NAME : NAVIYA DHARSHINI
ROLL NO : 23AD083
DAY-06

09/07/2025

DOCKER

WHAT IS DOCKER?

Docker is a platform that lets you build, ship, and run applications in lightweight, portable containers.

💡 Think of containers like boxes that hold your app and everything it needs — code, libraries, tools — so it runs the same anywhere: dev, test, or prod.

🔹 CORE CONCEPTS (w/ definitions)

Term	Meaning
Image	A lightweight, standalone, read-only package that has everything needed to run a piece of software (code, runtime, tools, libs). Think of it like a cake recipe.
Container	A running instance of an image. Like baking the cake using the recipe.
Dockerfile	A text file that contains instructions to build a Docker image.
Docker Engine	The core service that creates and runs Docker containers.
Docker Hub	A cloud-based registry where you can find and share Docker images (like GitHub but for containers).
Volume	Persistent storage used by containers. Survives container restarts.
Network	Connect containers to each other or the outside world.
Bind Mount	Mount a directory from your host into the container. Useful for dev work.

Port Mapping	Expose ports from the container to the host (e.g., `-p 8080:80` means localhost:8080 will hit container's port 80).

🔹 WHY DOCKER?

- 🚀 Fast and lightweight compared to VMs
- 🔁 Reproducible environments
- 📦 Portable across dev/stage/prod
- 💥 Isolated apps (no "it works on my machine" drama)
- ☁️ Perfect for microservices and CI/CD

🔹 BASIC ARCHITECTURE

plaintext

CopyEdit

```
[ Dockerfile ] → docker build → [ IMAGE ]
                          ↓
               docker run IMAGE → [ CONTAINER ]
```

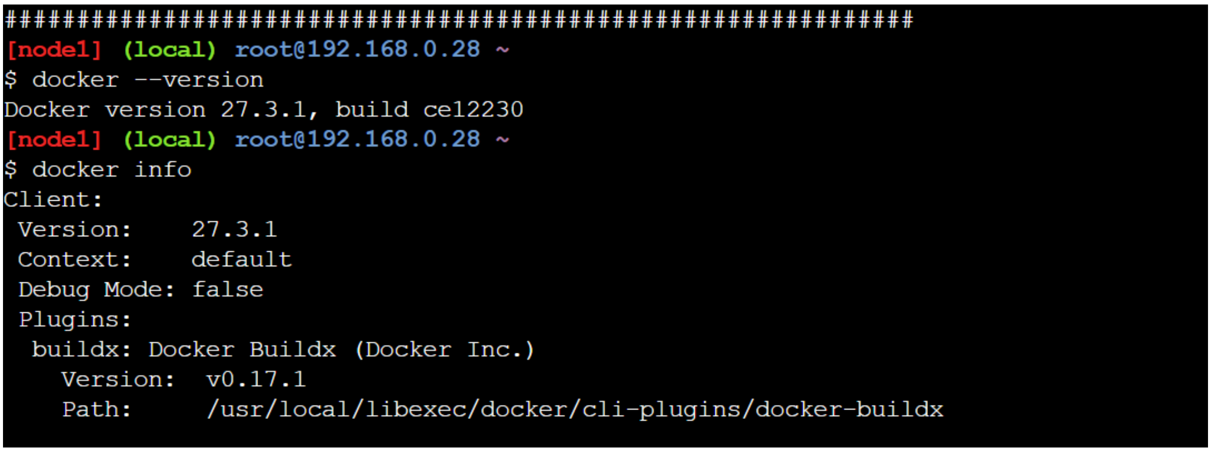
Example Flow:

1. You write a `Dockerfile`.
2. `docker build` creates an image.
3. `docker run` spins up a container from the image.

🔹 COMMON DOCKER COMMANDS

✅ Image Commands

bash

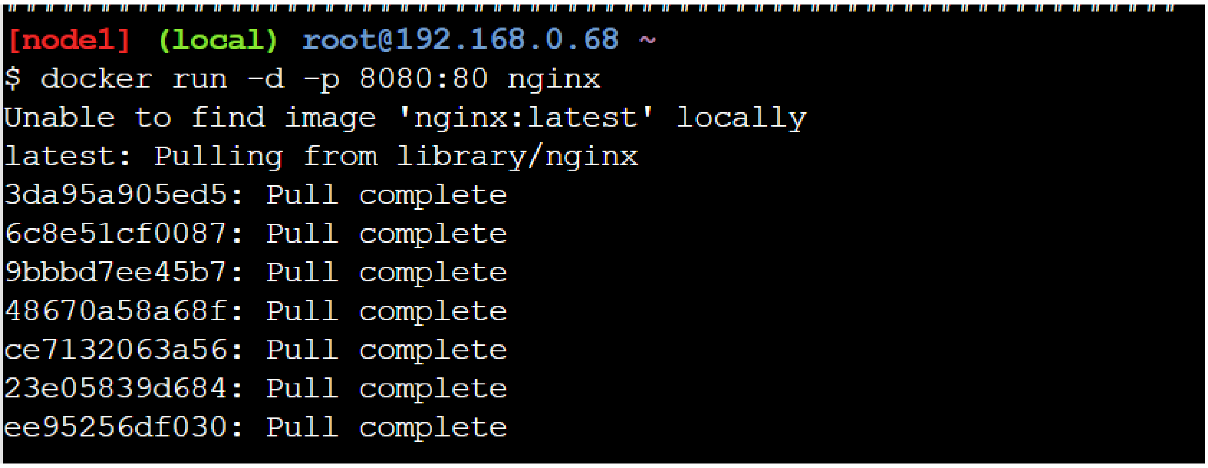


```
###############################################################
[node1] (local) root@192.168.0.28 ~
$ docker --version
Docker version 27.3.1, build ce12230
[node1] (local) root@192.168.0.28 ~
$ docker info
Client:
 Version:    27.3.1
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.17.1
    Path:     /usr/local/libexec/docker/cli-plugins/docker-buildx
```

```
docker build -t myimage:latest .
docker images                # List images
docker rmi <image-id>        # Remove image
docker pull nginx            # Download image from Docker Hub
```

