

Docker

- separate your applications from your infrastructure so you can deliver software quickly.
- manage your infrastructure in the same ways you manage your applications

Docker platform

- package and run an application in a loosely isolated environment called a container
- run many containers simultaneously on a given host
- Containers are lightweight and contain everything needed to run the application, so you don't need to rely on what's installed on the host

Use cases

- allowing developers to work in standardized environments using local containers which provide your applications and services
- highly portable - can run on a developer's local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.
- lightweight and fast. It provides a viable, cost-effective alternative to hypervisor-based virtual machines

Architecture

- Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers
- Docker client and daemon communicate using a REST API
- Another Docker client is Docker Compose, that lets you work with applications consisting of a set of containers.

Daemon and client

- The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes
- The Docker [command line] client (docker) is the primary way that many Docker users interact with Docker.
- Docker Desktop is an application that enables you to build and share containerized applications and microservices. Docker Desktop includes the Docker daemon (dockerd), the Docker client (docker), etc.

Docker registries

- A Docker registry stores Docker images.
- Docker Hub is a public registry that anyone can use.
- Another docker registry is Azure container registries (ACR)

Images

- An image is a read-only template with instructions for creating a Docker container.
- You might create your own images or you might only use those created by others and published in a registry.
- To build your own image, you create a Dockerfile with a simple syntax for defining the steps needed to create the image and run it.

Containers

- A container is a runnable instance of an image.
- By default, a container is relatively well isolated from other containers and its host machine.