



Introduction to Docker

Visma Labs, Michal Lukac



What is Docker

Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises. Docker is also a company that promotes and evolves this technology, working in collaboration with cloud, Linux, and Windows vendors

It is one of the most popular virtualization or containerization technology of the current time. Building, deploying, and running of applications by using containers makes this whole process easier and cost-effective.

In containers, developers package up an application with all the needed parts (e.g. libraries, databases, etc.) and wrap them into a single package. This gives a significant performance boost to its users and reduces application size which makes it easier to use.



Advantages of using Docker

- Ensures consistency across several progressive and release cycles.
- It comes with a simple and easy initial setup.
- You can elaborate on your application life-cycle easily.
- It helps to save money by reducing operational costs.
- Docker can transport software faster (7x) than other container platforms.
- It helps to keep unnecessary and rhetorical files out of the container.
- In case, you are moving a local development machine to production disposals, you will get a life-changing experience.



- Easy architect.
- Enhanced productivity.
- Easy application isolation technique.
- Smart security management.
- Integration of Docker Swarm – a clustering and scheduling mechanism for Docker.
- Offers routing mesh.
- Reliable service.



Docker Container

Docker containers are the packages of applications created from Docker Images. It is an active illustration of Docker Images. They hold the entire package needed to run the application. Containers are the main utility provider of Docker. It enables a user to work on a cross-platform basis without any interruption. Also, a Docker container occupies less space than any other virtual machine. So, in this sense, containers are lighter and more portable.

A blue shipping container, likely a 20-foot or 40-foot unit, shown from a three-quarter perspective. It has two doors on the left side with visible locking mechanisms. The container is set against a plain white background.

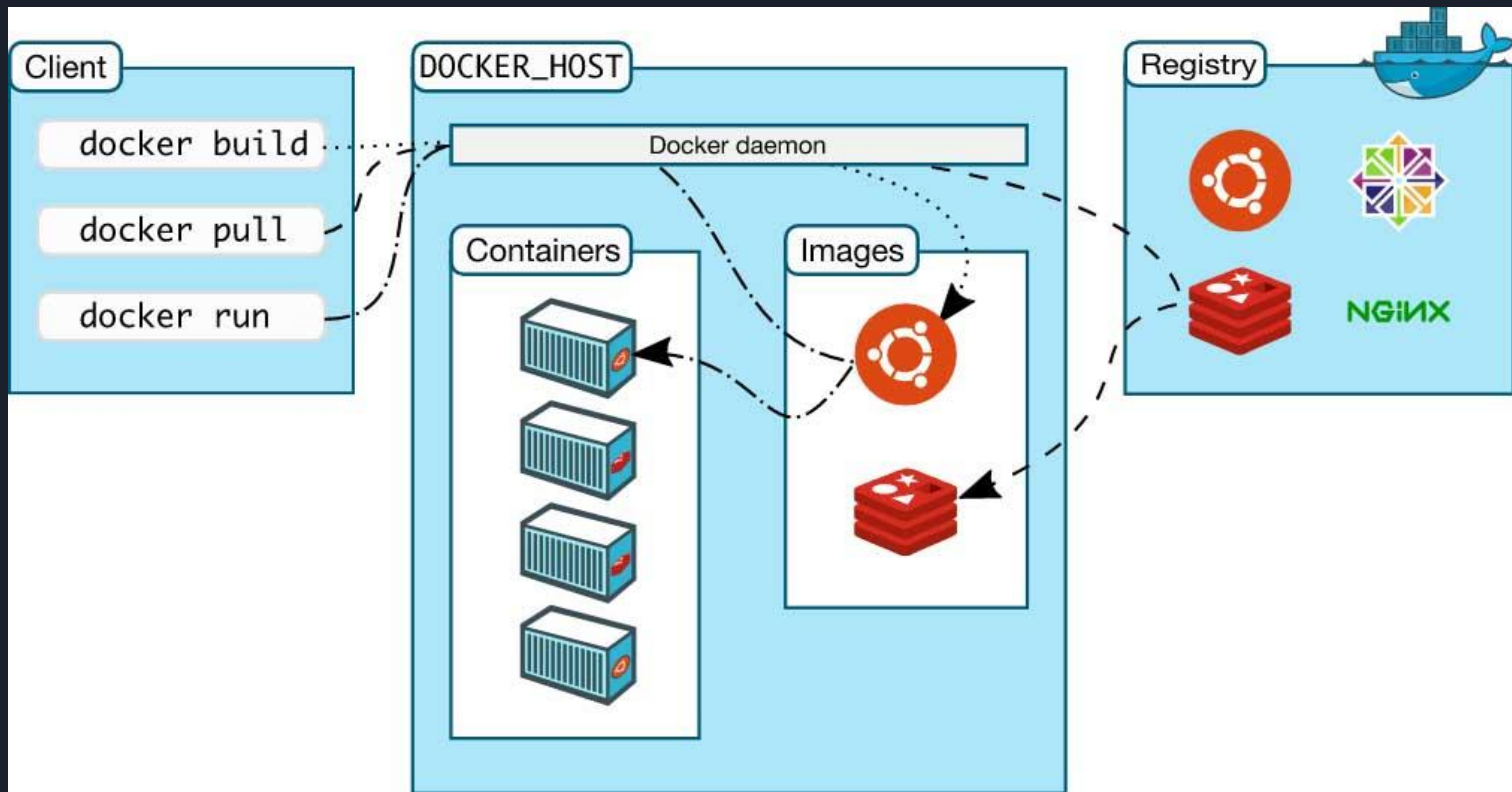
Timeline of Containerization:

- 1979:** Introduction of chroot system call
- 2000:** FreeBSD jails
- 2001:** Linux VServer project
- 2004:** Oracle Solaris Zones
- 2005:** Open Virtuozzo
- 2006:** Process Containers
- 2007:** Aix workload partition
- 2008:** LXC Linux Containers
- 2011:** Cloud Foundry Warden
- 2013:** Docker
- 2014:** Kubernetes
- 2016:** Apache Mesos Marathon

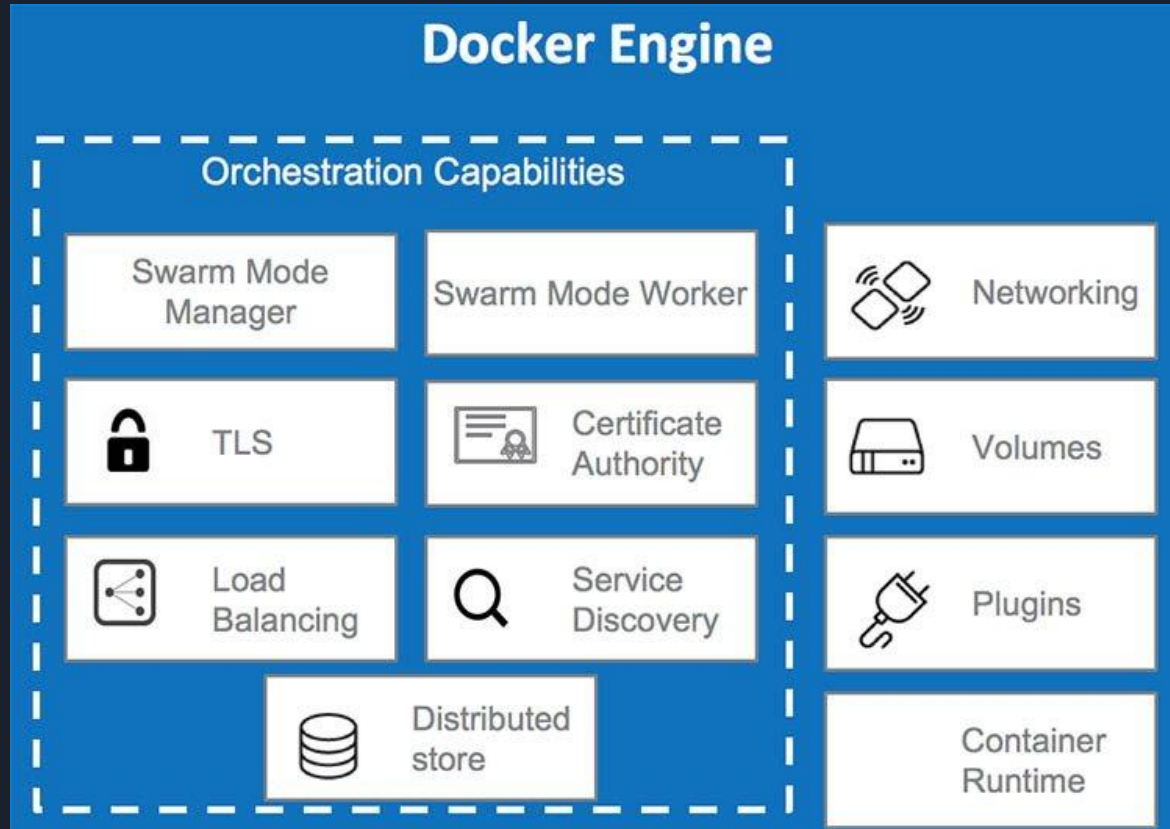
Additional milestones and details:

- 1979:** Introduction of chroot system call
- 2000:** Developed by Derrick Wolworth & adopted by Poul-Henning Kamp
- 2001:** Virtualization of operating systems
- 2004:** Addons like snapshot and cloning
- 2005:** Resource management and live migration
- 2006:** Merged in to Linux Kernel and renamed to control groups
- 2007:** Software implementation of OS level virtualization that provides app environment isolation and resource control
- 2008:** Provided a userspace interface for Linux kernel containerment features
- 2011:** The open Platform as a Service project
- 2013:** Open source version of Google's container stack, which provides Linux application containers
- 2014:** Automating deployment, scaling, and management of containerized applications
- 2016:** Windows server containers and Hyper-V containers

How Docker works



Docker Engine





Docker file

A Docker File is a simple text file with instructions on how to build your images



EXPLORER

OPEN EDITORS

DockerFile

DOCKER

DockerFile

DockerFile - Copy



DockerFile x

```
1 # mssql server 2014 express image listening on static port 1433
2 #
3 # Note: This dockerfile is based on Buc Rogers' work here:
4 # https://github.com/brogersyh/Dockerfiles-for-windows/tree/master/sqlexpress
5 #
6 # .NET 3.5 required for SQL Server
7 FROM microsoft/dotnet-framework:3.5
8
9 # maintainer for image metadata
10 MAINTAINER Perry Skountrianos
11
12 # set environment variables
13 ENV sql_express_download_url "https://download.microsoft.com/download/1/5/6/156"
14 ENV sa_password _
15 ENV attach_dbs "[]"
16
17 # make install files accessible
18 COPY . /
19 WORKDIR /
20
21 # download and install Microsoft SQL 2014 Express Edition in one step
22 RUN powershell -Command (New-Object System.Net.WebClient).DownloadFile('%sql_express_download_url%', '%WORKDIR%\SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS" /TCPENABLED=1 /NPENABLED=0 /IACCE')
23
24 RUN powershell -Command \
```



Docker compose

Docker Compose is used to run multiple containers as a single service. For example, suppose you had an application which required NGINX and MySQL, you could create one file which would start both the containers as a service without the need to start each one separately.

```
version: '2'
services:
  databases:
    image: mysql
    ports:
      - "3306:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=password
      - MYSQL_USER=user
      - MYSQL_PASSWORD=password
      - MYSQL_DATABASE=denodb
  web:
    image: nginx
```