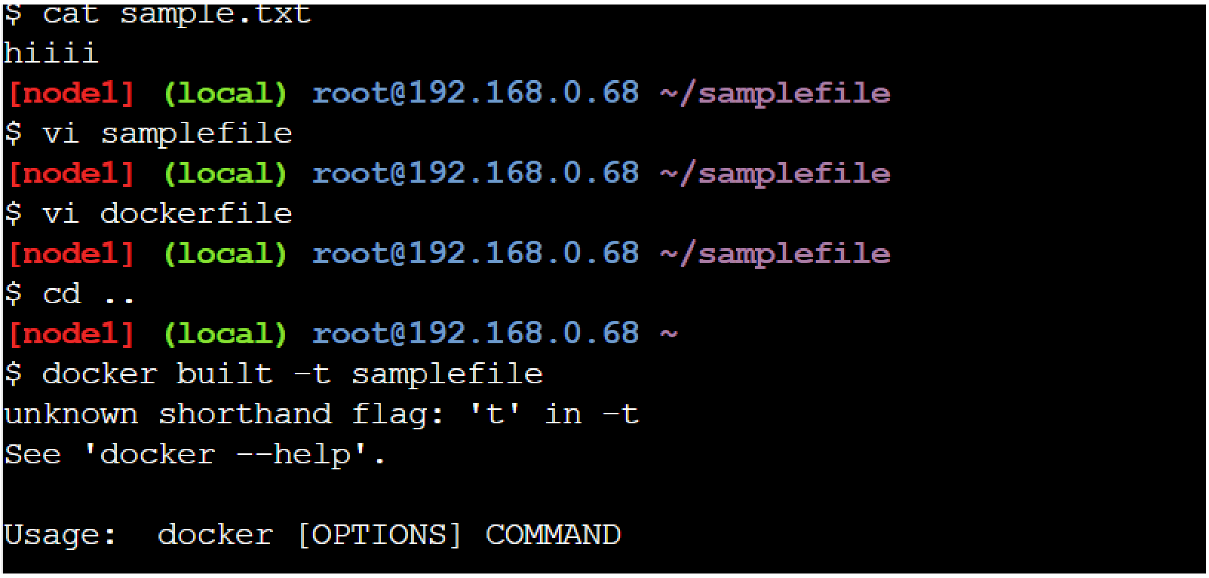
✅ Container Commands

Bash



```
[node1] (local) root@192.168.0.68 ~
$ docker run -d -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
3da95a905ed5: Pull complete
6c8e51cf0087: Pull complete
9bbbd7ee45b7: Pull complete
48670a58a68f: Pull complete
ce7132063a56: Pull complete
23e05839d684: Pull complete
ee95256df030: Pull complete
```

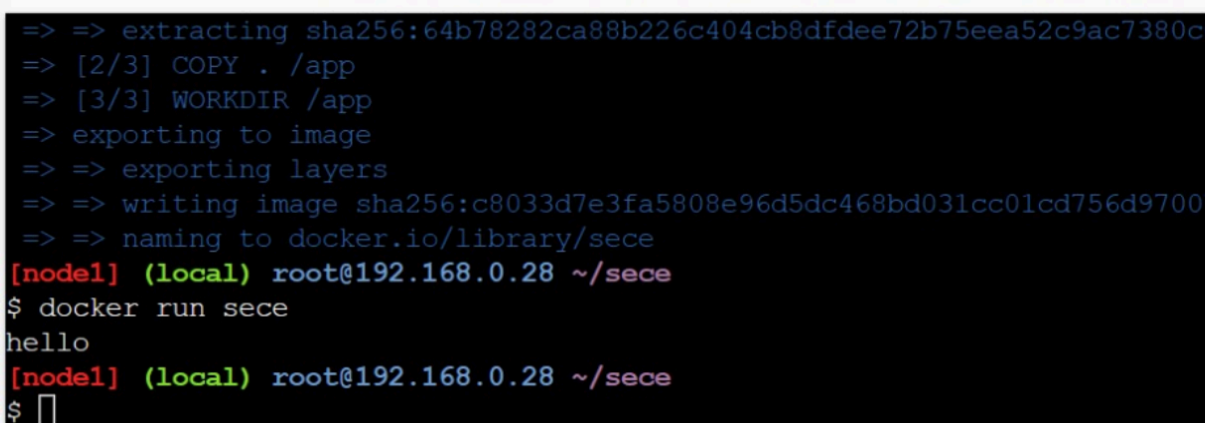
```
docker run -d -p 8080:80 --name mycontainer nginx
docker ps                    # List running containers
docker ps -a                 # List all containers (even stopped)
docker stop <container-id>
docker rm <container-id>
```



