Experiment-4

Prepared By

Bhuvnesh Sanathra

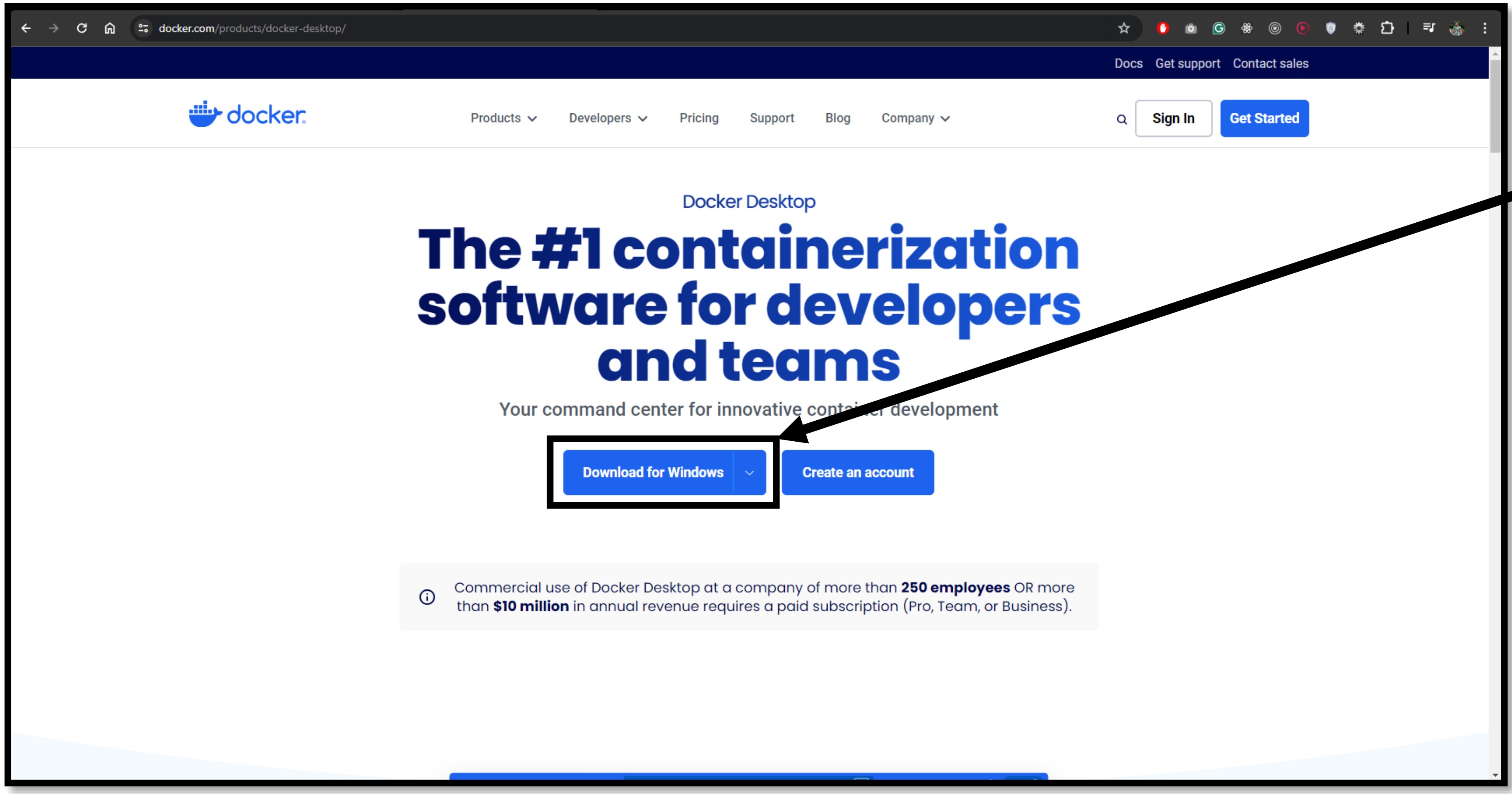
Semester-7

Department of Information Technology
Faculty of technology,
Dharmsinh Desai University
College road, Nadiad- 387001

Installation of Docker Desktop

Docker Desktop Installation

For downloading the Docker Desktop visit https://www.docker.com/products/docker-desktop/



docker.com/products/docker-desktop/
Docs Get support Contact sales
docker
Products Developers Pricing Support Blog Company
Sign In
Get Started
Download for Windows
Create an account

Docker Desktop

The #1 containerization software for developers and teams

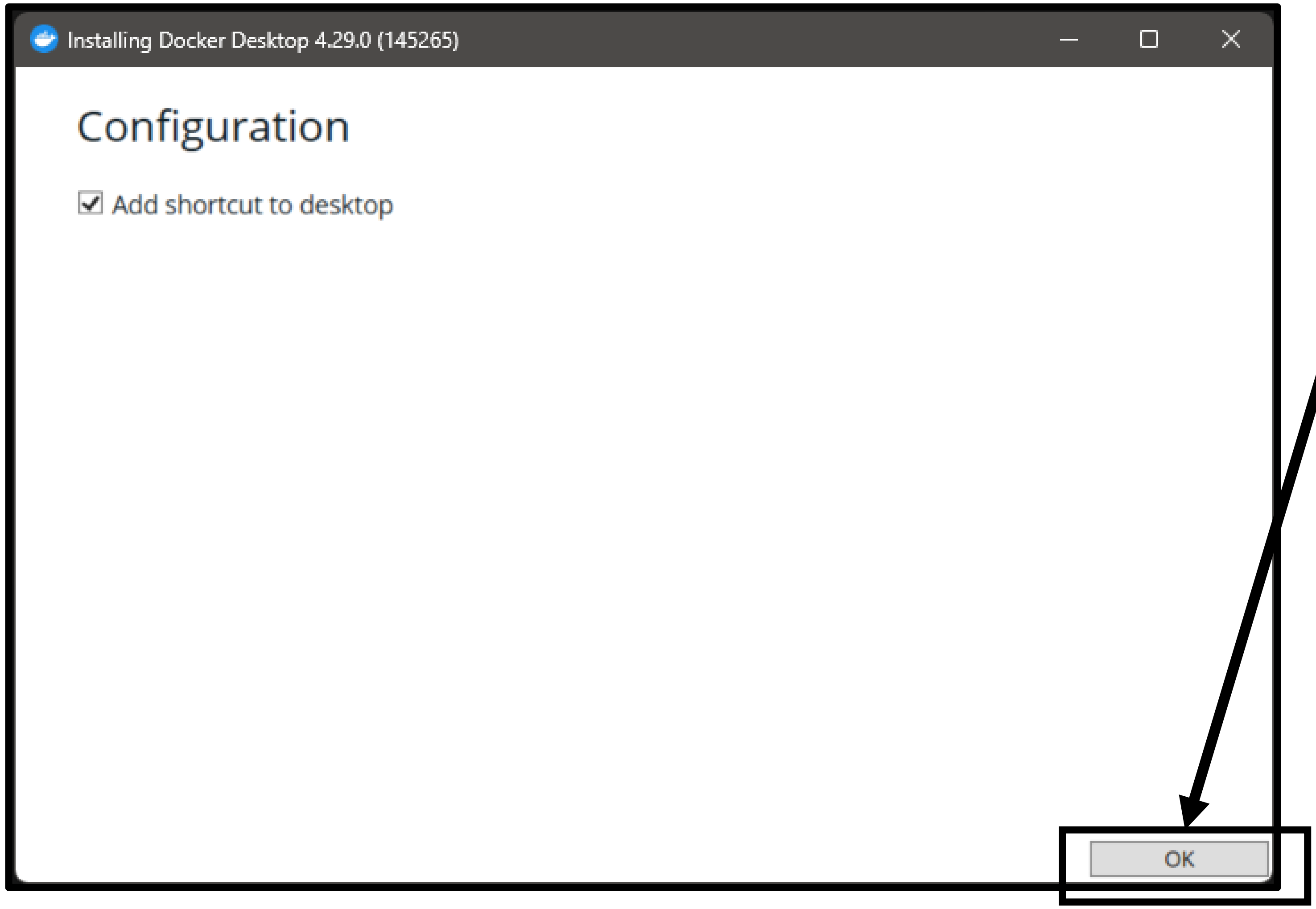
Your command center for innovative container development

Commercial use of Docker Desktop at a company of more than **250 employees** OR more than **$10 million** in annual revenue requires a paid subscription (Pro, Team, or Business).

Click on **Download For Windows**

Docker Desktop Installation

Open the setup once installation is done



Installing Docker Desktop 4.29.0 (145265)

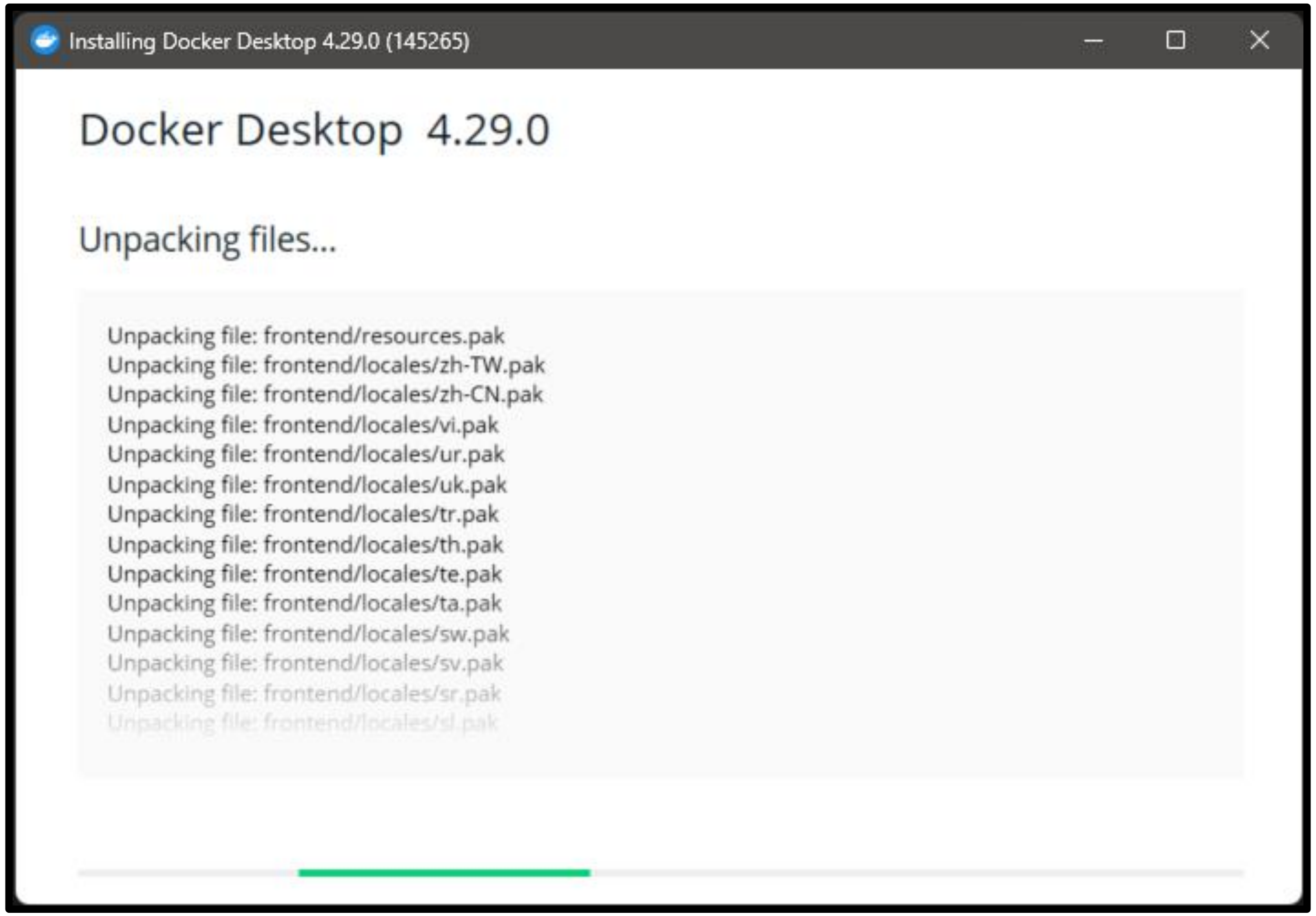
Configuration

☑ Add shortcut to desktop

Click **OK**

Docker Desktop Installation

Installation Started



Installing Docker Desktop 4.29.0 (145265)

