



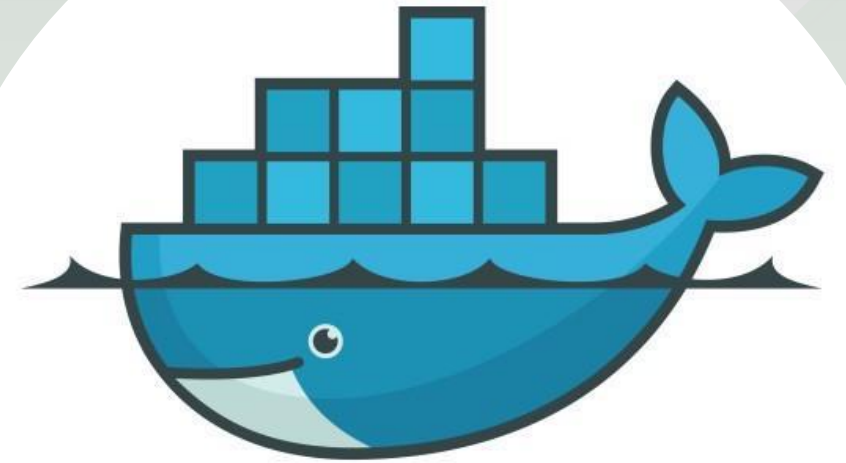
BATCH	:	B107 AWS-DevOps
LESSON	:	Docker
DATE	:	10.04.2023
SUBJECT	:	Intro

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ





Introduction to Docker



docker



Environmental Drift

Once in a country, there was a vigilant programmer in a company.
The company had a program to calculate financial transactions.





Environmental Drift

The program used to round some interest amounts to single decimal points.

Our programmer discovered that and thought he could use the rounded and removed parts for himself.

2567,19.00
,19.00



Environmental Drift

The programmer wrote an application that would get the removed fractional parts of the amounts and send them to his bank account.

The amounts were just the tiny fractional places so that no one could notice.

0,19

0,35

0,23

0,09

0,17

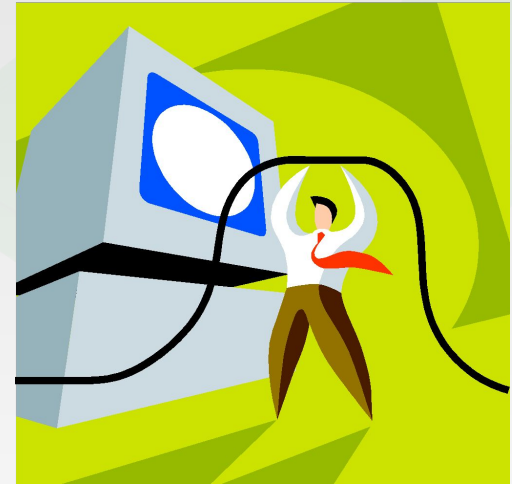
0,03

1,06



Environmental Drift

The vigilant programmer was a good programmer.
As good programmers did, he tested his app thoroughly. It worked very well!
Then he deployed it on the production environment.





Environmental Drift

After some time, when he checked his bank account, he was shocked!
The result was not as he expected!
There is no way, the amount was large enough to be noticed!

350456,23



Environmental Drift

The vigilant programmer cried: "But it worked on my computer!"

DEV



PROD





Environmental Drift

What was the vigilant programmer's fault?

He did not know that the production environment was up to date but his computer was not.

The last Saturday he said he would stay at the office after office hours. His lead forgot to inform about a patch he applied on the prod servers.





Environmental Drift

So, the main causes for the error:

- ▶ The code
- ▶ App server
- ▶ Runtime version
- ▶ System libraries
- ▶ Etc..



