INTRODUCTION TO

& Its Applications in Big Data

AGENDA

What is Docker?

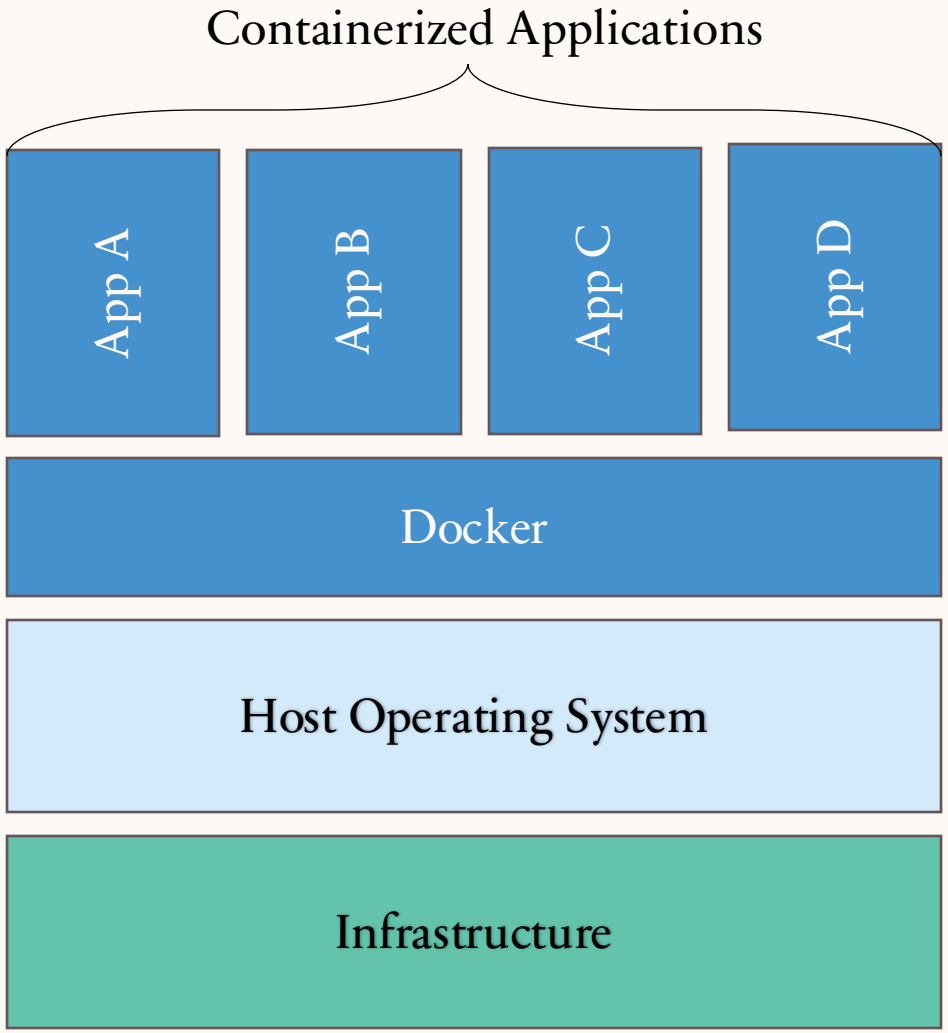
Docker Architecture

Why use Docker?