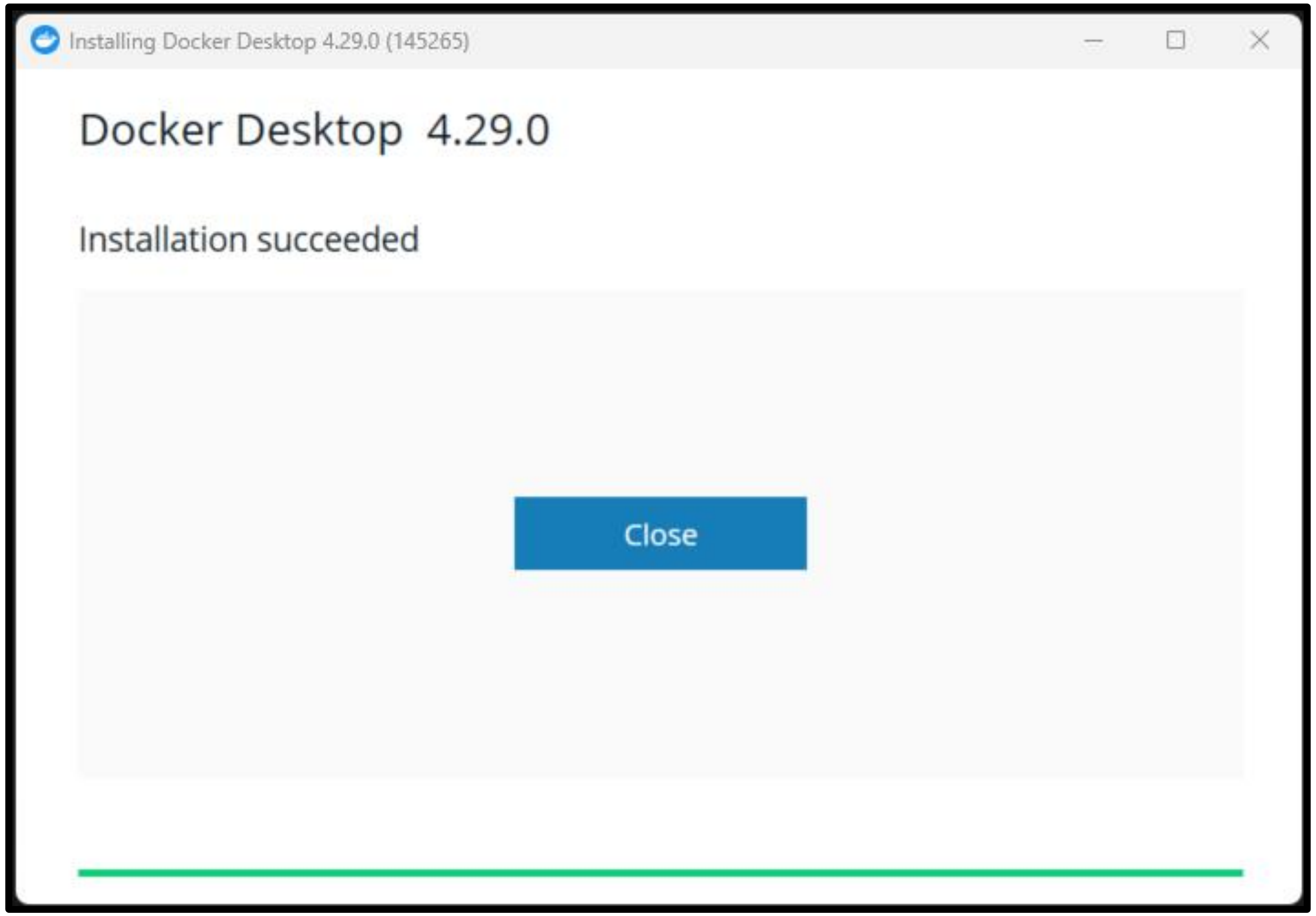
Docker Desktop 4.29.0

Unpacking files...

```
Unpacking file: frontend/resources.pak
Unpacking file: frontend/locales/zh-TW.pak
Unpacking file: frontend/locales/zh-CN.pak
Unpacking file: frontend/locales/vi.pak
Unpacking file: frontend/locales/ur.pak
Unpacking file: frontend/locales/uk.pak
Unpacking file: frontend/locales/tr.pak
Unpacking file: frontend/locales/th.pak
Unpacking file: frontend/locales/te.pak
Unpacking file: frontend/locales/ta.pak
Unpacking file: frontend/locales/sw.pak
Unpacking file: frontend/locales/sv.pak
Unpacking file: frontend/locales/sr.pak
Unpacking file: frontend/locales/sl.pak
```

Docker Desktop Installation

Installation Done!

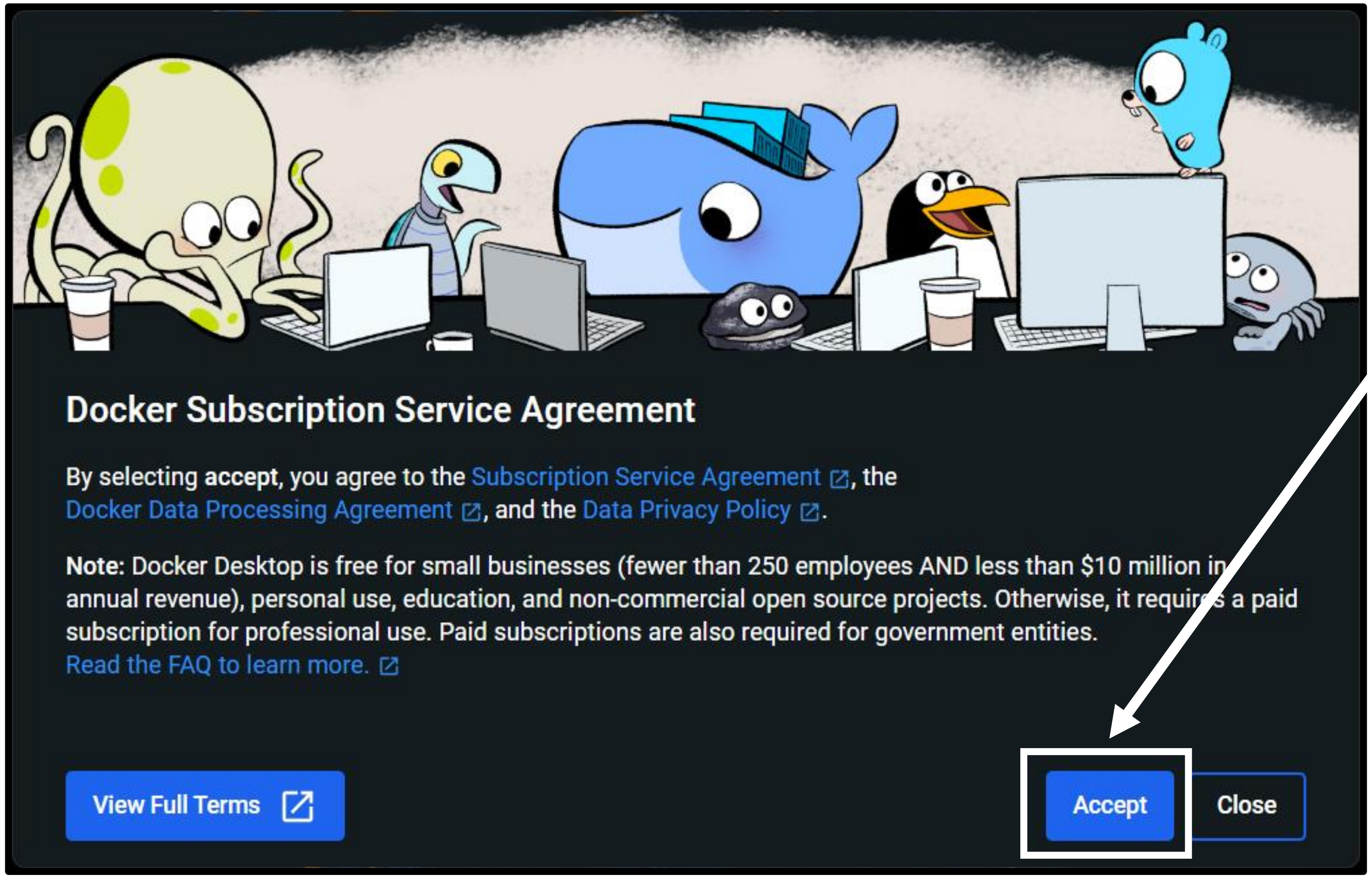

Installing Docker Desktop 4.29.0 (145265)

Docker Desktop 4.29.0

Installation succeeded

Docker Desktop Installation

Open the **Docker Desktop** & **Accept** it



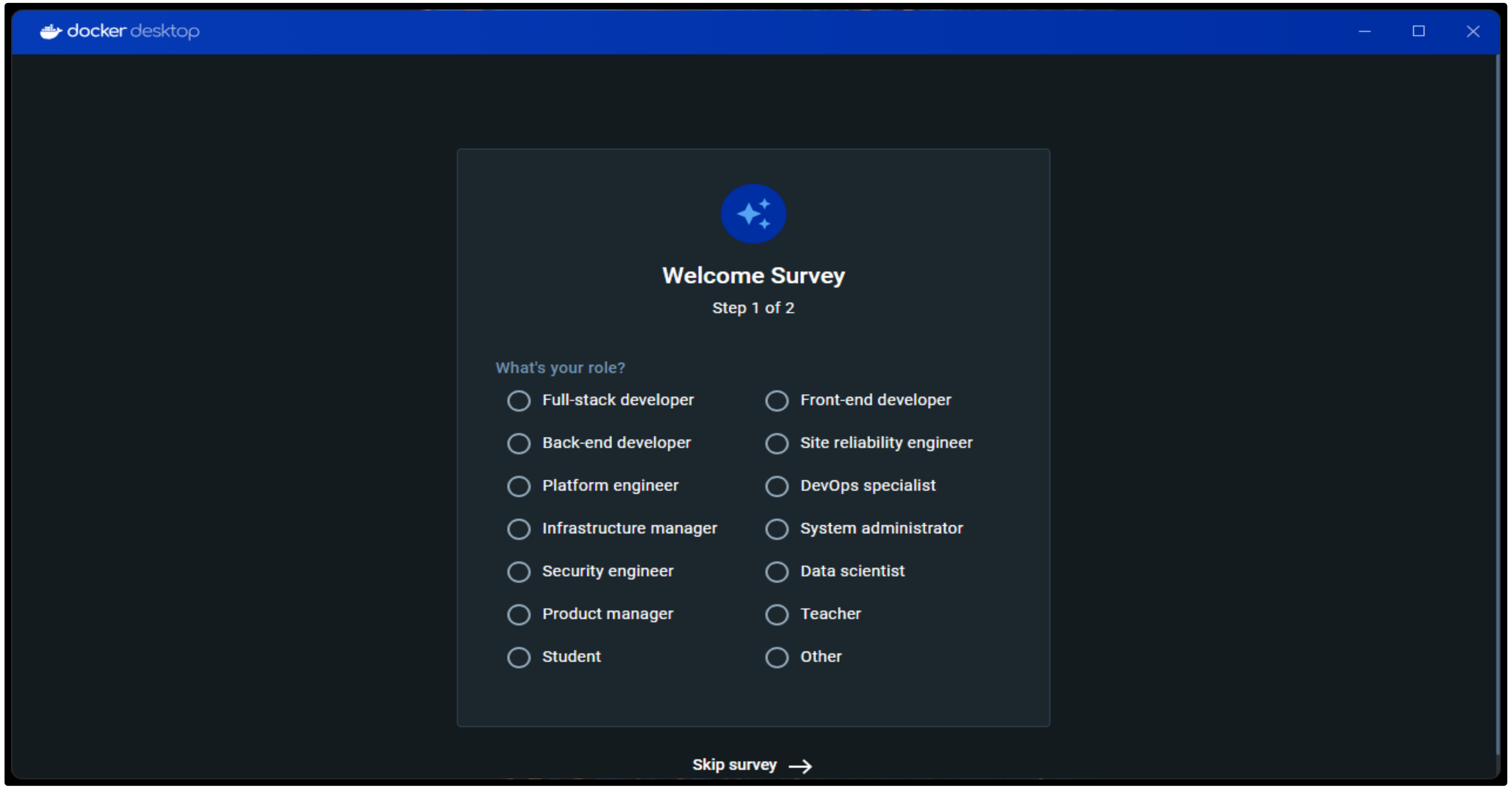
Docker Subscription Service Agreement

By selecting **accept**, you agree to the Subscription Service Agreement, the Docker Data Processing Agreement, and the Data Privacy Policy.

Note: Docker Desktop is free for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects. Otherwise, it requires a paid subscription for professional use. Paid subscriptions are also required for government entities.
Read the FAQ to learn more.

Accept

Docker Desktop Installation



Welcome Survey

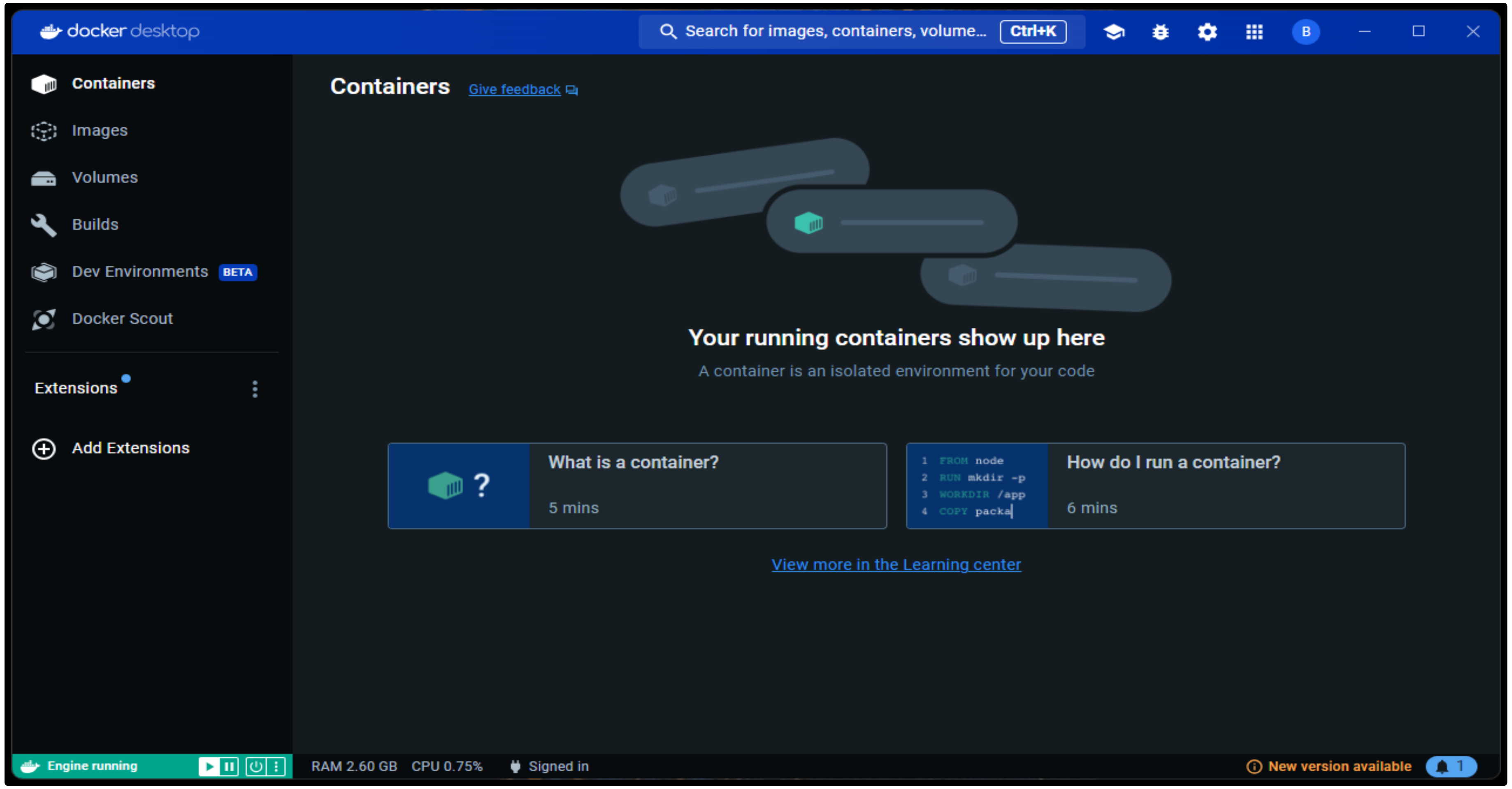
Step 1 of 2

What's your role?