Environmental Drift

The programmer tested:

- ▶ The Code

Forgot:

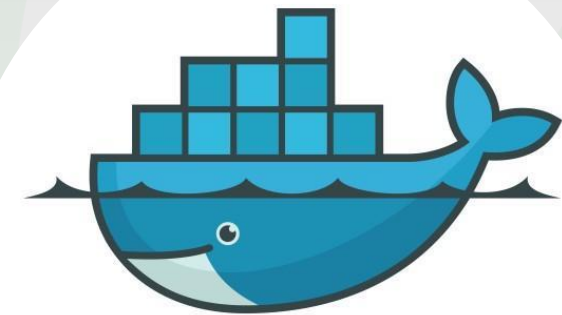
- ▶ App server
- ▶ Runtime version
- ▶ System libraries
- ▶ Etc..



Environmental Drift

Solution for the problem is having all these together, even changing from dev to prod

- ▶ The code
- ▶ App server
- ▶ Runtime version
- ▶ System libraries
- ▶ Etc..



docker



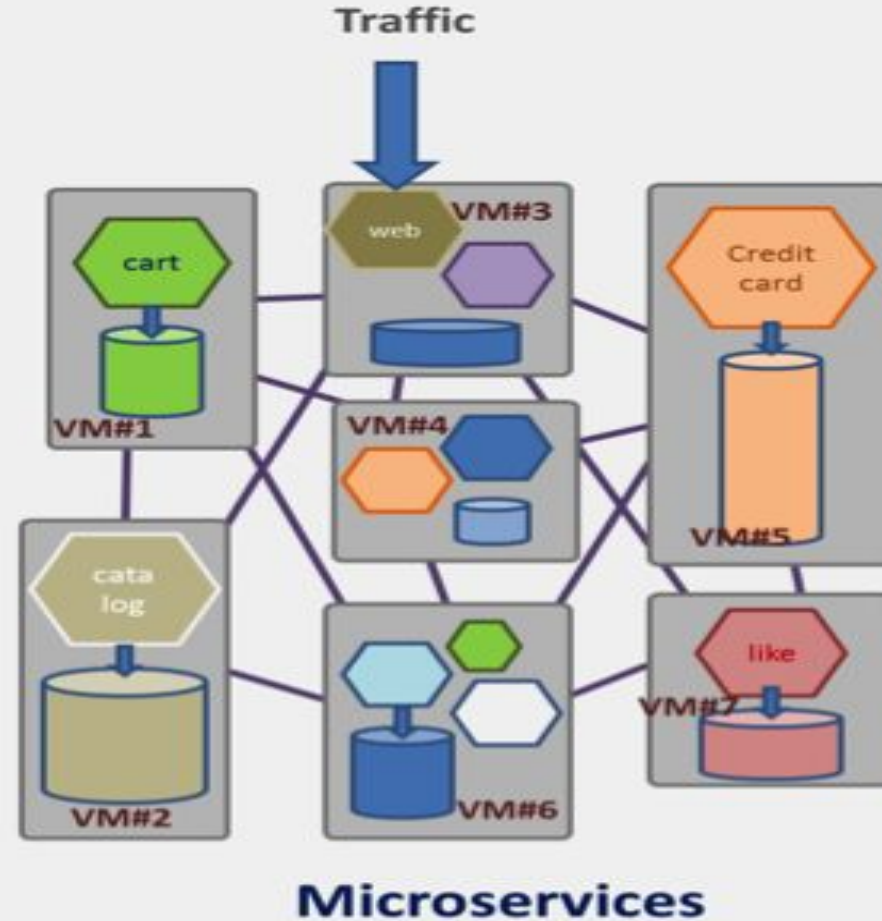
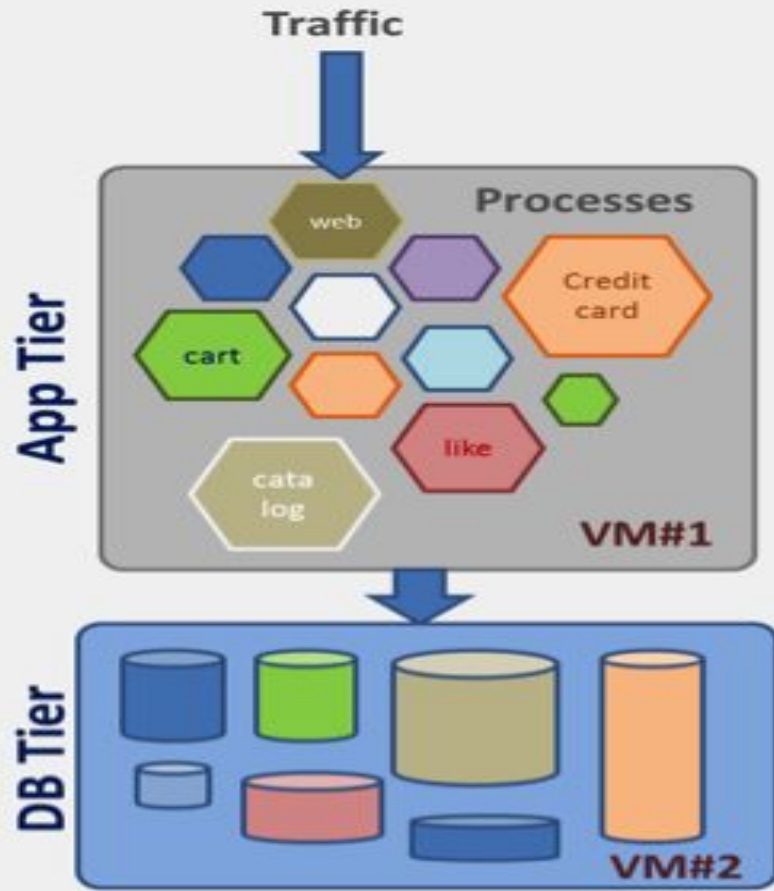
Content