Hands-On

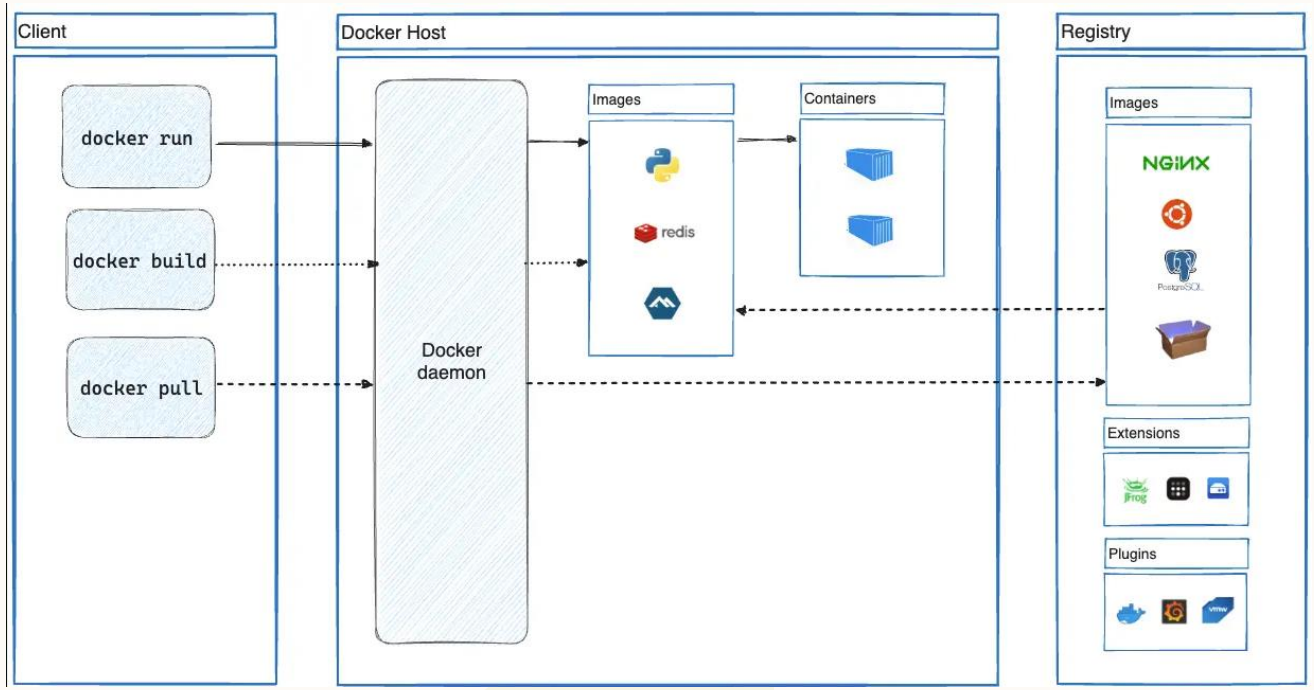
WHAT IS docker?

WHAT IS docker?