- ○ Full-stack developer
- ○ Front-end developer
- ○ Back-end developer
- ○ Site reliability engineer
- ○ Platform engineer
- ○ DevOps specialist
- ○ Infrastructure manager
- ○ System administrator
- ○ Security engineer
- ○ Data scientist
- ○ Product manager
- ○ Teacher
- ○ Student
- ○ Other

If you want to perform **Survey** select role else **Skip Survey**

Docker Desktop Installation



Containers Give feedback

Your running containers show up here

A container is an isolated environment for your code

What is a container?

5 mins

```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

How do I run a container?

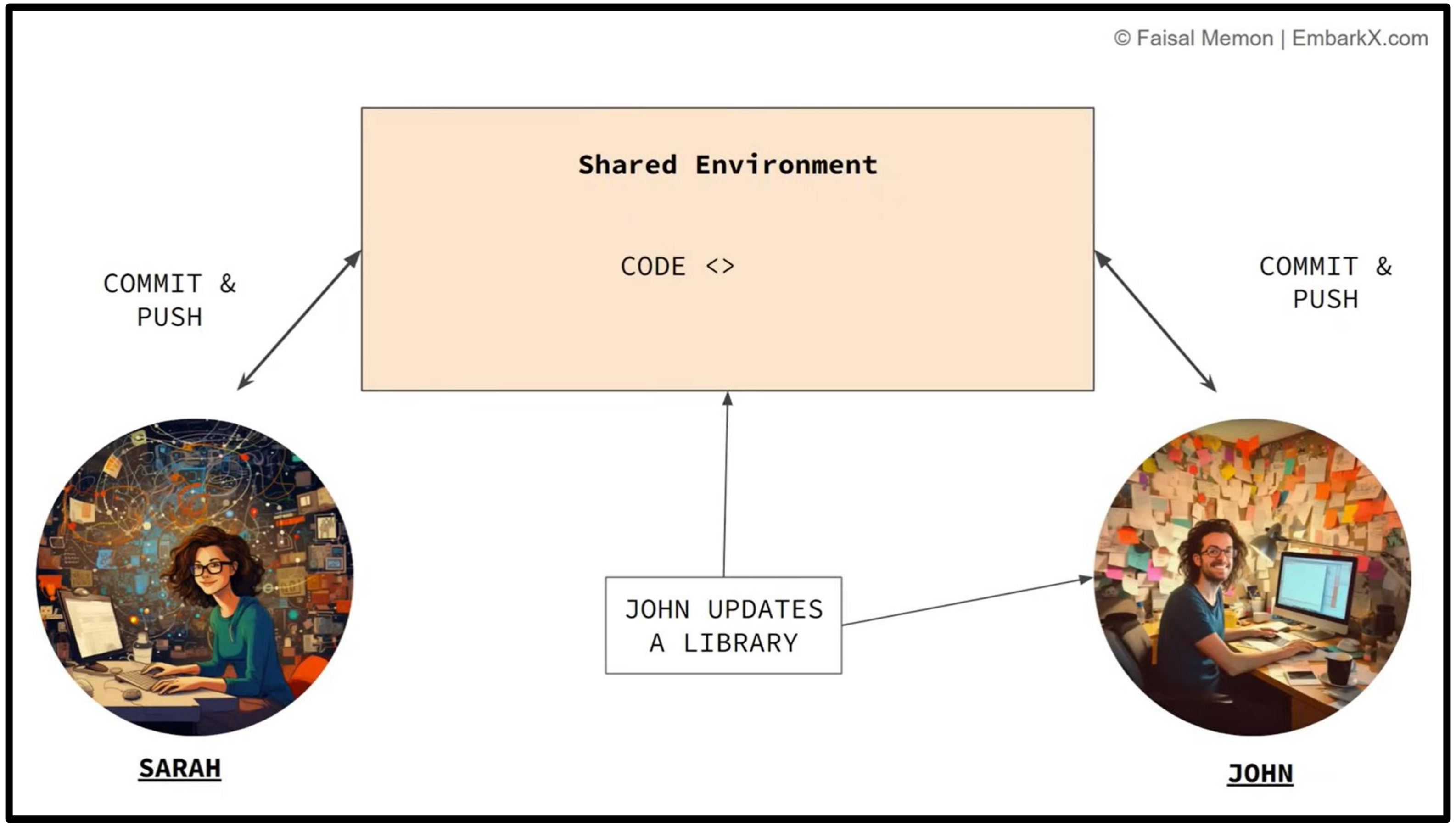
6 mins

View more in the Learning center

Docker Desktop is installed successfully.

Docker

Why Docker?



2 Person are working on same code base and let say JOHN changes the Version of library used in shared Environment. Now, it is difficult to manage the version of different user in same shared environment.

Why Docker?

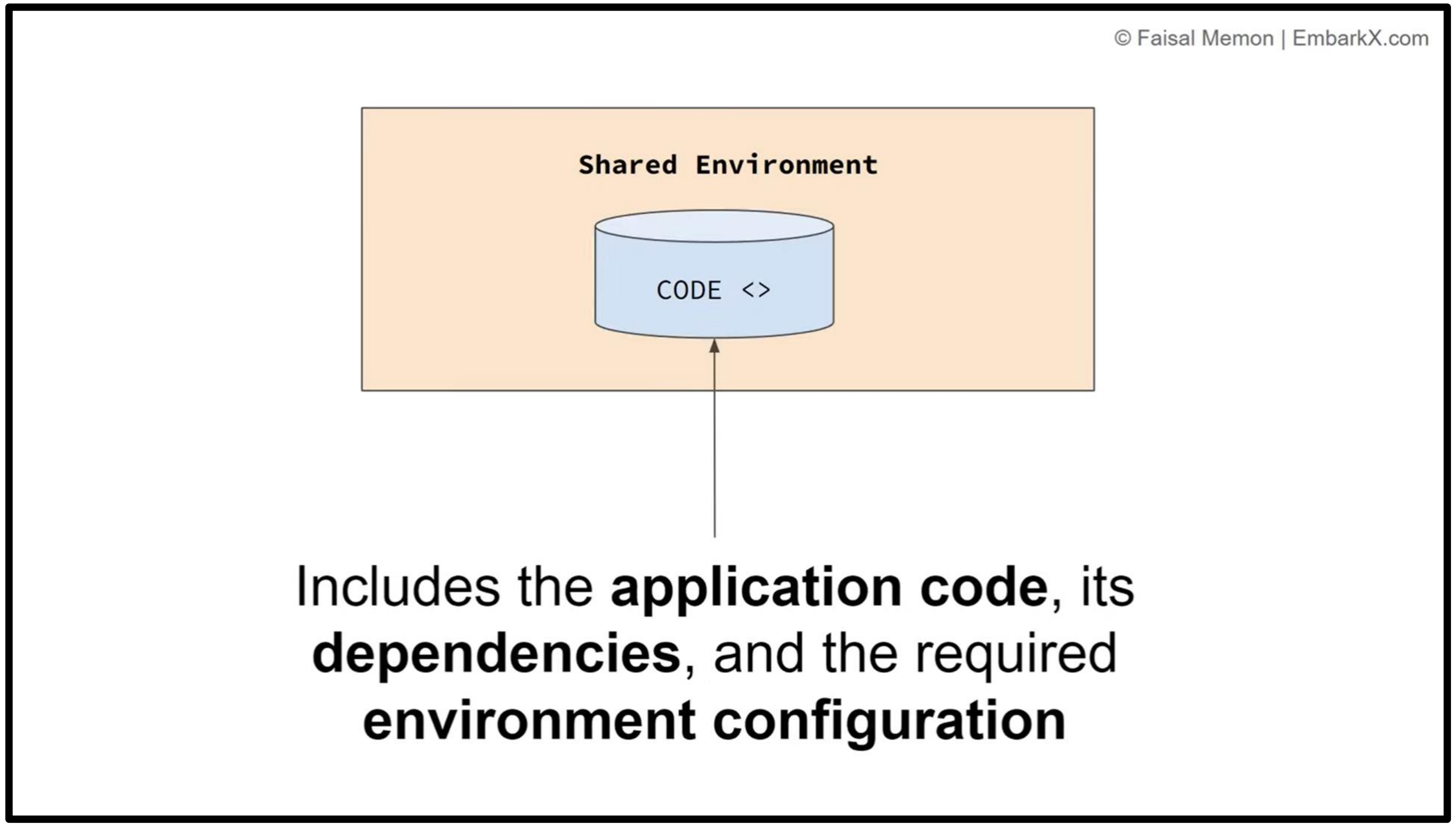
- It causes version conflict or compatibility problem.
- Which then causes the inconsistency in development.
- Application is working fine in one development environment but not in another developer environment.

What is solution to this??

Docker

HOW??

How?



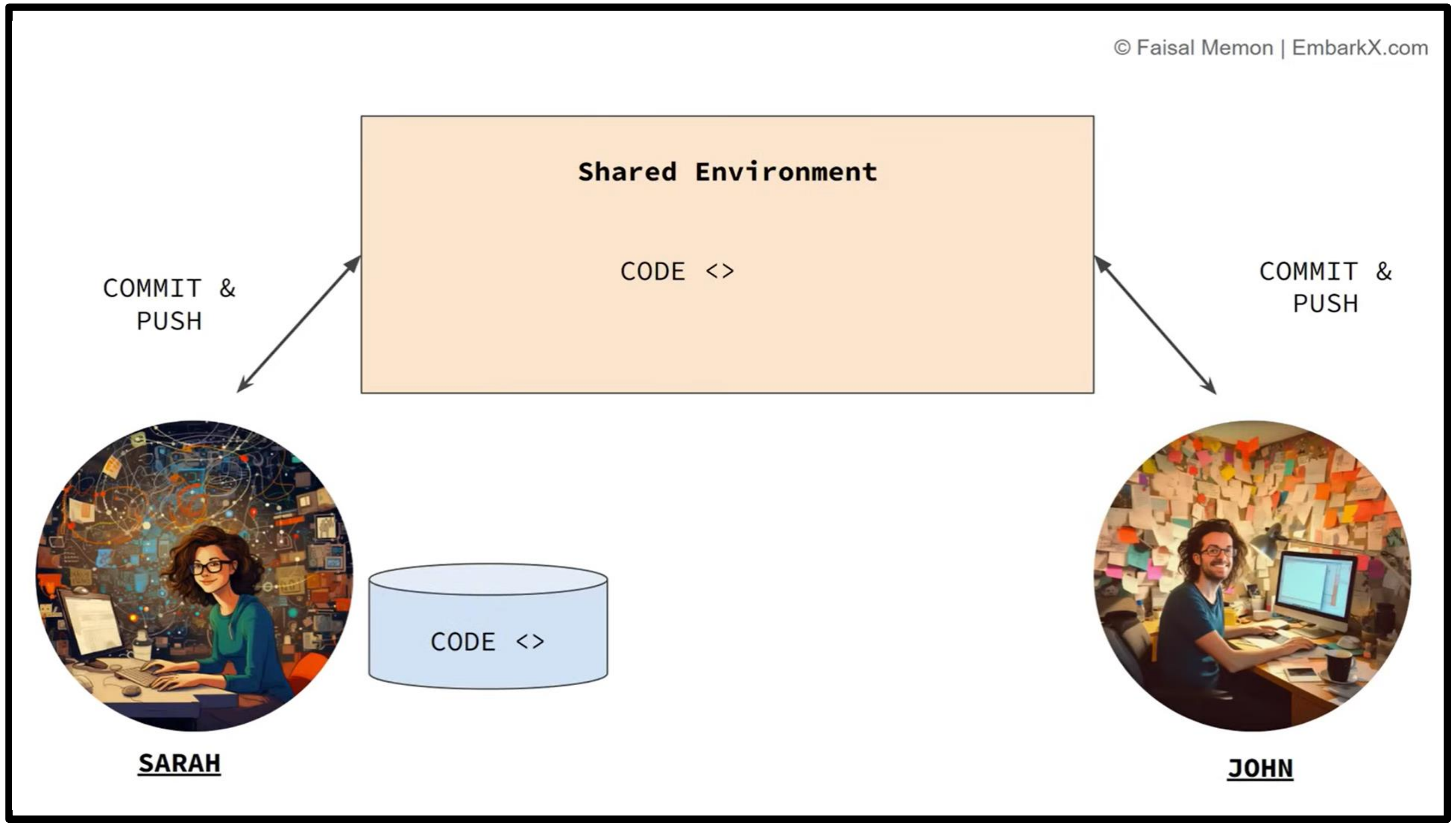
Includes the **application code**, its **dependencies**, and the required **environment configuration**

Containerized the application using **Docker.**

What does it means?

- Move codebase to container.
- Which includes **application code, dependencies, and the required environment configuration.**

How?

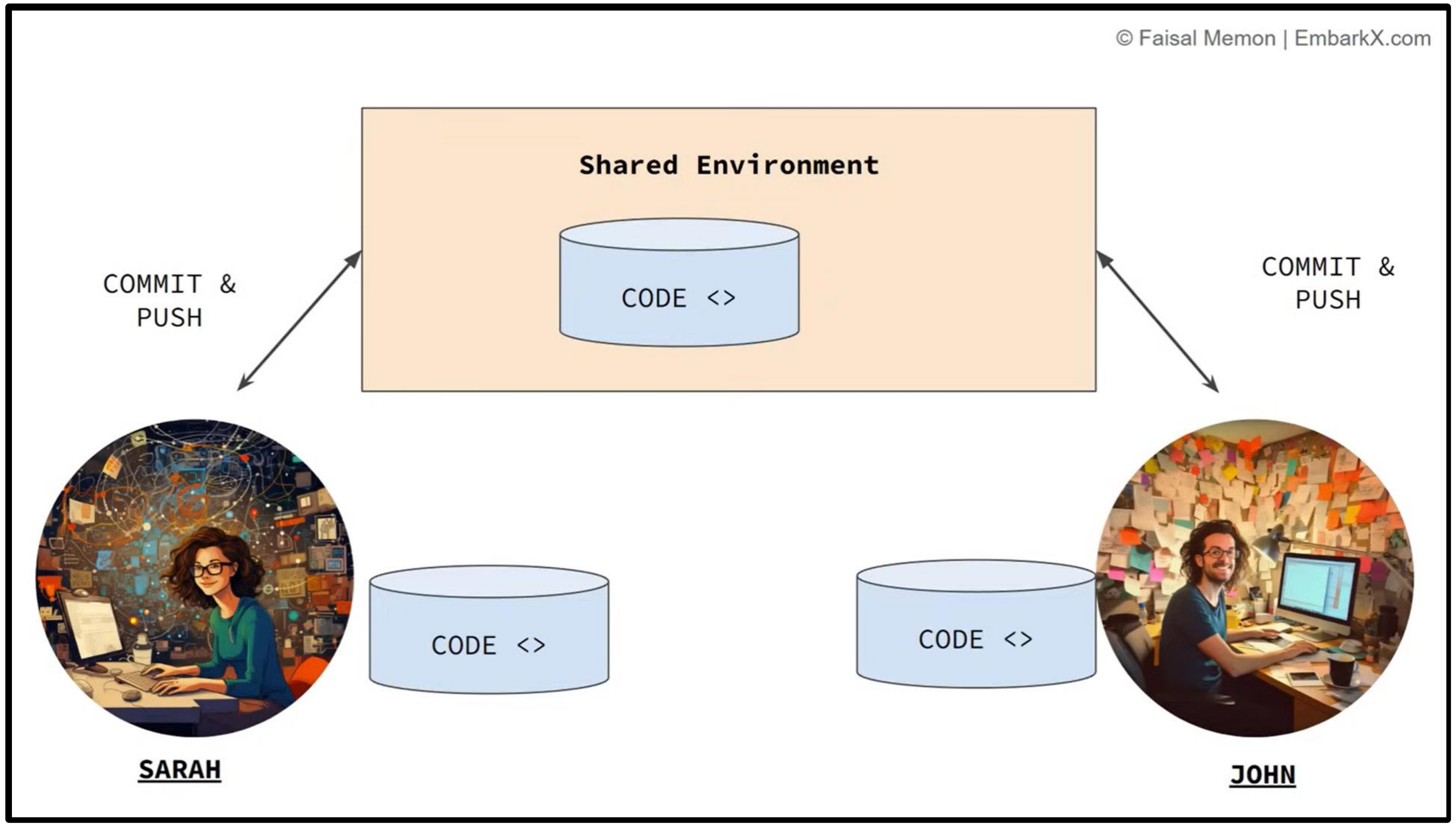


SARAH will use the Docker and has the container which contains all the required environment and application is running.

How?

- Now SARAH no need to worry about the conflicting versions or any other issues because of any versions conflicts.
- Now everybody in team of SARAH will upgrade to **Docker.**

How?



Now, isolated container Is shared between the team and this container has all the required item to run the application.

What **Docker** solves?

- Dependency Management.
- Compatibility issues.
- Environmental inconsistency.

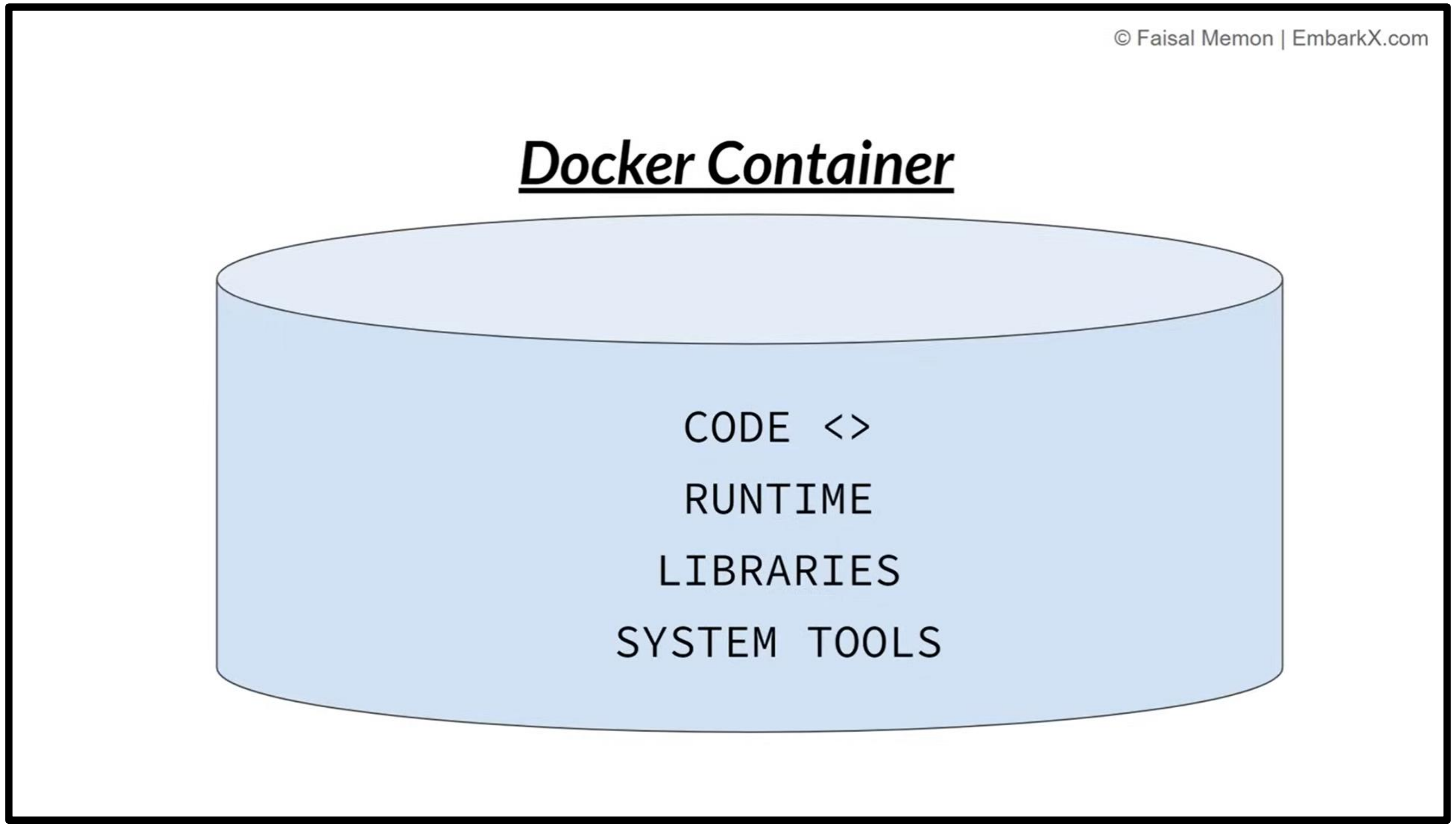
What is Docker?

Docker is an open-source platform that allows you to automate the deployment, scaling and management of applications using containerization.

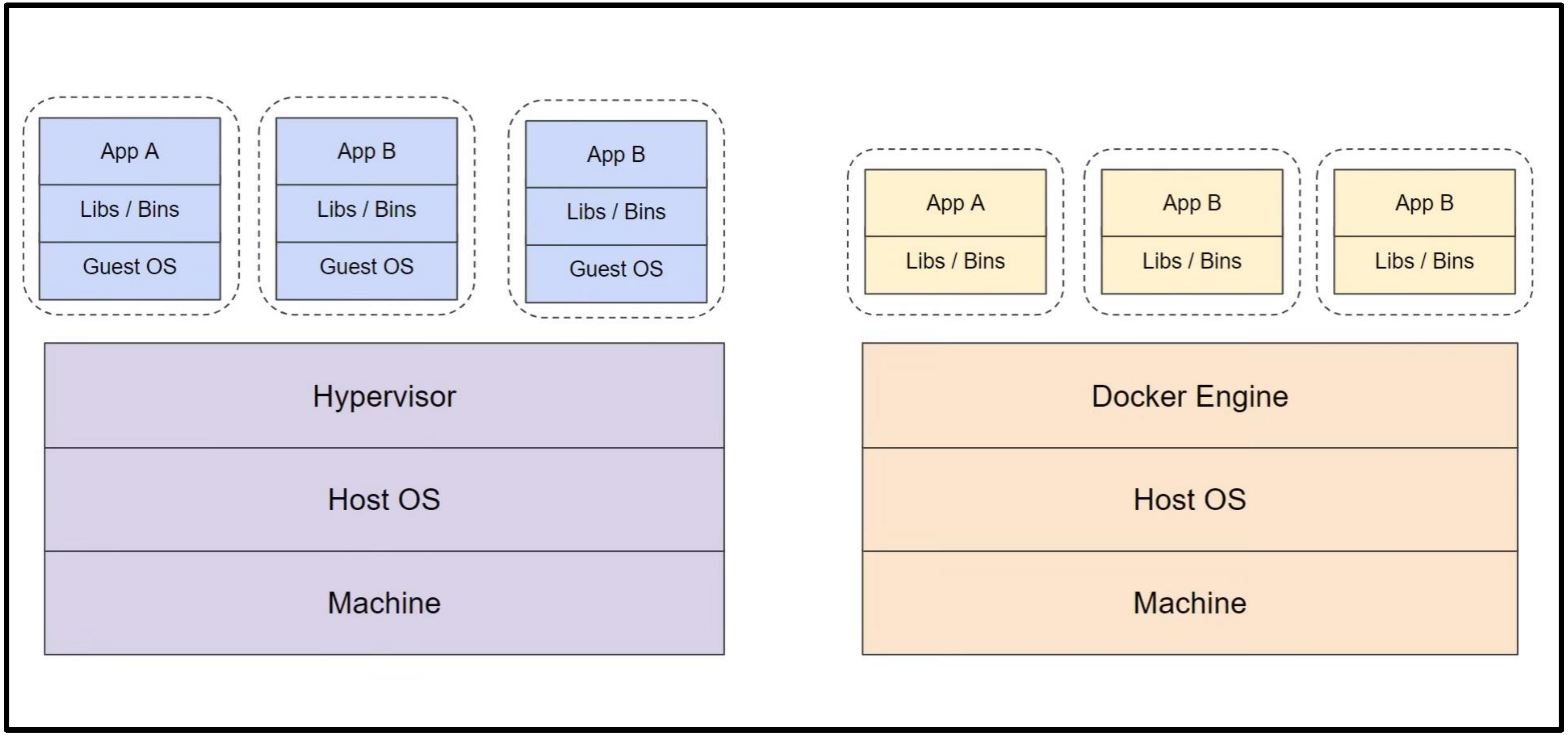
What is Containerization?

- lightweight virtualization technology
- package an application along with its dependencies that is needed for an application to run

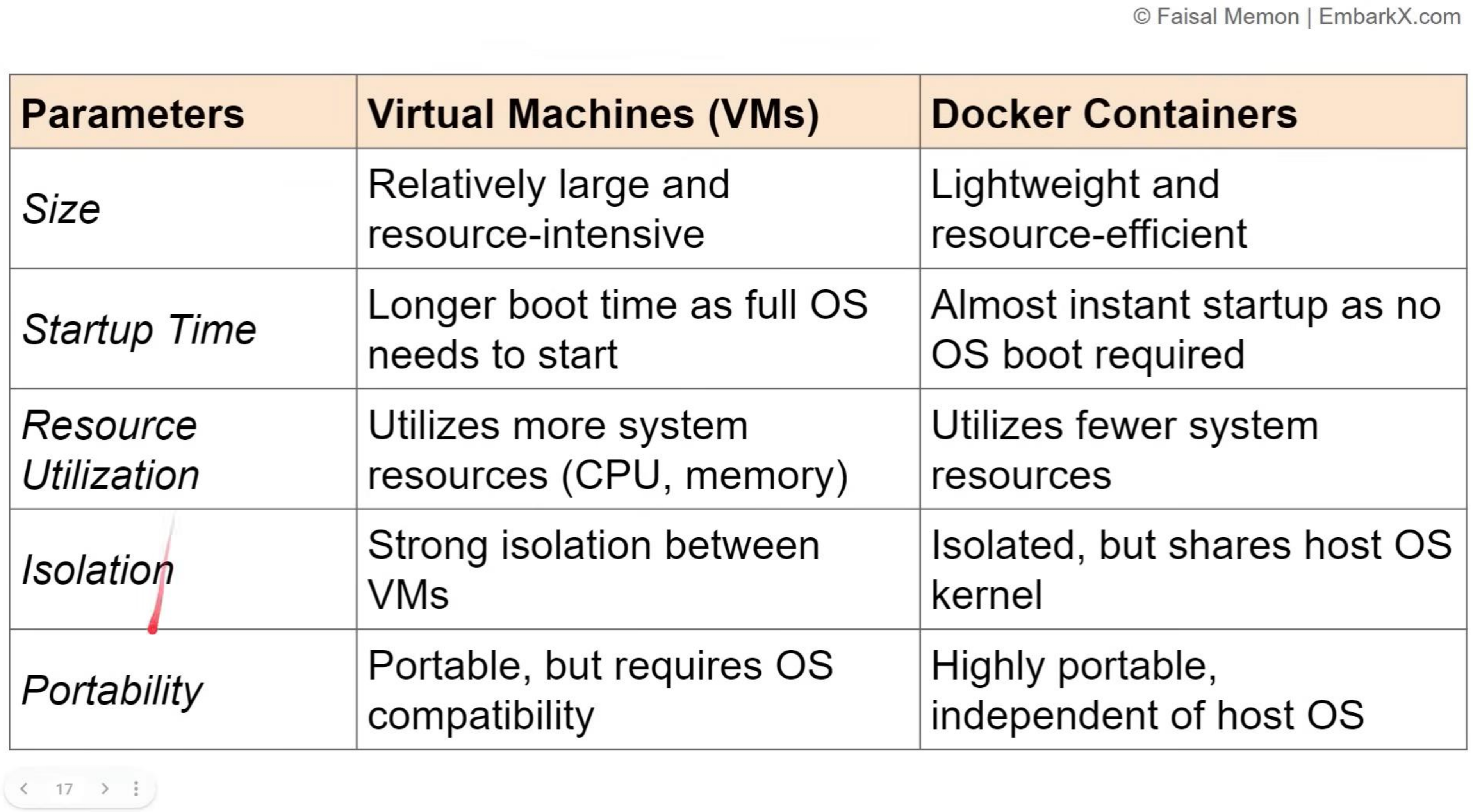
Docker Container



Docker over VM's



Docker over VM’s

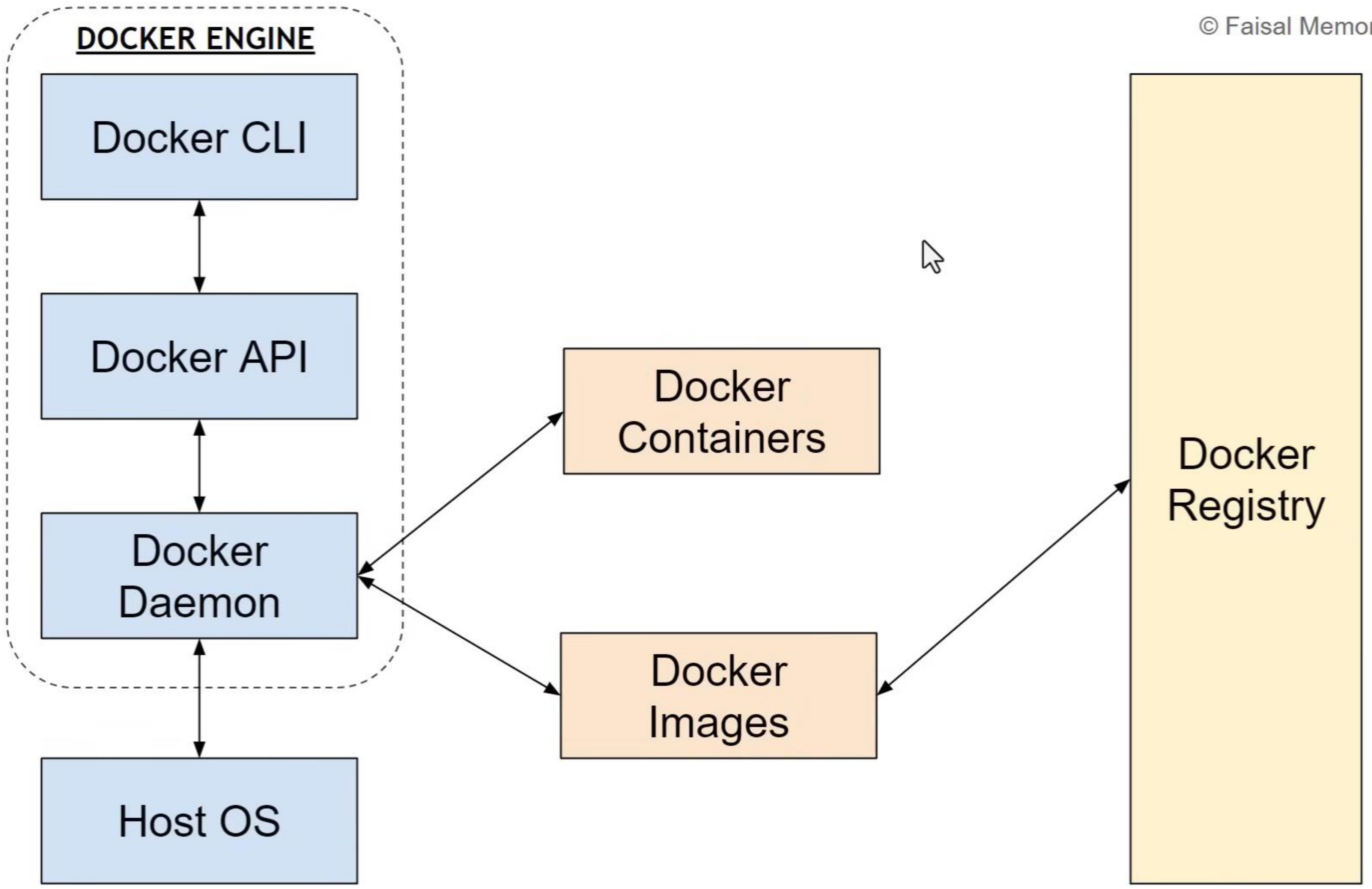


