**DOCKER**

Docker is a platform for developing, shipping, and running applications. It provides a way to package applications and their dependencies into standardised units called containers. These containers can then be deployed across different environments, whether it's a developer's laptop, a testing environment, or a production server, with consistent behaviour regardless of the underlying infrastructure.

Key concepts in Docker include:

1. Container: A lightweight, standalone, and executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and dependencies. Containers are isolated from each other and from the host system.

2. Dockerfile: A text file that contains instructions for building a Docker image. These instructions specify the base image to use, any additional software to install, environment variables, and other configuration settings.

3. Image: An immutable snapshot of a Docker container. Images are created from Dockerfiles and can be stored in registries such as Docker Hub or a private registry. They serve as the basis for creating and running containers.

4. Registry: A centralised repository for storing and distributing Docker images. Docker Hub is the default public registry maintained by Docker, but organisations can also set up private registries for internal use.

5. Docker Engine: The core component of Docker that manages containers. It includes a server daemon, a REST API for interacting with the daemon, and a command-line interface (CLI) for interacting with Docker.

6. Compose: A tool for defining and running multi-container Docker applications. Compose uses a YAML file to define the services, networks, and volumes that make up an application, making it easy to manage complex application architectures.

**Important Links**

* [Definition/ introduction](https://learn.microsoft.com/en-us/training/modules/intro-to-docker-containers/2-what-is-docker)
* [How docker images work](https://learn.microsoft.com/en-us/training/modules/intro-to-docker-containers/3-how-docker-images-work)
* [How docker containers work](https://learn.microsoft.com/en-us/training/modules/intro-to-docker-containers/4-how-docker-containers-work)
* [When to use docker containers](https://learn.microsoft.com/en-us/training/modules/intro-to-docker-containers/4-how-docker-containers-work)
* [Summary/overview](https://learn.microsoft.com/en-us/training/modules/intro-to-docker-containers/6-summary)