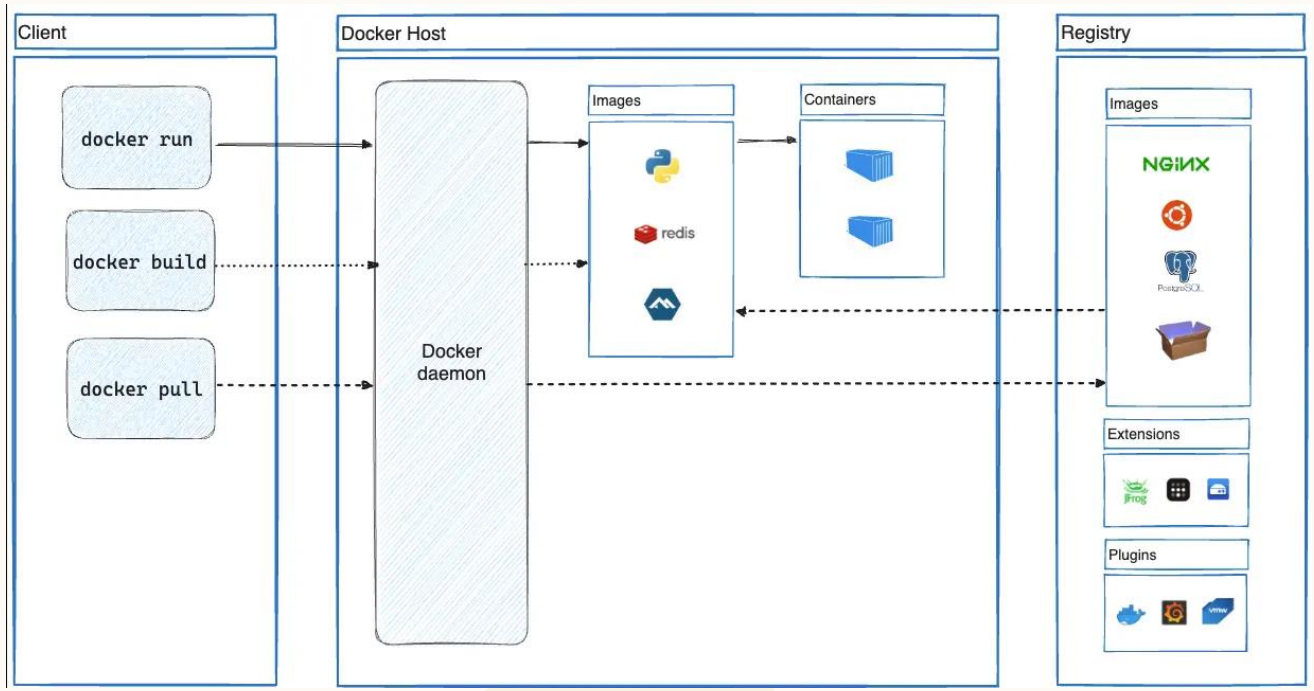
DOCKER ARCHITECTURE

How is it different from Virtual Machines?



DOCKER ARCHITECTURE

- **Docker Engine**: Manages the containers.
- **Docker Client**: The interface you use to communicate with Docker.
- **Docker Daemon**: The background process that runs containers.
- **Docker Images**: Blueprints for containers.
- **Docker Containers**: Running instances of Docker images.