Parameters	**Virtual Machines (VMs)**	**Docker Containers**
Size	Relatively large and resource-intensive	Lightweight and resource-efficient
Startup Time	Longer boot time as full OS needs to start	Almost instant startup as no OS boot required
Resource Utilization	Utilizes more system resources (CPU, memory)	Utilizes fewer system resources
Isolation	Strong isolation between VMs	Isolated, but shares host OS kernel
Portability	Portable, but requires OS compatibility	Highly portable, independent of host OS

Docker over VM's

Parameters	**Virtual Machines (VMs)**	**Docker Containers**
Scalability	Scaling requires provisioning of new VMs	Easy to scale by creating more containers
Ecosystem	VM-specific tools and management frameworks	Docker ecosystem with extensive tooling
Development Workflow	Slower setup and provisioning process	Faster setup and dependency management
Deployment Efficiency	More overhead due to larger VM size	Efficient deployment with smaller container

Docker Architecture


DOCKER ENGINE
Docker CLI
Docker API
Docker Daemon
Host OS
Docker Containers
Docker Images
Docker Registry

Docker Engine

It contains:

1. Docker CLI
2. Docker API
3. Docker Daemon

Docker Engine

Docker CLI

- It's a client that allows users to interact with Docker
- It is used to communicate with the Docker.
- It will interact with Docker Daemon.

Docker Engine

Docker Daemon

- Runs on Host OS.
- Responsible for building docker images & managing containers.

Docker Images

- Blueprint for creating a container.
- From one image we can create multiple image.
- Templates that define the container and dependencies
- Image are base on which container is build.

Docker Containers

- Light weight
- Running instance of Docker image.

Dockerfile

- Instruction to build a Docker Image.

Docker Hub

- Registry that has the vast collection of Docker images.

Docker Registry

- It stores docker images.
- Images can be public or private.
- Repo of all the images.

Docker Registry

Docker Registry

- Store docker images with different version.

Importance of Docker Registry

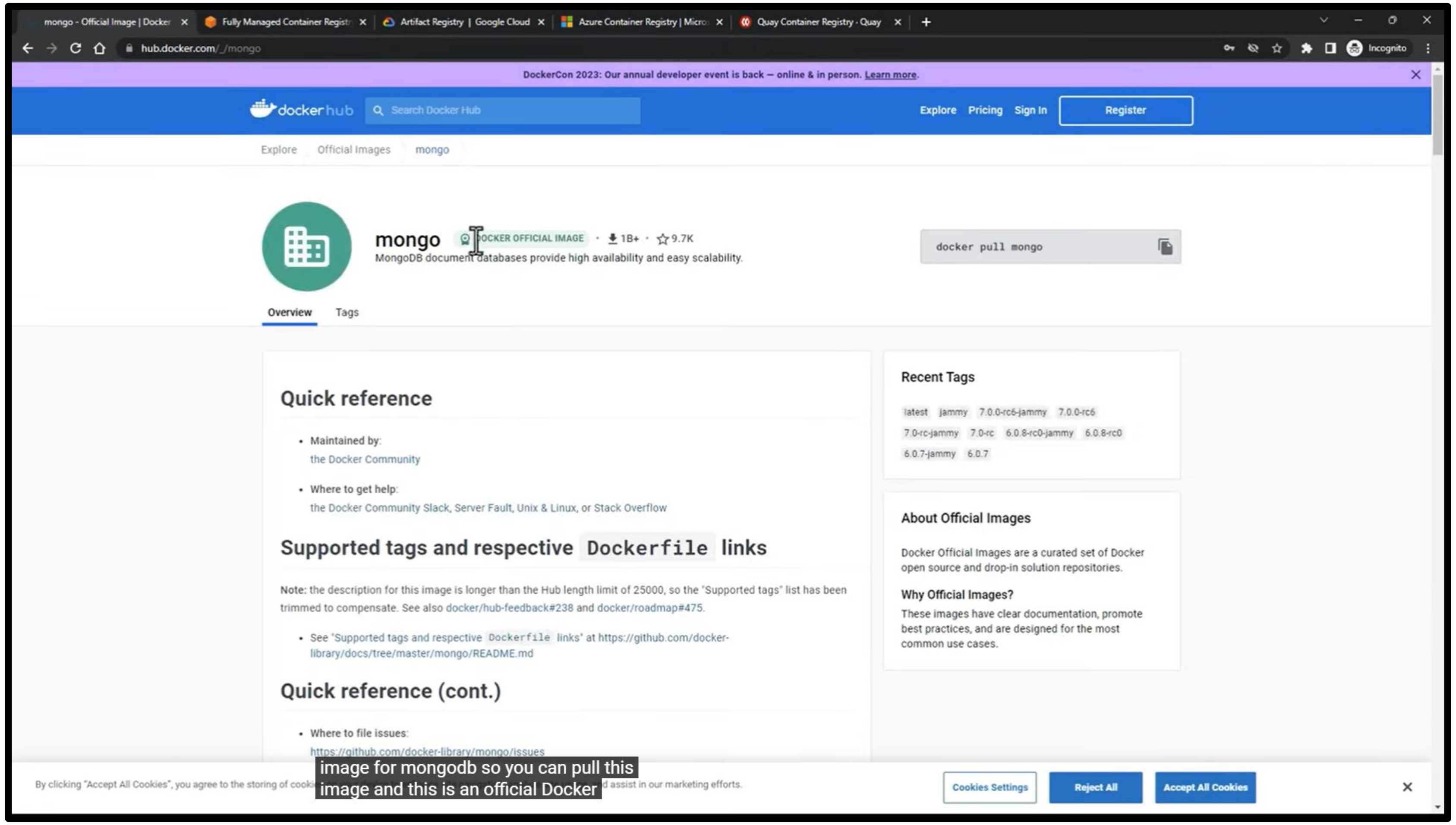
- Centralized Resource.
- Easy Versioning.
- Share your Docker Image.

Available Docker Registry

- https://hub.docker.com
    - Default registry used by Docker CLI and most of the docker tool.
- https://aws.amazon.com/ecr/
- https://cloud.google.com/artifact-registry
- https://azure.Microsoft.com/en-in/product/container-registry
- https://quay.io

Available Docker Registry

- Mongo image (https://hub.docker.com)



DockerCon 2023: Our annual developer event is back — online & in person. Learn more.

mongo DOCKER OFFICIAL IMAGE · 1B+ · 9.7K

MongoDB document databases provide high availability and easy scalability.