- Terminology
- What is container?
- Container vs. VMs
- What is Docker?
- Docker Architecture





Monolithic vs Microservice





Monolithic



●ステレオ録音に威力を発揮する
ワンポイントステレオマイク
別売：CM-2000 ¥7,900

Everything is integrated



Modular



Each part is independent

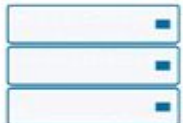


Microservice

Monolithic Architecture



App Services

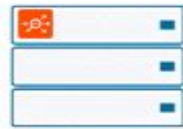


Bare Metal

Microservices Architecture



Microservice



Bare Metal



Microservice



Virtualized



Microservice



Containers



Microservice



Public Cloud

Applications

A **microservice** is an application **with a** single function, such as routing network traffic, making an online payment or analysing a medical result.

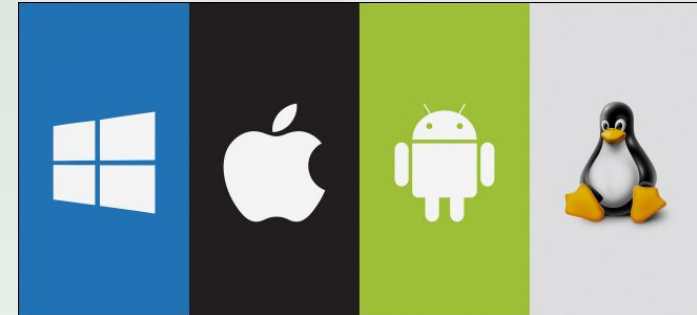


How an application runs

An application needs an **OS**, a **runtime**,
application code itself to run.

It needs an environment.

Container technology provides this
environment.





What is Container?



Containers are easily packaged, lightweight virtual computing environments and they are designed to run anywhere.