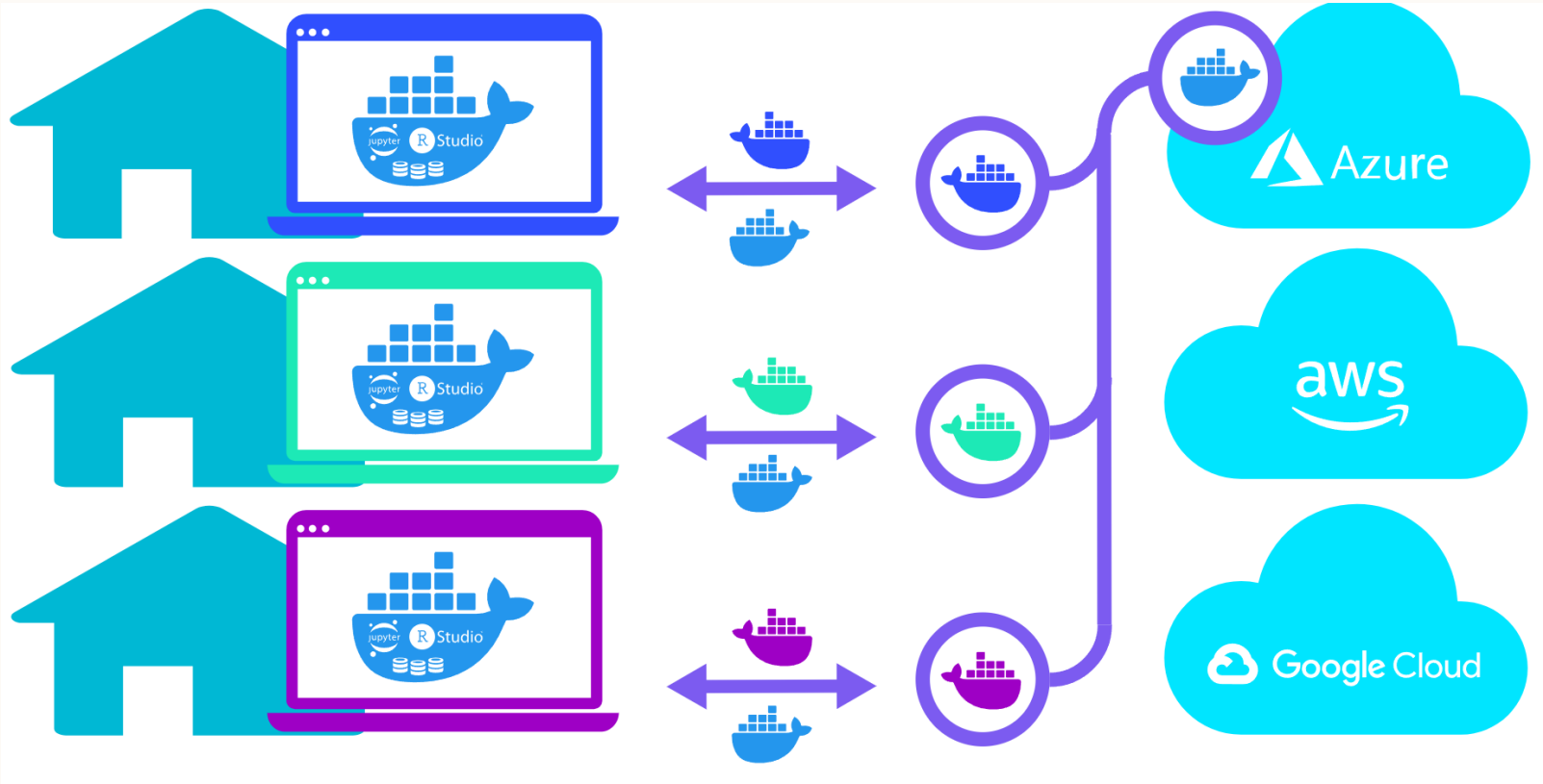


WHY USE DOCKER?

- **Portability**: Run applications anywhere, regardless of the host system.
- **Efficiency**: Uses fewer resources compared to VMs.
- **Consistency**: The same behavior across development, testing, and production environments.
- **Scalability**: Easily scalable using orchestration tools like Kubernetes.

WHY USE DOCKER?