```
docker pull mongo
```

Quick reference

- Maintained by:
  the Docker Community
- Where to get help:
  the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

Supported tags and respective `Dockerfile` links

Note: the description for this image is longer than the Hub length limit of 25000, so the "Supported tags" list has been trimmed to compensate. See also docker/hub-feedback#238 and docker/roadmap#475.

- See "Supported tags and respective `Dockerfile` links" at https://github.com/docker-library/docs/tree/master/mongo/README.md

Quick reference (cont.)

- Where to file issues:
  https://github.com/docker-library/mongo/issues

Recent Tags

latest jammy 7.0.0-rc6-jammy 7.0.0-rc6 7.0-rc-jammy 7.0-rc 6.0.8-rc0-jammy 6.0.8-rc0 6.0.7-jammy 6.0.7

About Official Images

Docker Official Images are a curated set of Docker open source and drop-in solution repositories.

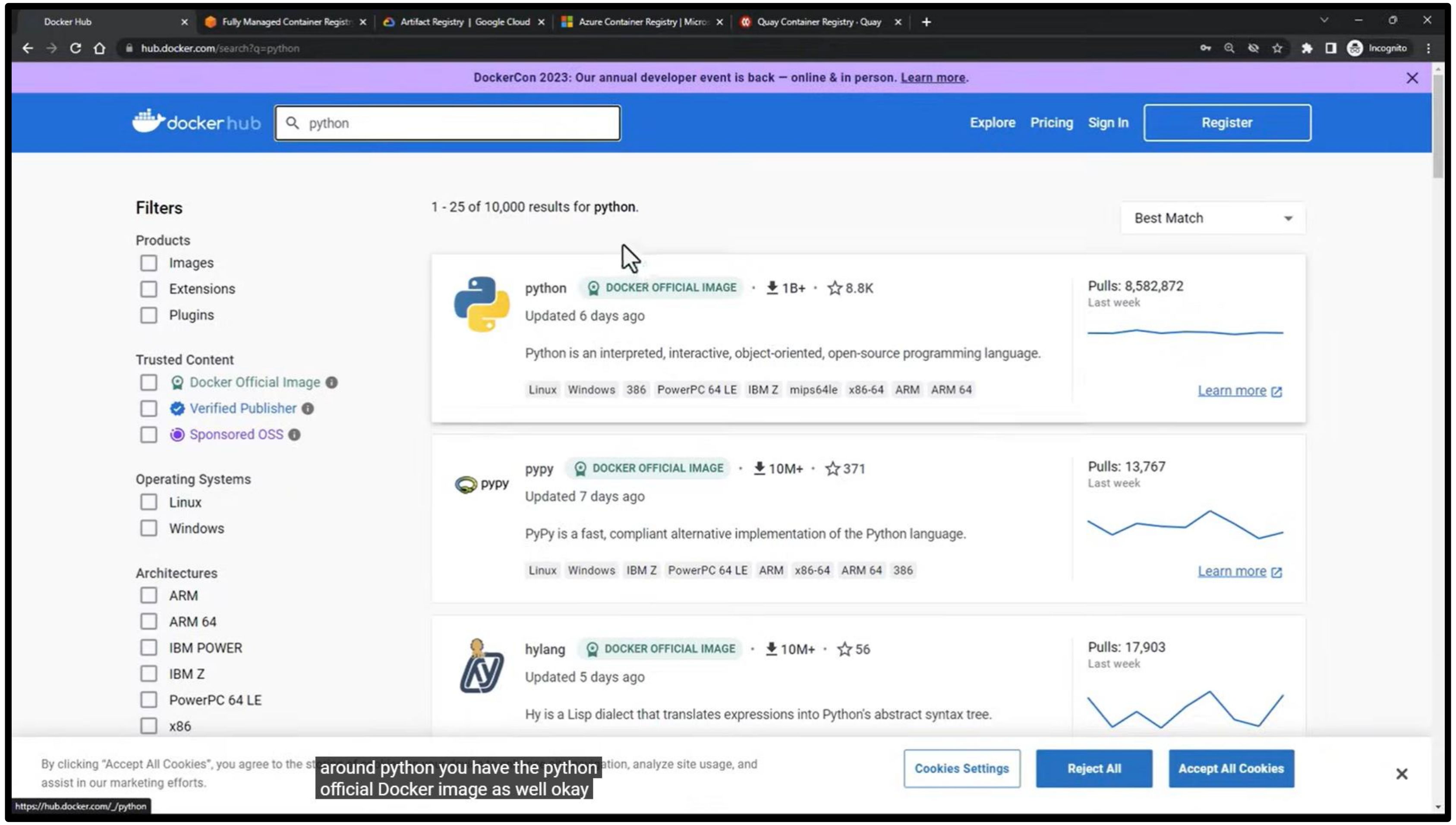
Why Official Images?

These images have clear documentation, promote best practices, and are designed for the most common use cases.

By clicking "Accept All Cookies", you agree to the storing of cooki image for mongodb so you can pull this image and this is an official Docker d assist in our marketing efforts.

Available Docker Registry

- python image (https://hub.docker.com)



Docker Hub | Fully Managed Container Regist | Artifact Registry | Google Cloud | Azure Container Registry | Micro | Quay Container Registry · Quay

hub.docker.com/search?q=python

DockerCon 2023: Our annual developer event is back — online & in person. Learn more.

docker hub | python | Explore | Pricing | Sign In | Register

Filters

Products
- Images
- Extensions
- Plugins

Trusted Content
- Docker Official Image
- Verified Publisher
- Sponsored OSS

Operating Systems
- Linux
- Windows

Architectures
- ARM
- ARM 64
- IBM POWER
- IBM Z
- PowerPC 64 LE
- x86

1 - 25 of 10,000 results for **python**.

Best Match

python DOCKER OFFICIAL IMAGE · 1B+ · 8.8K
Updated 6 days ago
Python is an interpreted, interactive, object-oriented, open-source programming language.
Linux Windows 386 PowerPC 64 LE IBM Z mips64le x86-64 ARM ARM 64
Pulls: 8,582,872
Last week
Learn more

pypy DOCKER OFFICIAL IMAGE · 10M+ · 371
Updated 7 days ago
PyPy is a fast, compliant alternative implementation of the Python language.
Linux Windows IBM Z PowerPC 64 LE ARM x86-64 ARM 64 386
Pulls: 13,767
Last week
Learn more

hylang DOCKER OFFICIAL IMAGE · 10M+ · 56
Updated 5 days ago
Hy is a Lisp dialect that translates expressions into Python's abstract syntax tree.
Pulls: 17,903
Last week

By clicking "Accept All Cookies", you agree to the st[...]ation, analyze site usage, and assist in our marketing efforts.

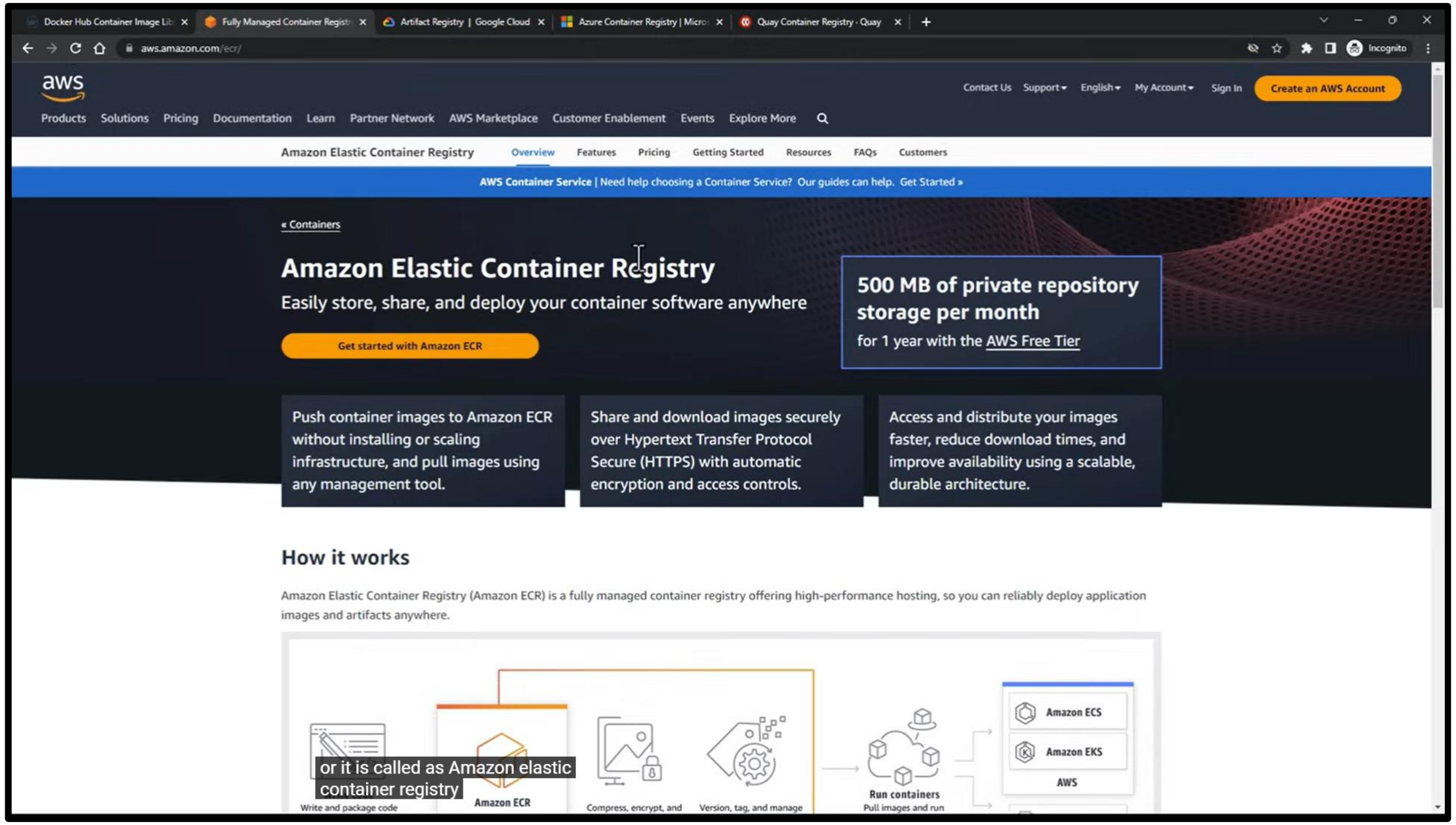
Cookies Settings | Reject All | Accept All Cookies

around python you have the python official Docker image as well okay

https://hub.docker.com/_/python

Available Docker Registry

- (https://aws.amazon.com/ecr/)



AWS Container Service | Need help choosing a Container Service? Our guides can help. Get Started »

Amazon Elastic Container Registry

Easily store, share, and deploy your container software anywhere

500 MB of private repository storage per month

for 1 year with the AWS Free Tier

Push container images to Amazon ECR without installing or scaling infrastructure, and pull images using any management tool.

Share and download images securely over Hypertext Transfer Protocol Secure (HTTPS) with automatic encryption and access controls.

Access and distribute your images faster, reduce download times, and improve availability using a scalable, durable architecture.

How it works

Amazon Elastic Container Registry (Amazon ECR) is a fully managed container registry offering high-performance hosting, so you can reliably deploy application images and artifacts anywhere.

or it is called as Amazon elastic container registry

Steps to create container from Docker Image

How to create container from Docker Hub?

- Open browser and open Docker Hub after that search images what you have to download and make the container

How to create container from Docker Hub?

- There are 2 options:
  - Pull image from Docker hub and then create container.
  - Direct use **Docker run** command to create container.

Steps to create container from Docker Hub?

- To create container with container name and port