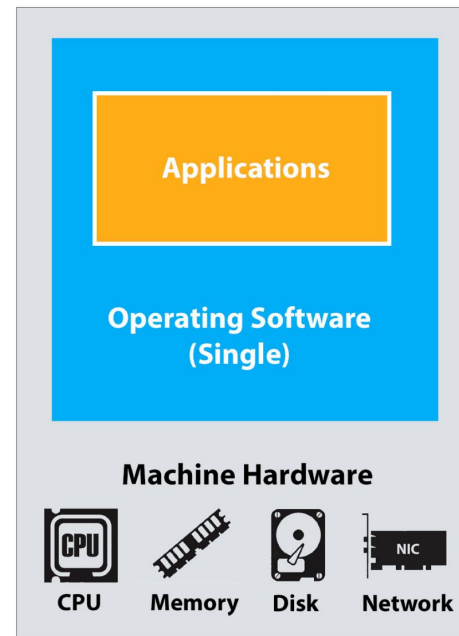


What is a Virtual Machine?

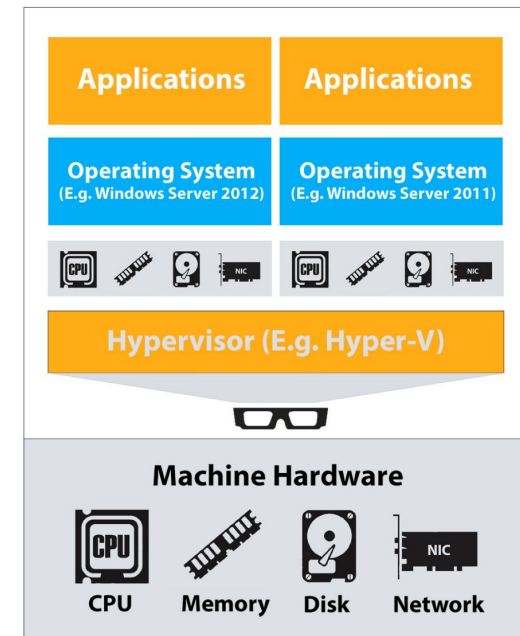


vmware®

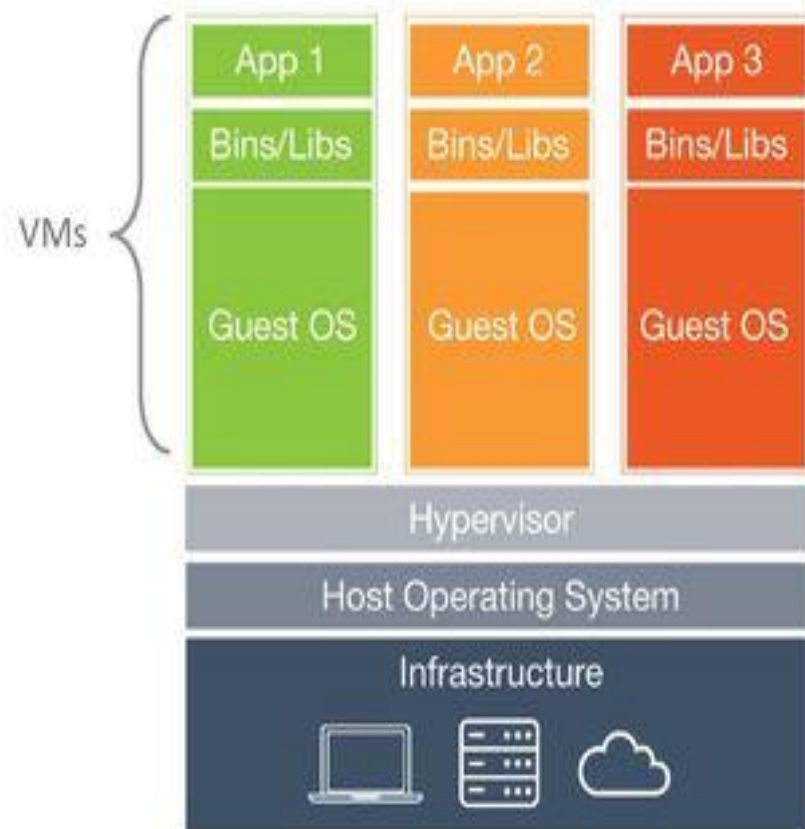
A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps.