In Big Data Analytics



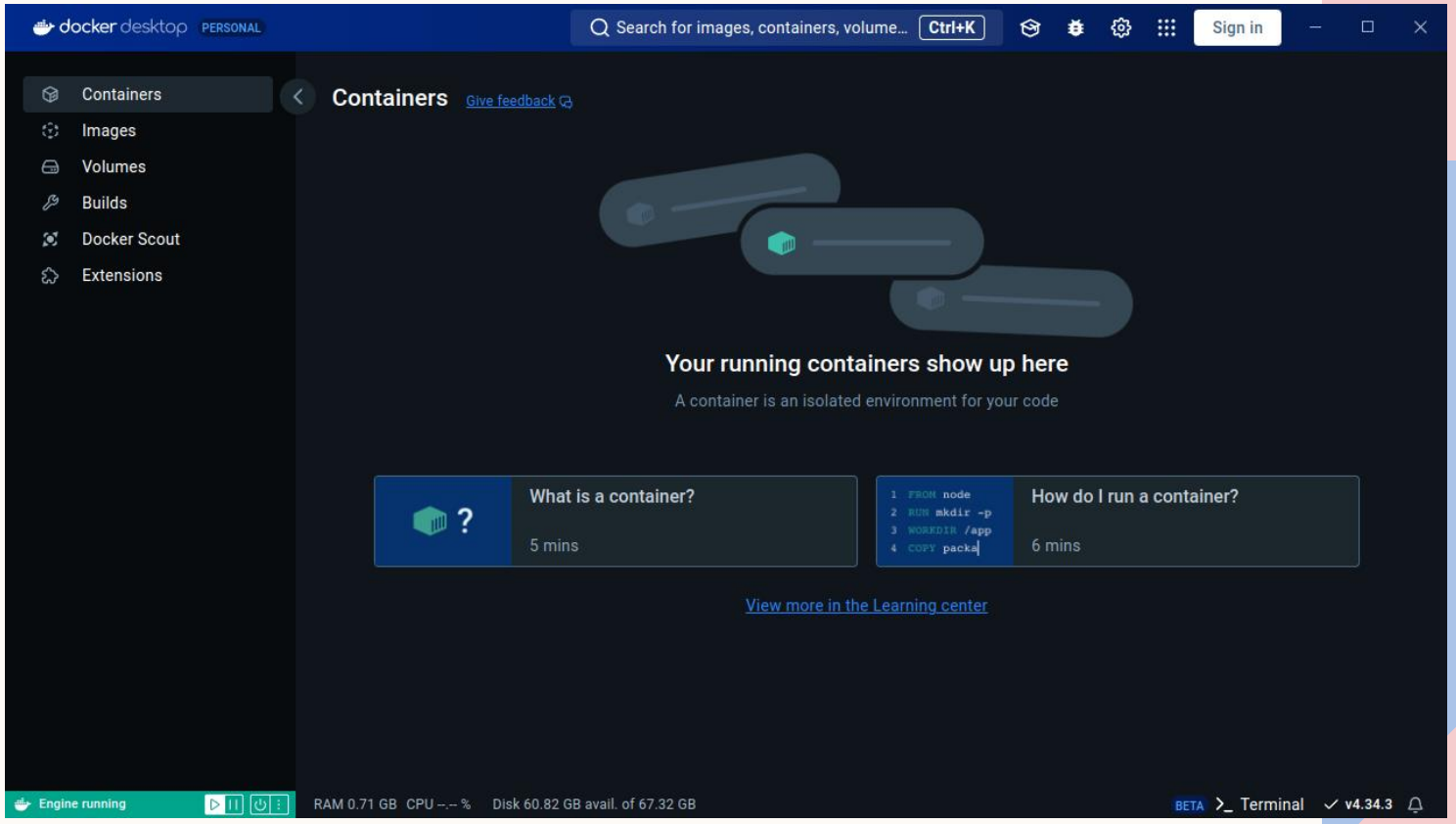
HANDS-ON

Breakdown of Steps:

- **Step 1**: Create a Dockerfile that defines the app's environment.
- **Step 2**: Build a Docker image using docker build.
- **Step 3**: Run the image as a container with docker run.