```
# docker run -it -name bhuvi -p 80:80 nginx:latest
```

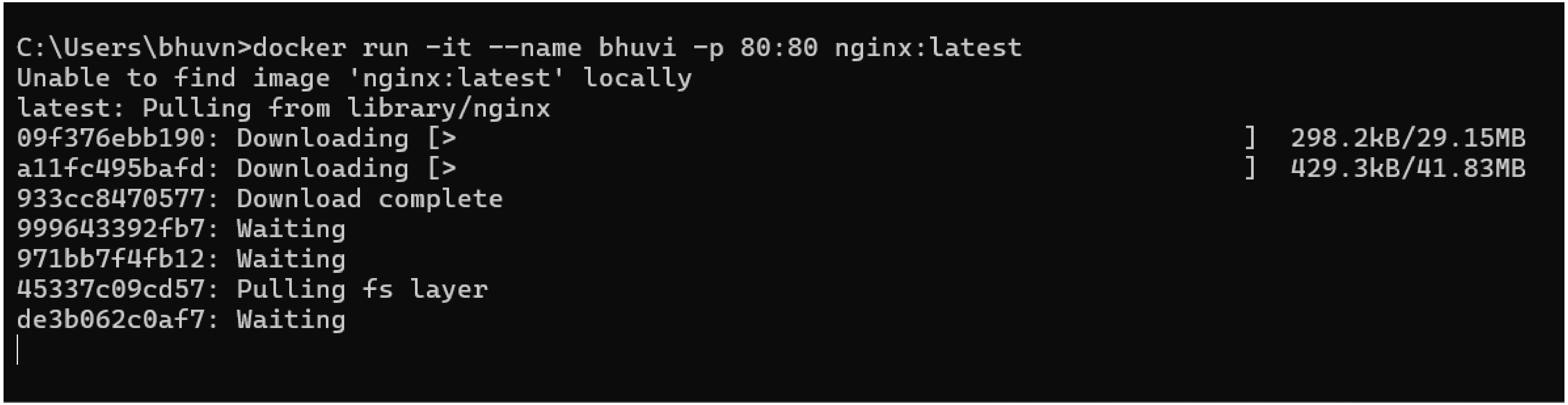
Name of container

Port for outside access

Image name on docker hub

Steps to create container from Docker Hub?

- To create container with container name and port

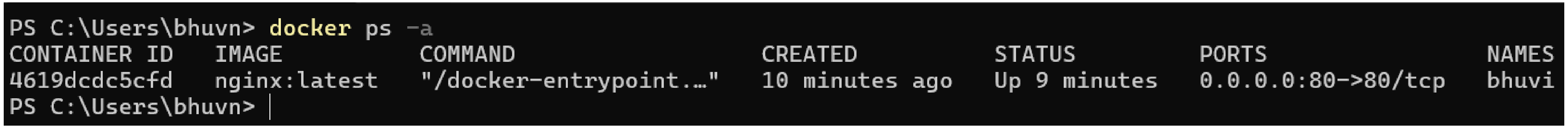