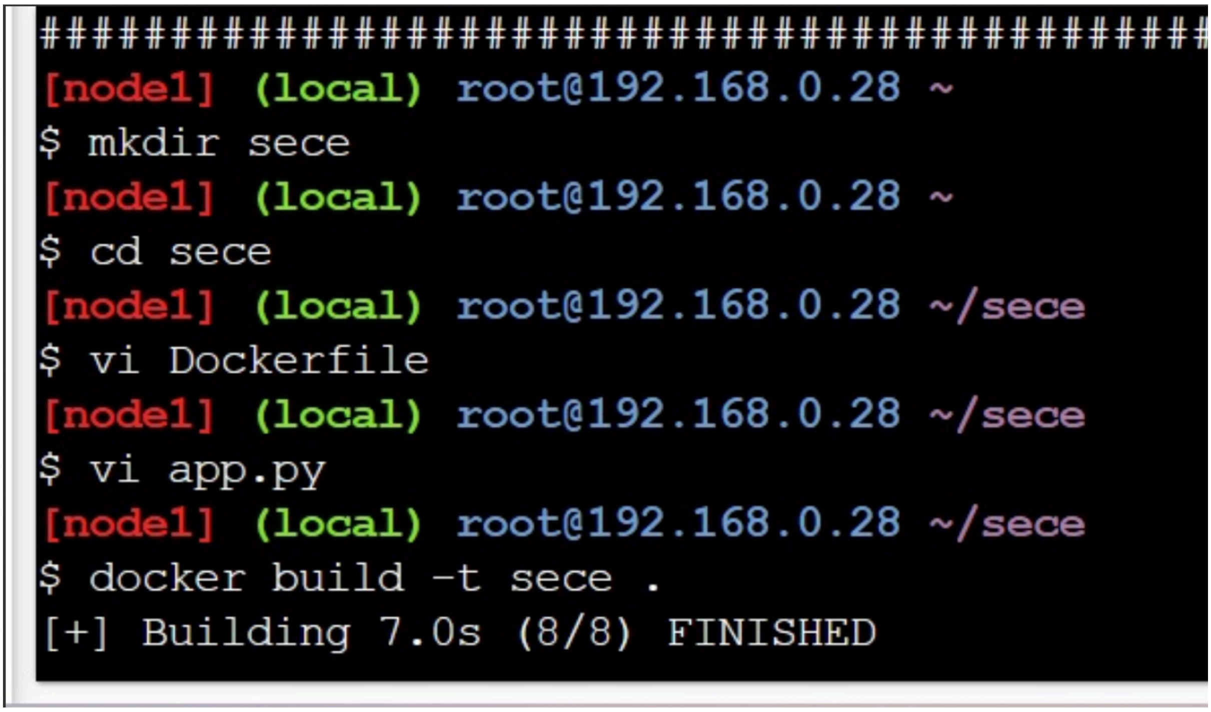
```
$ cat sample.txt
hiiii
[node1] (local) root@192.168.0.68 ~/samplefile
$ vi samplefile
[node1] (local) root@192.168.0.68 ~/samplefile
$ vi dockerfile
[node1] (local) root@192.168.0.68 ~/samplefile
$ cd ..
[node1] (local) root@192.168.0.68 ~
$ docker built -t samplefile
unknown shorthand flag: 't' in -t
See 'docker --help'.

Usage:  docker [OPTIONS] COMMAND
```