HANDS-ON

Docker Desktop/ Docker Engine



HANDS-ON

Simple App using FastAPI

app.py 1

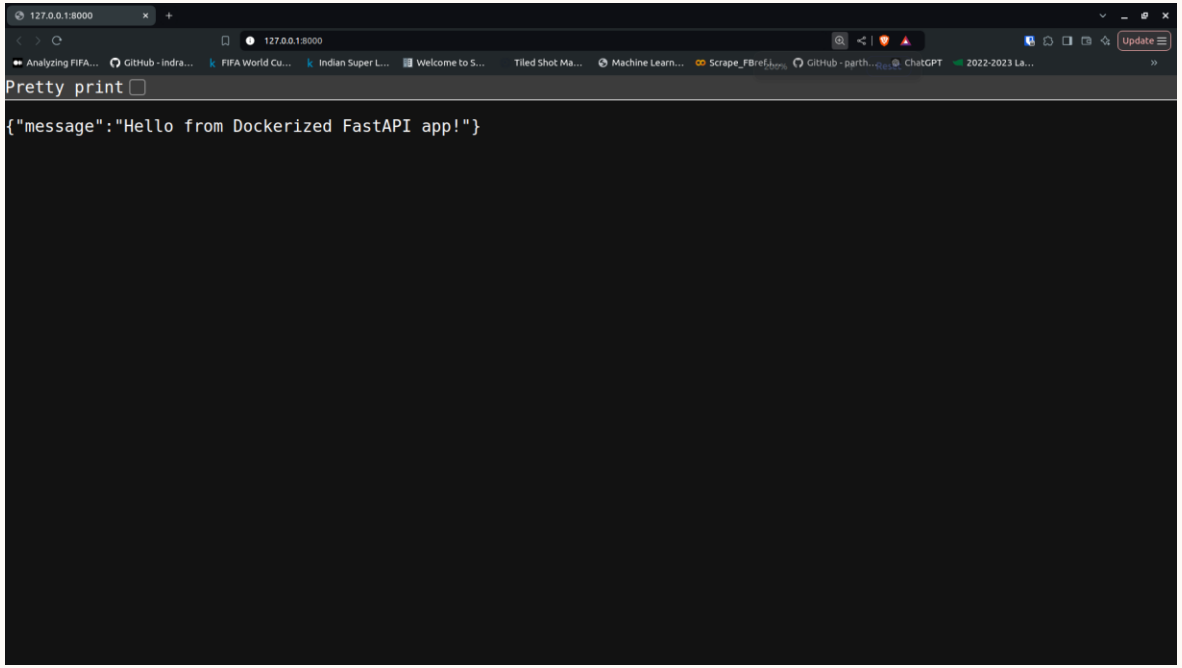
app.py > root

```python
# app.py
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello from Dockerized FastAPI app!"}
```

