extracting sha256:4f4cb1a24c66f1a92f204ba0bbd6d2a7c941a853c83161ffa38bbfa121448861 1.1s
=> => extracting sha256:c876ae22765e4a125855eb121718c3f8f07bd8b0dae0ad4e68e716571961f37 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 259B                                   0.0s
=> [2/4] WORKDIR /app                                           0.4s
=> [3/4] COPY . /app                                           0.0s
=> ERROR [4/4] RUN pip install --no-cache-dir -r requirements.txt 2.5s
-----
> [4/4] RUN pip install --no-cache-dir -r requirements.txt:
1.922 ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
2.237
2.237 [notice] A new release of pip is available: 23.0.1 -> 25.0
2.237 [notice] To update, run: pip install --upgrade pip
-----

1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 6)
Dockerfile:4
-----
 2 | WORKDIR /app
 3 | COPY . /app
 4 | >>> RUN pip install --no-cache-dir -r requirements.txt
 5 | EXPOSE 5000
 6 | ENV NAME World
-----
ERROR: failed to solve: process "/bin/sh -c pip install --no-cache-dir -r requirements.txt" did not complete successfully: exit code: 1
```

43. in the directory “mydocker1” create a new file “requirements.txt” , open it using notepad and in that txt file write name of python dependancies required , for example flask and save and close :



44. Re-run command : “docker build -t my-first-image .”

```
Command Prompt

C:\Users\admin\Desktop\mydocker1>docker build -t my-first-image .
[+] Building 6.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 197B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:bb8009c87ab69e751a1dd2c6c7f8abaae3d9fce8e072802d4a23c95594d16d84
=> [internal] load build context
=> => transferring context: 111B
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:5f59ca7977a61dda61557159d025d7319533fd23154f8b17ecf1ca3c0fa251ae
=> => naming to docker.io/library/my-first-image

1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 6)


C:\Users\admin\Desktop\mydocker1>
```

## 45. run the image using command : “docker run my-first-image”

```
Command Prompt






C:\Users\admin\Desktop\mydocker1>docker run my-first-image
Hello, Docker!

C:\Users\admin\Desktop\mydocker1>
```

 docker desktop

PERSONAL

Ctrl+K

More Docker. Easy Access. New Streamlined Plans. [Learn more](#)

Containers

Images

Volumes

Builds

Docker Hub

Docker Scout

Extensions



Containers [Give feedback](#)







View all your running containers and applications. [Learn more](#)

Container CPU usage ⓘ  
No containers are running.

Container memory usage ⓘ  
No containers are running.


Show charts

  Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/>	cool_visvesvara	00e73566cf34	hello-world		N/A	37 minut	  
<input type="checkbox"/>	angry_nobel	f250e11f8260	my-first-iru		N/A	1 second	  

Showing 2 items

Walkthroughs



Engine running

RAM 0.97 GB CPU 0.00% Disk: 1.23 GB used (limit 1006.85 GB)

Terminal

v4.37.1