Physical Machine

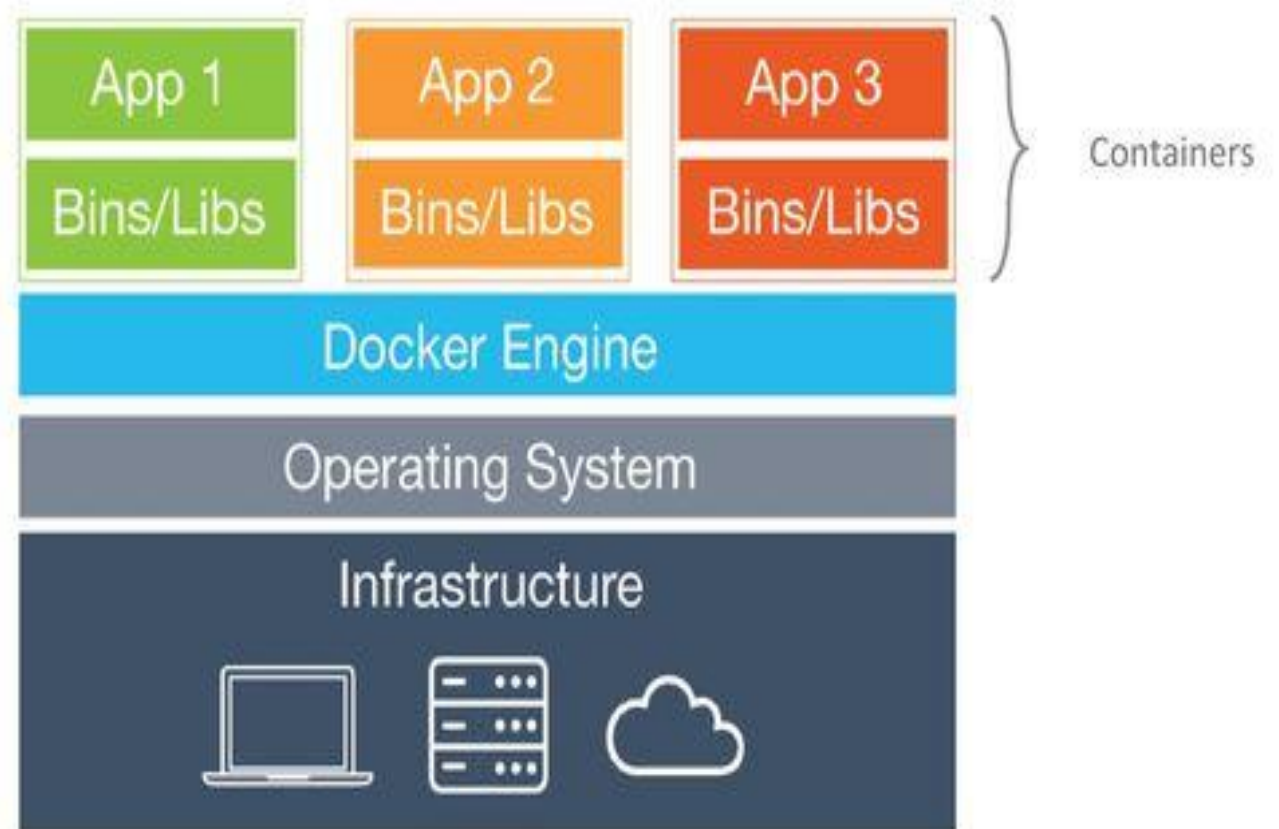


Virtual Machine



Virtual Machines

- Each virtual machine (VM) includes the app, the necessary binaries and libraries and an entire guest operating system



Containers

- Containers include the app & all of its dependencies, but share the kernel with other containers.
- Run as an isolated process in userspace on the host OS
- Not tied to any specific infrastructure – containers run on any computer, infrastructure and cloud.