HANDS-ON

To run the App, regularly we use:

```
$ uvicorn app:app --reload --port 8000
```



HANDS-ON

•**Step 1**: Create a Dockerfile that defines the app's environment.

app.py 1 ● dockerfile ●

dockerfile > ...

```
FROM python:3.8-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

EXPOSE 80

CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "80"]
```

- Use an official Python runtime as a parent image
- Set the working directory inside the container
- Copy the current directory contents into the container at /app
- Install any needed packages specified in requirements.txt
- Make port 80 available to the world outside this container
- Run app.py when the container launches with uvicorn

HANDS-ON

- **Step 2**: Build a Docker image using docker build.

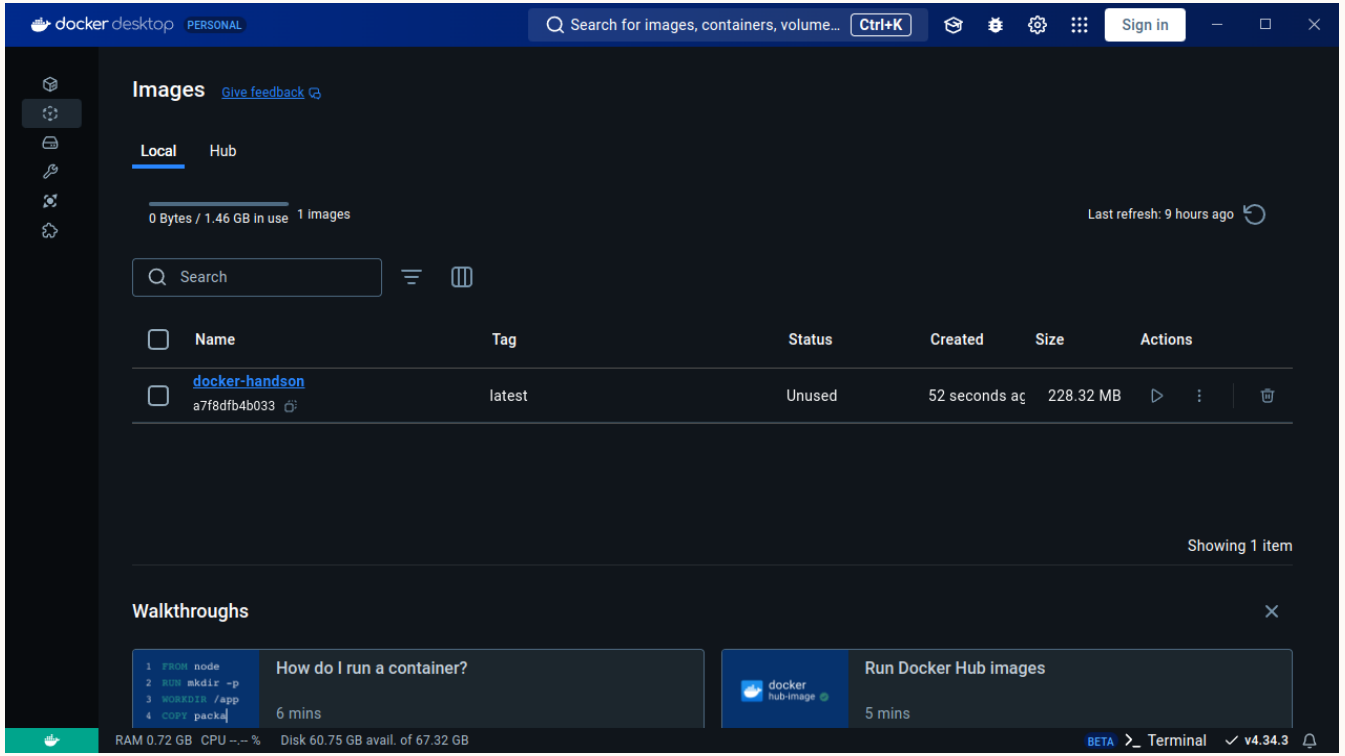
```
n$ docker build -t docker-handson .
```

```
[+] Building 25.8s (9/9) FINISHED
 => [internal] load build definition from dockerfile
 => => transferring dockerfile: 540B
 => [internal] load metadata for docker.io/library/python:3.8-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
 => => resolve docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
 => [internal] load build context
 => => transferring context: 722B
 => CACHED [2/4] WORKDIR /app