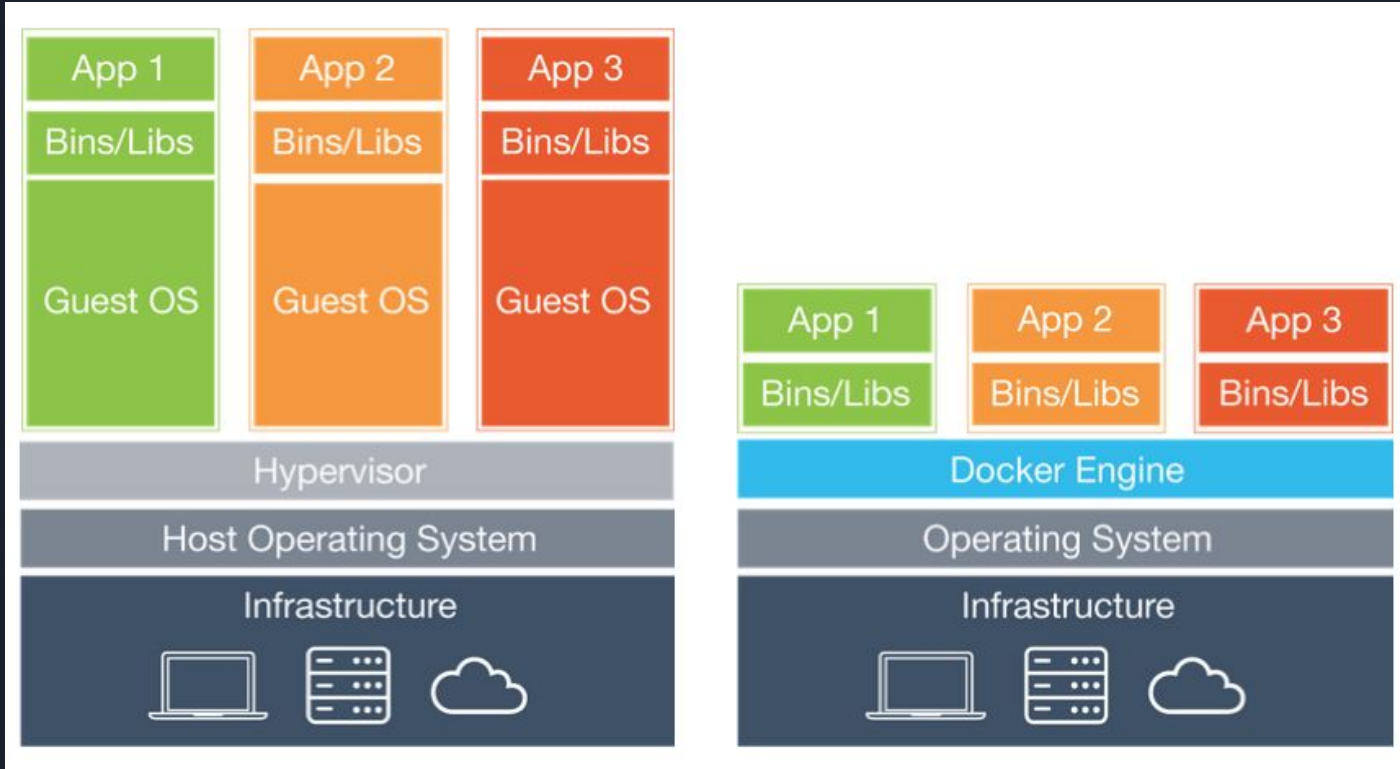


Docker hub

Cloud-based application registry and development team collaboration services.

Docker Hub provides some useful features such as repositories for push/pull container images, official and publisher images, webhooks. And it can naturally build advanced container images from BitBucket and GitHub.

Docker vs VMs





- Docker is easier to set up but without a robust cluster. However, Kubernetes comes with a complicated setup with the assurance of a robust cluster.
- Docker offers scalability that is five times faster than Kubernetes.
- However, unlike Kubernetes, Docker does not have any GUI.
- Docker Swarm, clustering and scheduling tool of Docker, offers automatic load balancing whereas in Kubernetes you need manual intervention for load balancing of traffic.
- Docker Swarm needs third party tools for logging and monitoring, Kubernetes does not.
- Docker swarm can share storage volumes with other containers easily but Kubernetes is limited in terms of data volumes as it can share storage with other containers in the same pod only.



Q&A

Thank you for attendance