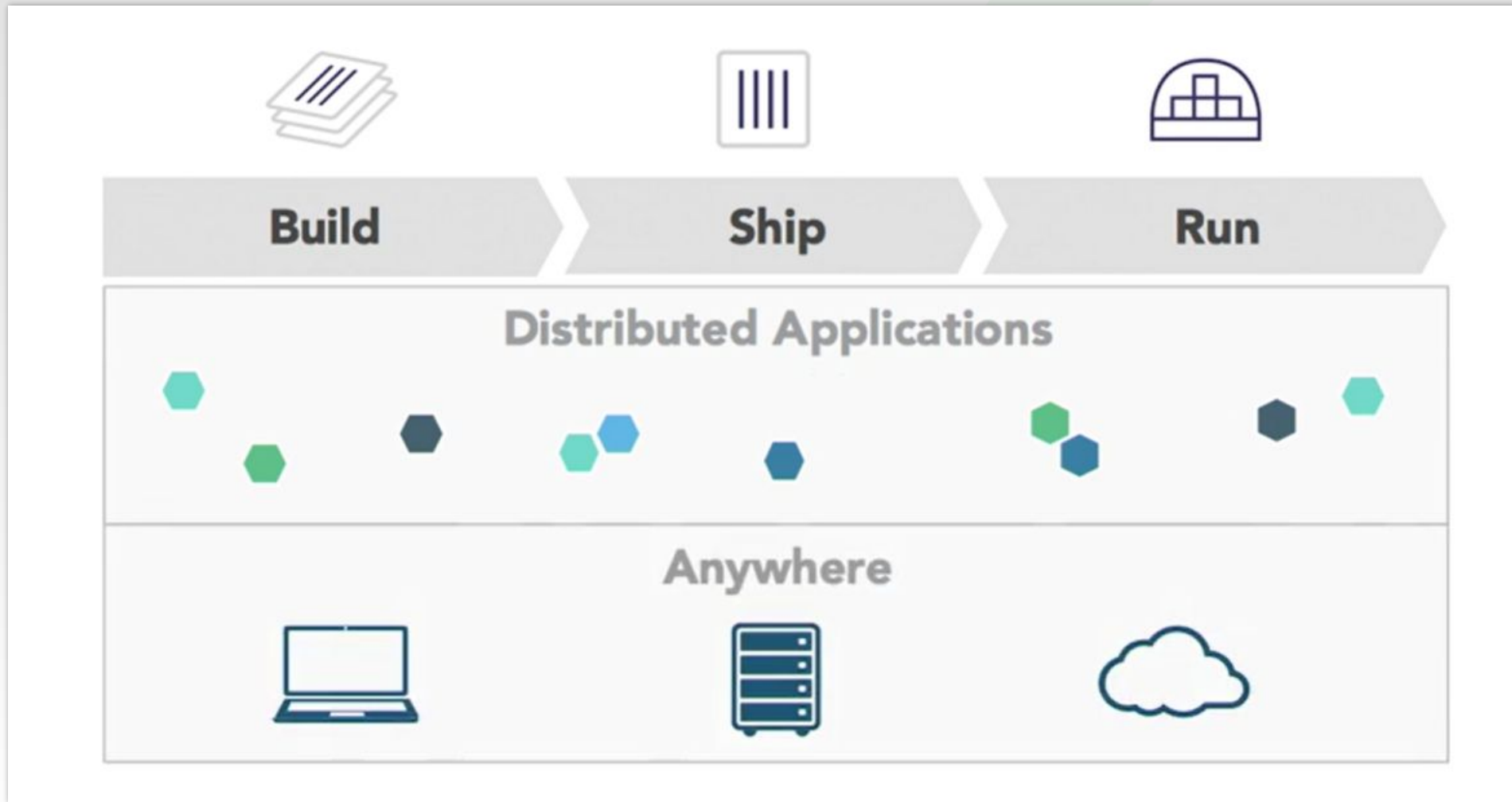


Virtual Machines vs Containers

Virtual Machines	Docker
Each VM runs its own OS	All containers share the same Kernel of the host
Boot up time is in minutes	Containers instantiate in seconds
VMs snapshots are used sparingly	Images are built incrementally on top of another like layers. Lots of images/snapshots
Not effective diffs. Not version controlled	Images can be diffed and can be version controlled. Dockerhub is like GITHUB
Cannot run more than couple of VMs on an average laptop	Can run many Docker containers in a laptop.
Only one VM can be started from one set of VMX and VMDK files	Multiple Docker containers can be started from one Docker image

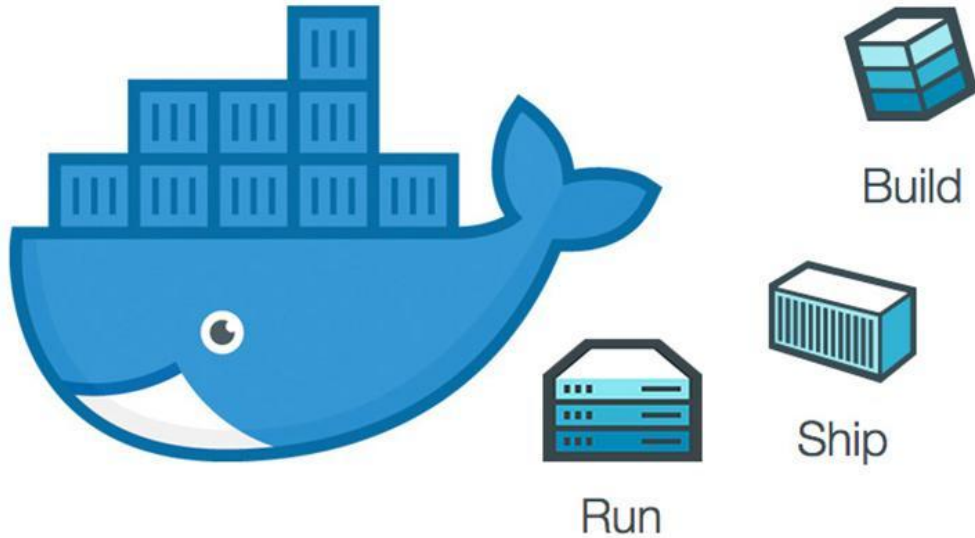


Container's Mission





What is Docker?



- Docker is an open source project. It was started in 2013.
- It is very popular and used in the market to deploy apps as a **container**.
- It can run on docker engine that can be installed on every kind of OS's.



What Docker does?

What Docker Does

- Carves up a computer into sealed containers that run your code
- Gets the code to and from your computers
- Builds these containers for you
- Is a social platform for you to find and share containers, which are different from virtual machines



Build and ship any application
anywhere !



Why Docker?

Ease of use: It allows anyone to package an application on their laptop, which in turn can run unmodified anywhere

The mantra is: “build once, run anywhere.”

