# Overview

**develop, ship** and **run** anywhere



Docker is an open platform for developing, shipping, and running applications.

# Platform

Docker provides the ability to package and run an application in a loosely isolated environment called a **container**.

Many containers run simultaneously on a given **host**.

Containers are **lightweight** because they don’t need the extra load of a hypervisor, but run directly within the host machine’s kernel. This means you can run more containers on a given hardware combination than if you were using virtual machines. You can even run Docker containers within host machines that are actually virtual machines!

Docker manage lifecycle of containers.

When you’re ready, deploy your application into your production environment, as a container or an orchestrated service. This **works the same** whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

# Docker Engine

Client-server application with these major components:

* A server which is a type of long-running program called a daemon process (the dockerd command).
* A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
* A command line interface (CLI) client (the docker command).

# Components of Docker

Docker has the following components

* **Docker for Mac** − It allows one to run Docker containers on the Mac OS.
* **Docker for Linux** − It allows one to run Docker containers on the Linux OS.
* **Docker for Windows** − It allows one to run Docker containers on the Windows OS.
* **Docker Engine** − It is used for building Docker images and creating Docker containers.
* **Docker Hub** − This is the registry which is used to host various Docker images.
* **Docker Compose** − This is used to define applications using multiple Docker containers.