Docker

- Docker containers are lightweight, portable, and self-sufficient units that can run applications and their dependencies isolated from the host system. Containers provide process and file system isolation, ensuring that applications run consistently across different environments.

Terms in Docker –

Images

- Containers are created from Docker images, which are pre-configured, read-only templates containing the application and its dependencies.

Docker Engine

- Docker containers run on the Docker Engine, a client-server application that automates the deployment and management of containers.

Orchestration

- Orchestration tools like Docker Compose and Kubernetes help manage and scale containerized applications, ensuring seamless deployment and scaling.

DockerFile

- Docker containers are defined using Dockerfiles, which specify the steps to build a Docker image, including the base image, dependencies, and application configuration.

Volumes

- Docker volumes allow data to persist beyond the lifecycle of a container, enabling data sharing and separation of concerns.

Registry

- Docker images can be stored and shared through Docker registries like Docker Hub, enabling collaboration and easy distribution of containerized applications.

DevOps Integration

- Docker containers are integral to DevOps practices, streamlining the development, testing, and deployment processes for continuous integration and continuous delivery (CI/CD).

Docker commands:

1. Image Commands:

Pull an Image :	docker pull <image_name>
List Images :	docker images
Remove Image :	docker rmi <image_name>

2. Container Commands:

Run a Container :	docker run <options> <image_name>
List Running Containers :	docker ps
List All Containers (including stopped) :	docker ps -a
Stop a Running Container :	docker stop <container_id>
Remove a Container :	docker rm <container_id>

3. Container Lifecycle :

Start a Stopped Container :	docker start <container_id>
Restart a Container :	docker restart <container_id>
Pause/Unpause a Container :	docker pause <container_id>
	docker unpause <container_id>

4. Logs and Information:

View Container Logs :	docker logs <container_id>
Inspect Container Details :	docker inspect <container_id>

5. Interactive Mode and TTY:

Run Container Interactively :	docker run -it <image_name> /bin/bash
Attach to a Running Container :	docker exec -it <container_id> /bin/bash

6. Networking:

Expose Container Port to Host : docker run -p <host_port>:<container_port> <image_name>
View Container IP Address :

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
<container_id>
```

7. Volumes:

Mount a Volume:	docker run -v /host/path:/container/path <image_name>
List Volumes :	docker volume ls

8. Docker Compose:

Run Compose :	docker-compose up
Stop and Remove Compose Containers :	docker-compose down

9. Registry and Repository:

Login to Docker Hub :	docker login
Push Image to Registry :	docker push <image_name>

10. System Commands:

Show Docker Disk Usage :	docker system df
Clean Up Unused Resources :	docker system prune