

SESSION 1 — Docker Fundamentals & Containers Mindset

■ Objectives

- Understand containerization concepts
- Build foundational Docker skills for Kubernetes

■ Content

1. What is a container? Why Docker?

Containers are lightweight, portable units that package applications and their dependencies.

Docker provides tooling to build, run, and distribute containers efficiently.

2. Images, Layers, and Registries

Docker images are built in layers. Each instruction in a Dockerfile creates a new layer.

Registries (Docker Hub, GHCR, etc.) store and distribute built images.

3. Dockerfile Best Practices

- Use minimal base images
- Leverage multi-stage builds
- Pin versions
- Reduce layers
- Avoid copying unnecessary files

4. Build and Push Images

Commands:

- docker build -t yourname/app:tag .
- docker login
- docker push yourname/app:tag

5. Networking & Volumes

Containers communicate through Docker networks.

Volumes persist data outside the container lifecycle.

6. Hands-on Lab

- Build a simple app (Node/Python/Java)
- Create a Dockerfile
- Build & push to Docker Hub / GHCR
- Run multi-container apps with Docker Compose