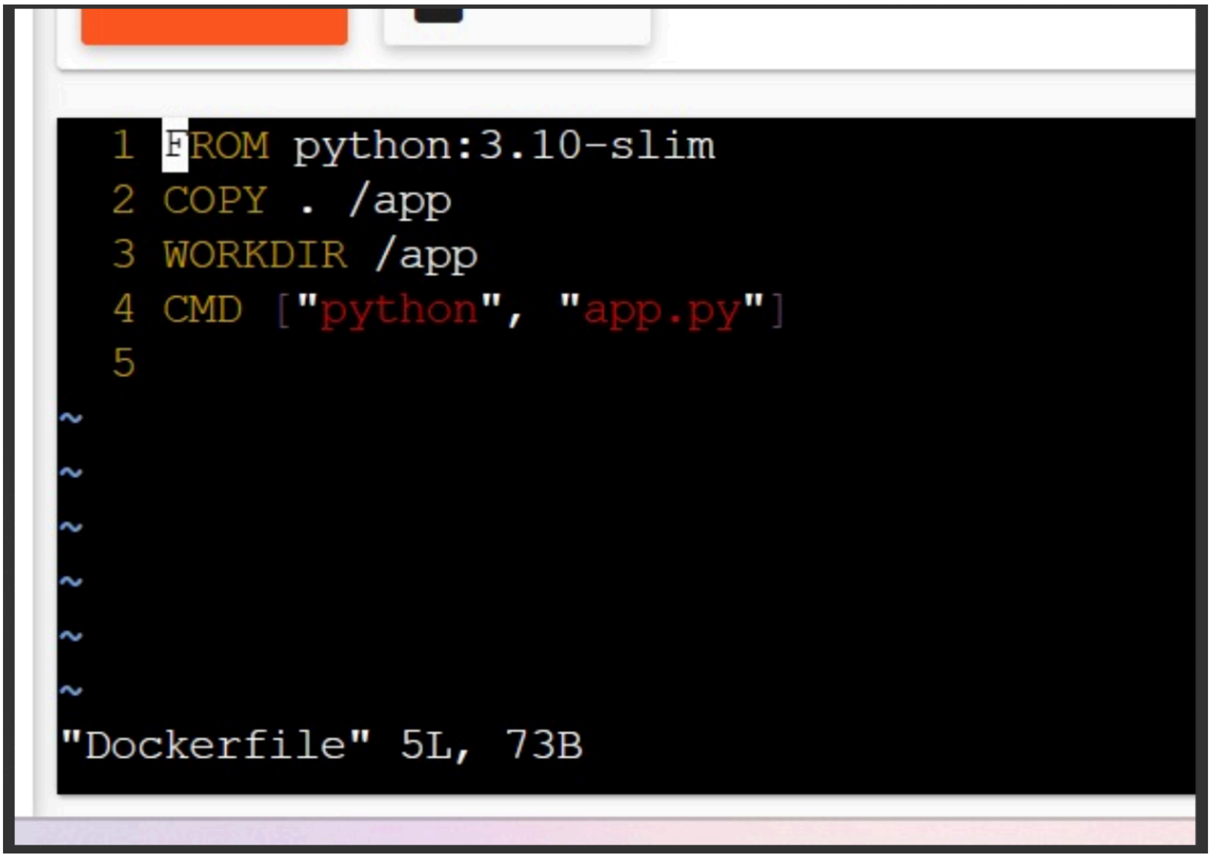
```
docker exec -it <container-id> bash   # Access shell inside
```



```
 => => extracting sha256:64b78282ca88b226c404cb8dfdee72b75eea52c9ac7380c
 => [2/3] COPY . /app
 => [3/3] WORKDIR /app
 => exporting to image
 => => exporting layers
 => => writing image sha256:c8033d7e3fa5808e96d5dc468bd031cc01cd756d9700
 => => naming to docker.io/library/sece
[node1] (local) root@192.168.0.28 ~/sece
$ docker run sece
hello
[node1] (local) root@192.168.0.28 ~/sece
$ 
```