 => [3/4] COPY . /app
 => [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:e546c581220429272c49fee9252a1dee7f90f137640a250515f0003d63974b3f
 => => exporting config sha256:43a116e83ac5c661b9c9fb83ccb8a8ab839baae6fdae213cd0b15434252e58e5
 => => exporting attestation manifest sha256:811f47609d88ee03cffd68e5fba664a82b664f0500ed4bbe9f8f66252c810bb5
 => => exporting manifest list sha256:a7f8dfb4b0338087150c393de8fb28a60dba3e4774d895f49e5f525083448fbc
 => => naming to docker.io/library/docker-handson:latest
 => => unpacking to docker.io/library/docker-handson:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ezi00o6bx0z5m9xk70ucn7oq7

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

HANDS-ON

- **Step 2**: Build a Docker image using docker build.

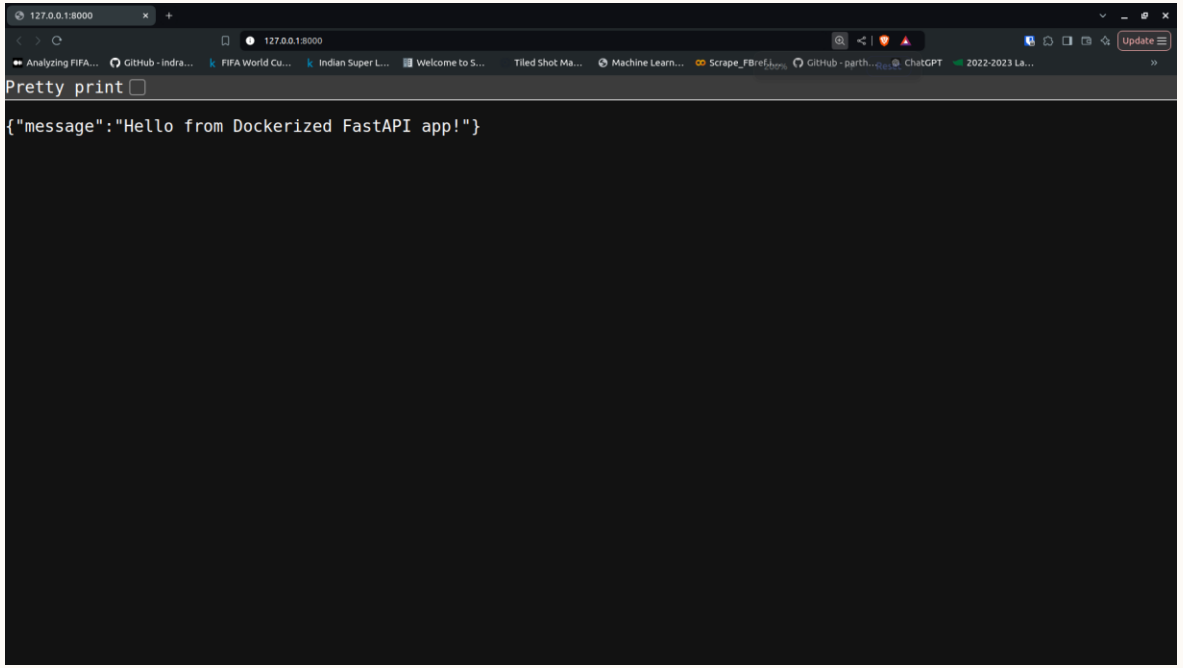
```
n$ docker build -t docker-handson .
```

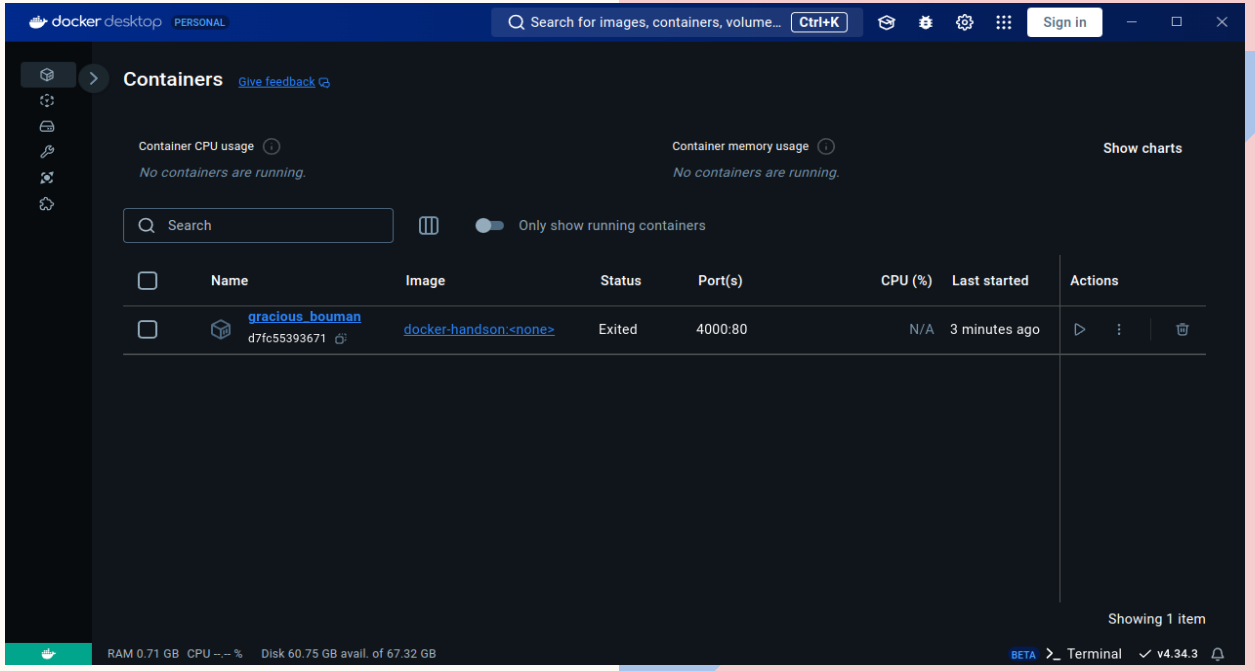


HANDS-ON

- **Step 3**: Run the image as a container with docker run

```
$ docker run -p 4000:80 docker-handson
```





HANDS-ON

Docker Commands you should know:

- docker ps: List running containers.
- docker stop <container-id>: Stop a running container.
- docker rm <container-id>: Remove a container.
- docker images: List Docker images on your system.