```
C:\Users\bhuvn>docker run -it --name bhuvi -p 80:80 nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
09f376ebb190: Downloading [>                                                  ]  298.2kB/29.15MB
a11fc495bafd: Downloading [>                                                  ]  429.3kB/41.83MB
933cc8470577: Download complete
999643392fb7: Waiting
971bb7f4fb12: Waiting
45337c09cd57: Pulling fs layer
de3b062c0af7: Waiting
```

Steps to create container from Docker Hub?

- Check that container is available or not

  # docker ps -a

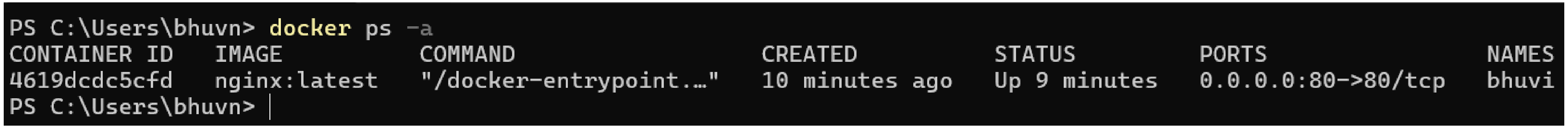


```
PS C:\Users\bhuvn> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS         PORTS                NAMES
4619dcdc5cfd   nginx:latest   "/docker-entrypoint.…"   10 minutes ago   Up 9 minutes   0.0.0.0:80->80/tcp   bhuvi
PS C:\Users\bhuvn>
```

Steps to create container from Docker Hub?

- To start container

  # docker run bhuvi

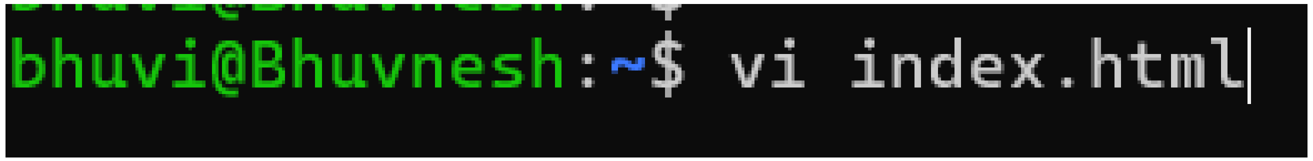


```
PS C:\Users\bhuvn> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS         PORTS                NAMES
4619dcdc5cfd   nginx:latest   "/docker-entrypoint.…"   10 minutes ago   Up 9 minutes   0.0.0.0:80->80/tcp   bhuvi
PS C:\Users\bhuvn>
```

Steps to create container from Docker Hub?

- Create index.html file

# vi index.html

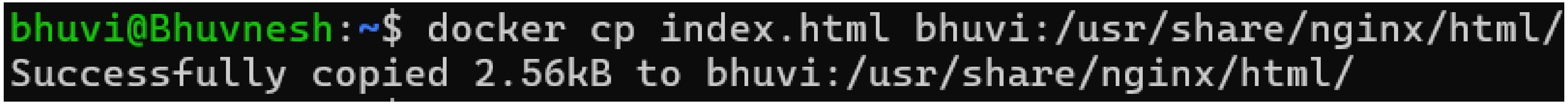


```
bhuvi@Bhuvnesh:~$ vi index.html
```

Steps to create container from Docker Hub?

- Copy index.html and paste it in nginx directory

# docker cp index.html bhuvi:/usr/share/nginx/html/

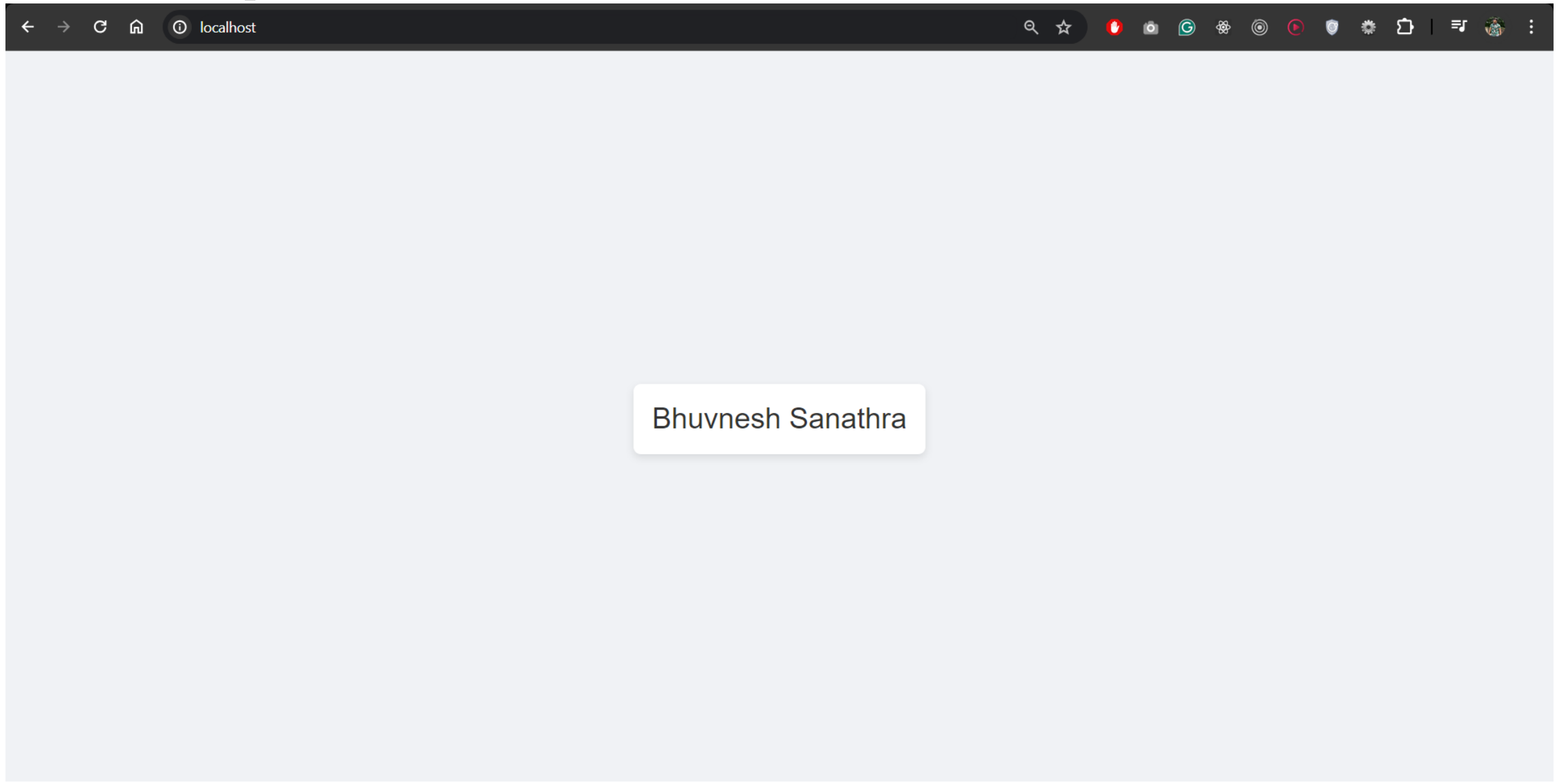


```
bhuvi@Bhuvnesh:~$ docker cp index.html bhuvi:/usr/share/nginx/html/
Successfully copied 2.56kB to bhuvi:/usr/share/nginx/html/
```

Steps to create container from Docker Hub?

- Run in localhost on port 80

  localhost:80



Using images from Docker Hub

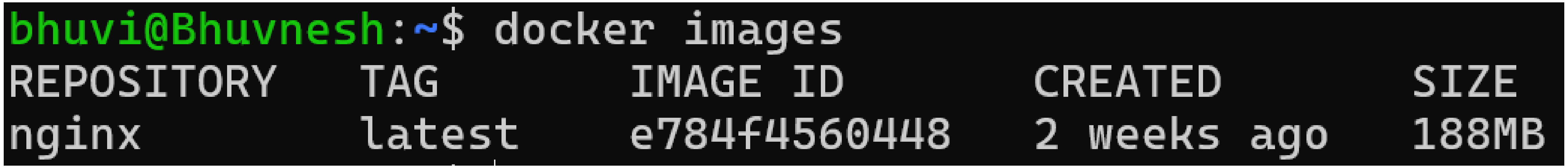
How to Use of existing images from Docker Hub ?

- I took the example of **MongoDB** image from docker hub

Steps to Use of existing images from Docker Hub ?

- Check the docker images

```
# docker images
```

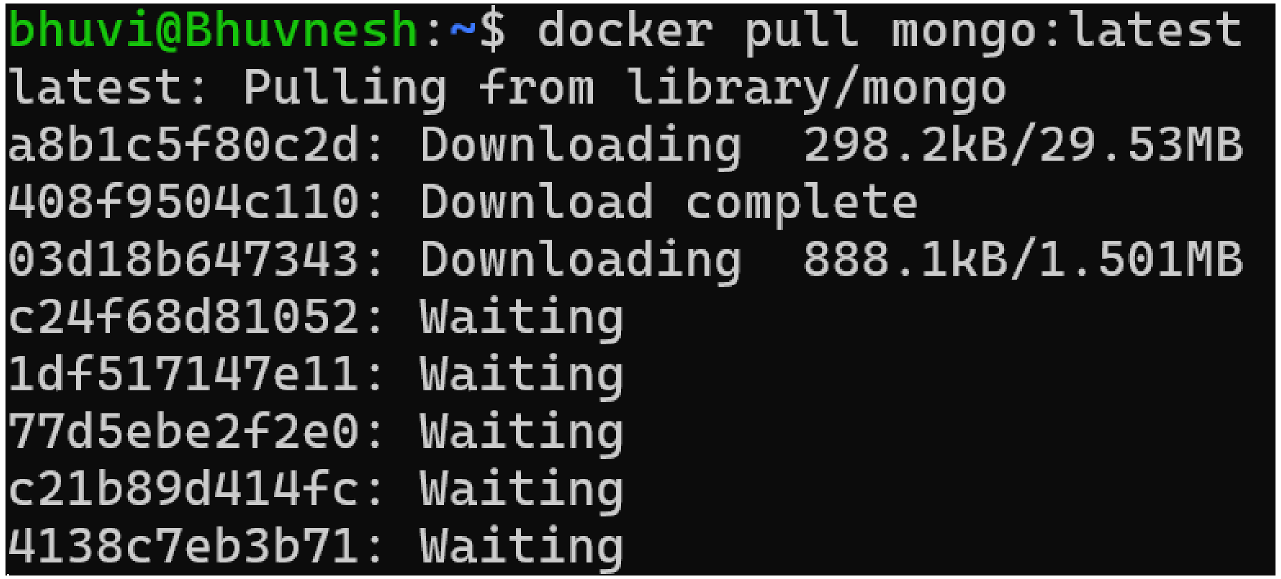


```
bhuvi@Bhuvnesh:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
nginx         latest    e784f4560448   2 weeks ago   188MB
```

Steps to Use of existing images from Docker Hub ?

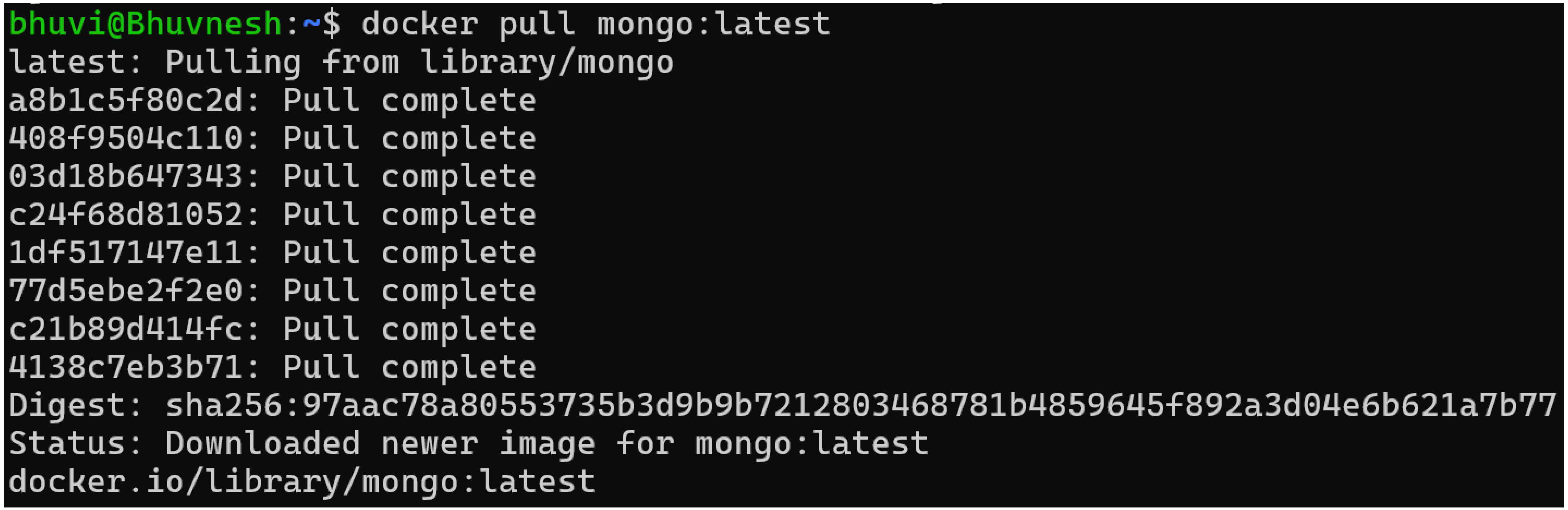
- Pull the mongoDB image