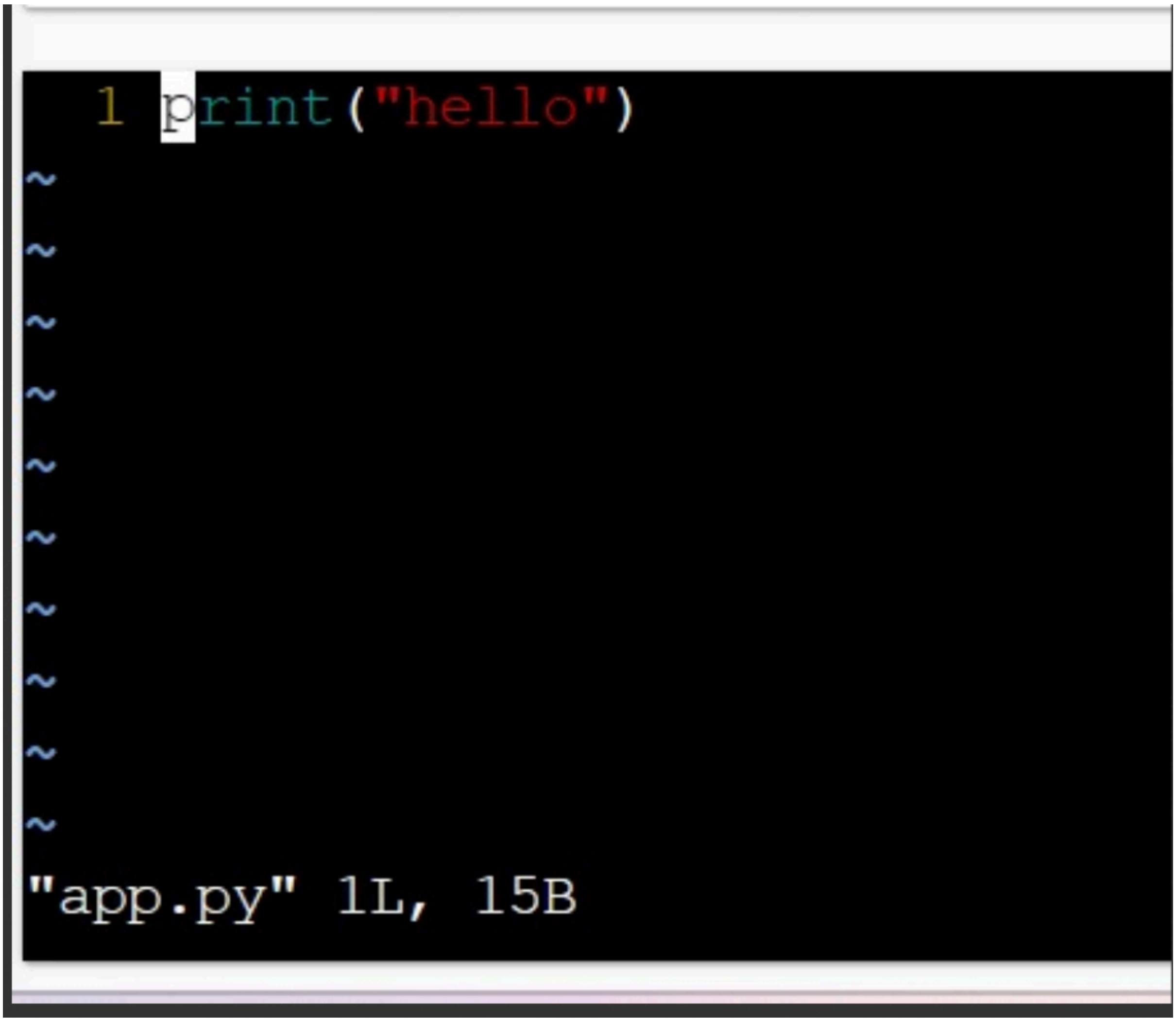


```
############################################
[node1] (local) root@192.168.0.28 ~
$ mkdir sece
[node1] (local) root@192.168.0.28 ~
$ cd sece
[node1] (local) root@192.168.0.28 ~/sece
$ vi Dockerfile
[node1] (local) root@192.168.0.28 ~/sece
$ vi app.py
[node1] (local) root@192.168.0.28 ~/sece
$ docker build -t sece .
[+] Building 7.0s (8/8) FINISHED
```



```
FROM python:3.10-slim
COPY . /app
WORKDIR /app
CMD ["python", "app.py"]

~
~
~
~
~
~
"Dockerfile" 5L, 73B
```



```
print("hello")
~
~
~
~
~
~
~
~
~
~
"app.py" 1L, 15B
```

✅ Volume Commands

bash

CopyEdit

```
docker volume create myvol
docker run -v myvol:/app/data myimage
docker volume ls
```

✅ Network Commands

bash

CopyEdit

```
docker network create mynetwork
docker network ls
docker network connect mynetwork mycontainer
```