Docker & Containerization

1. What is Containerization?

Containerization is a lightweight virtualization technique where applications run in isolated environments called **containers**. Containers share the **host OS kernel**, making them faster and more efficient than Virtual Machines.

Key Points: - Containers package application + dependencies. - They ensure the application runs the same across environments. - They are faster to start, require less resources.

Difference: VM vs Container | Feature | Virtual Machine | Container | |--------|----------------|-----------| | OS | Has full OS | Shares Host OS kernel | | Resource usage | Heavy | Lightweight | | Startup time | Slow (seconds to minutes) | Fast (milliseconds) | | Package size | Large (GBs) | Small (MBs) |

2. What is Docker?

Docker is a platform to **build, run, ship and manage containers**. It simplifies packaging applications as images and running them as containers.

Why Docker? - Consistent environment (no "works on my machine" issue) - Faster deployments - Easier scaling - Portable across cloud/server/local

3. Docker Architecture

Component	Description
Docker Client	CLI tool used to interact with Docker Daemon
Docker Daemon	Runs on host, manages images & containers
Docker Images	Read-only templates used to create containers
Docker Container	A running instance of an image
Docker Hub / Registry	Online storage for images

4. Docker Image

A **Docker Image** is a blueprint of a container. It contains: - Application code - Runtime - Libraries - Configurations

Images are **read-only**. When run, they become **containers**.

Check images:

```
docker images
```

5. Docker Container

A **container** is a runnable instance of an image.

Basic Commands:

```
docker run <image-name>
docker ps          # show running containers
docker stop <id>   # stop container
```

6. Dockerfile

Used to **build custom images**.

Example:

```
FROM python:3.9
COPY app.py /app.py
CMD ["python", "app.py"]
```

Then build image:

```
docker build -t mypythonapp .
```

7. Docker Hub

Docker Hub is a cloud-based container image registry used to **store and share Docker images**.

Login & Push:

```
docker login
docker tag mypythonapp username/mypythonapp:v1
docker push username/mypythonapp:v1
```

8. Common Useful Commands

Action	Command
Run container	`docker run <image>`
Run with interactive shell	`docker run -it <image> /bin/bash`
Stop container	`docker stop <container_id>`
Remove container	`docker rm <container_id>`

Action	Command
Remove image	`docker rmi <image_id>`
View logs	`docker logs <container_id>`

9. Real Use Cases

- Microservices deployments
- Dev / Test consistency
- CI/CD pipelines
- Cloud application scaling

10. Summary

- Containerization = Lightweight app isolation
- Docker provides tools to create & manage containers
- Images = Template | Containers = Running state
- Docker Hub = Image sharing platform