```
# docker pull mongo:latest
```



```
bhuvi@Bhuvnesh:~$ docker pull mongo:latest
latest: Pulling from library/mongo
a8b1c5f80c2d: Downloading  298.2kB/29.53MB
408f9504c110: Download complete
03d18b647343: Downloading  888.1kB/1.501MB
c24f68d81052: Waiting
1df517147e11: Waiting
77d5ebe2f2e0: Waiting
c21b89d414fc: Waiting
4138c7eb3b71: Waiting
```

Steps to Use of existing images from Docker Hub ?

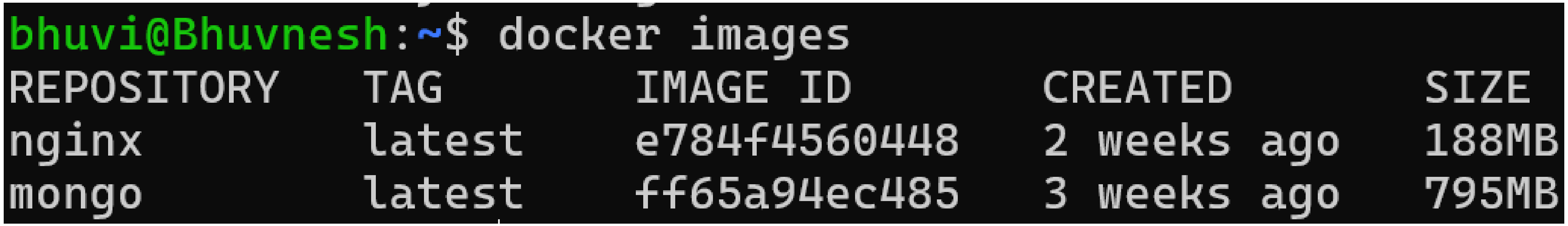


```
bhuvi@Bhuvnesh:~$ docker pull mongo:latest
latest: Pulling from library/mongo
a8b1c5f80c2d: Pull complete
408f9504c110: Pull complete
03d18b647343: Pull complete
c24f68d81052: Pull complete
1df517147e11: Pull complete
77d5ebe2f2e0: Pull complete
c21b89d414fc: Pull complete
4138c7eb3b71: Pull complete
Digest: sha256:97aac78a80553735b3d9b9b7212803468781b4859645f892a3d04e6b621a7b77
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
```

Steps to Use of existing images from Docker Hub ?

- Check the docker images

```
# docker images
```



```
bhuvi@Bhuvnesh:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    e784f4560448   2 weeks ago    188MB
mongo         latest    ff65a94ec485   3 weeks ago    795MB
```

Steps to Use of existing images from Docker Hub ?

- Make directory inside root folder

```
# mkdir mongoDB-bhuvi
```

- Change directory

```
# cd mongoDB-bhuvi
```

Steps to Use of existing images from Docker Hub ?

- Run the image inside the folder

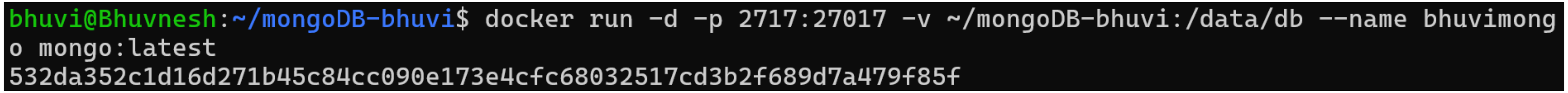
```
   # docker run –d –p 2717:27017 –v ~/mongoDB-
bhuvi:/data/db –name bhuvimongo mongo:latest

-d -> run in background
-p -> bind the port
-v -> Volume
--name -> name of the image
```

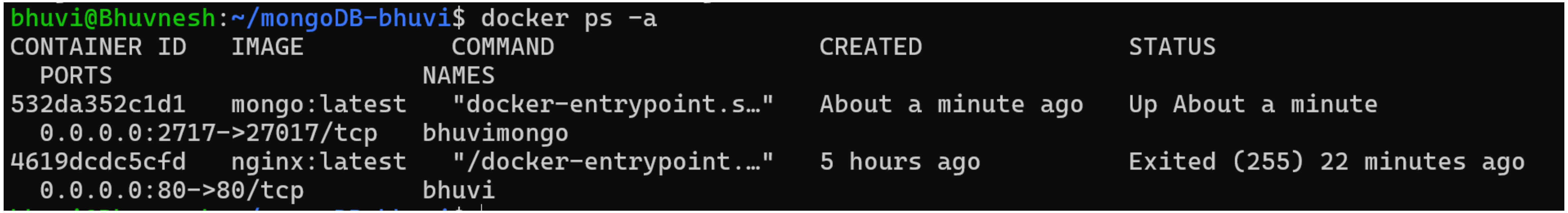
Steps to Use of existing images from Docker Hub ?

- Container created check the container

```
# docker ps -a
```



```
bhuvi@Bhuvnesh:~/mongoDB-bhuvi$ docker run -d -p 2717:27017 -v ~/mongoDB-bhuvi:/data/db --name bhuvimong
o mongo:latest
532da352c1d16d271b45c84cc090e173e4cfc68032517cd3b2f689d7a479f85f
```

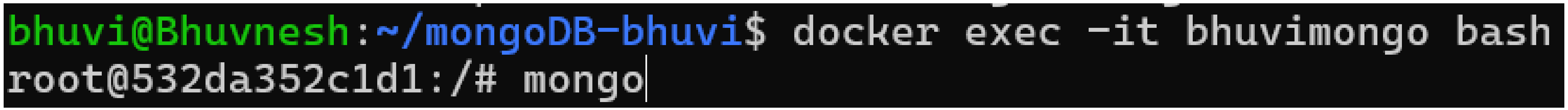


```
bhuvi@Bhuvnesh:~/mongoDB-bhuvi$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS
  PORTS                     NAMES
532da352c1d1   mongo:latest   "docker-entrypoint.s…"   About a minute ago   Up About a minute
  0.0.0.0:2717->27017/tcp   bhuvimongo
4619dcdc5cfd   nginx:latest   "/docker-entrypoint.…"   5 hours ago          Exited (255) 22 minutes ago
  0.0.0.0:80->80/tcp        bhuvi
```

Steps to Use of existing images from Docker Hub ?

- Go inside container

```
# docker exec -it bhuvimongo bash
```

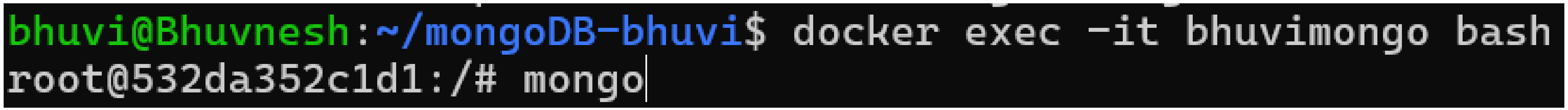


```
bhuvi@Bhuvnesh:~/mongoDB-bhuvi$ docker exec -it bhuvimongo bash
root@532da352c1d1:/# mongo
```

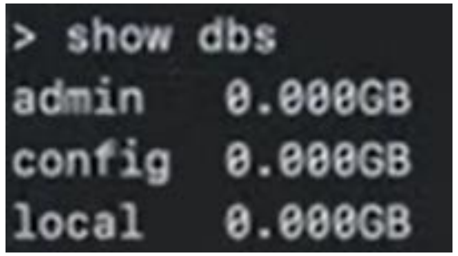
Steps to Use of existing images from Docker Hub ?

- Go inside container

```
# docker exec -it bhuvimongo bash
```

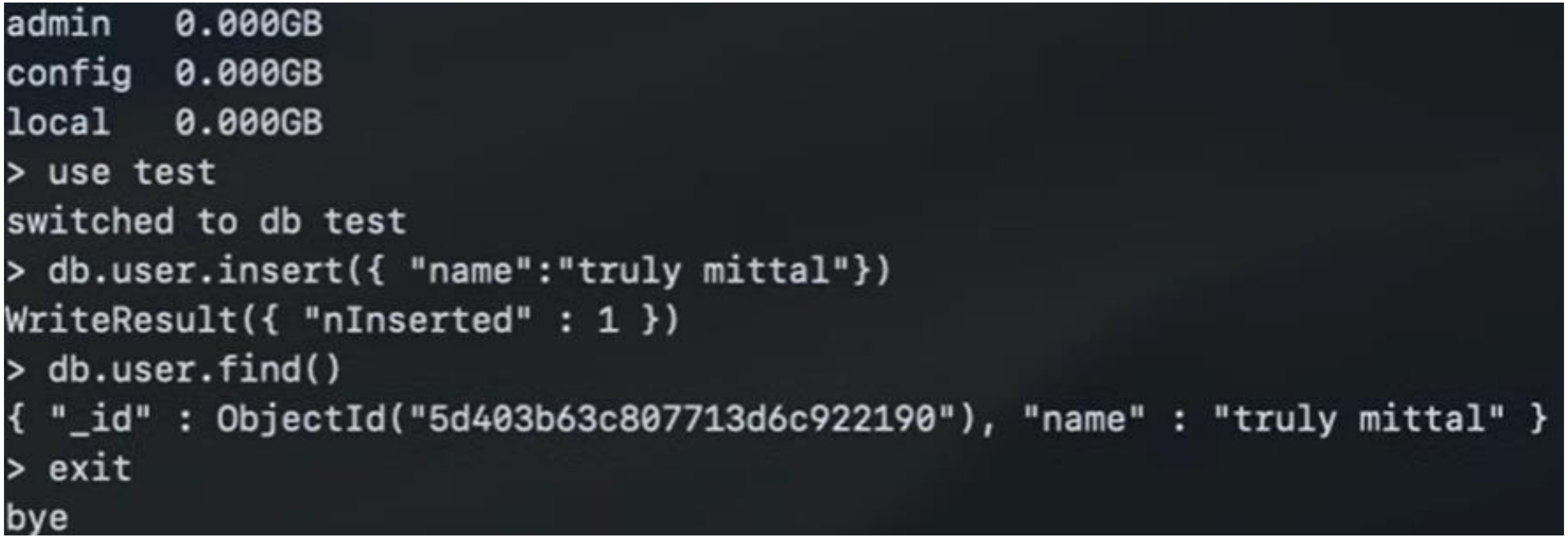


```
bhuvi@Bhuvnesh:~/mongoDB-bhuvi$ docker exec -it bhuvimongo bash
root@532da352c1d1:/# mongo
```



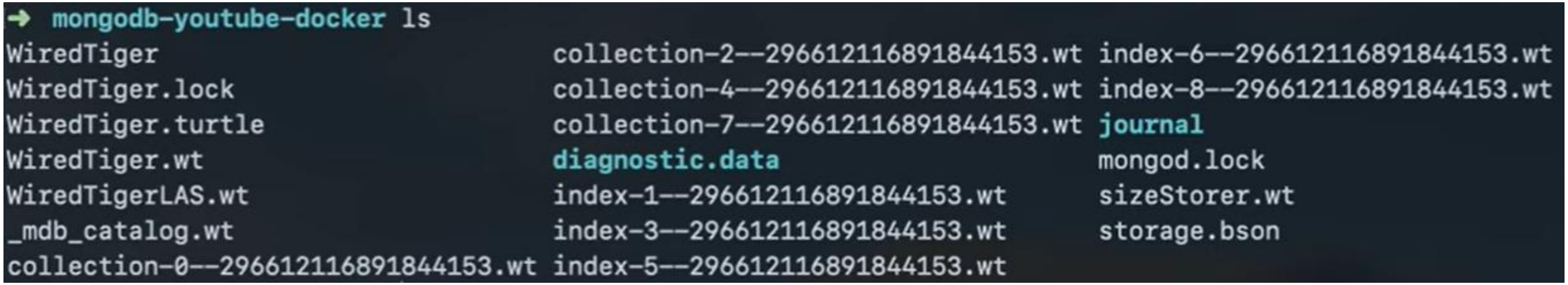
```
> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
```

Steps to Use of existing images from Docker Hub ?



```
admin   0.000GB
config  0.000GB
local   0.000GB
> use test
switched to db test
> db.user.insert({ "name":"truly mittal"})
WriteResult({ "nInserted" : 1 })
> db.user.find()
{ "_id" : ObjectId("5d403b63c807713d6c922190"), "name" : "truly mittal" }
> exit
bye
```

Steps to Use of existing images from Docker Hub ?



```
➜  mongodb-youtube-docker ls
WiredTiger                          collection-2--296612116891844153.wt index-6--296612116891844153.wt
WiredTiger.lock                     collection-4--296612116891844153.wt index-8--296612116891844153.wt
WiredTiger.turtle                   collection-7--296612116891844153.wt journal
WiredTiger.wt                       diagnostic.data                     mongod.lock
WiredTigerLAS.wt                    index-1--296612116891844153.wt      sizeStorer.wt
_mdb_catalog.wt                     index-3--296612116891844153.wt      storage.bson
collection-0--296612116891844153.wt index-5--296612116891844153.wt
```

Thank You