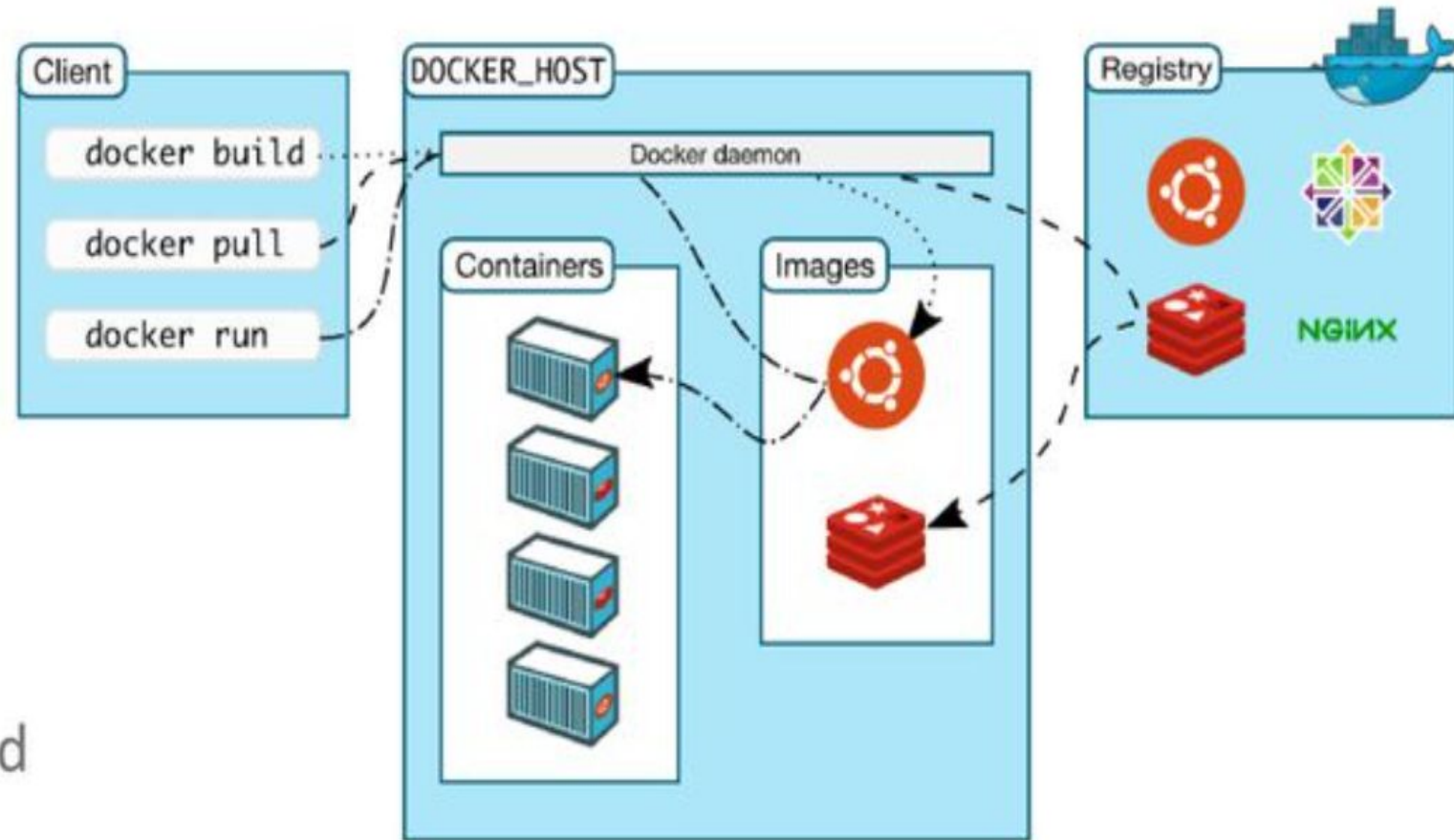
Speed: Docker containers are very lightweight and fast. You can create and run a Docker container in seconds, compared to VMs which might take longer because they have to boot up a full virtual operating system every time.

Docker Hub: You can think of as an “app store for Docker images.” Docker Hub has tens of thousands of public images created by the community that are readily available for use.

Modularity and Scalability: Docker makes it easy to break out your application’s functionality into individual containers making it easy to scale or update components independently in the future.

Docker Architecture

- Docker client – Command Line Interface (CLI) for interfacing with the Docker
- Dockerfile – Text file of Docker instructions used to assemble a Docker Image
- Image – Hierarchies of files built from a Dockerfile, the file used as input to the docker build command
- Container – Running instance of an Image using the docker run command
- Registry – Image repository





Docker Architecture

Docker Hub



Images

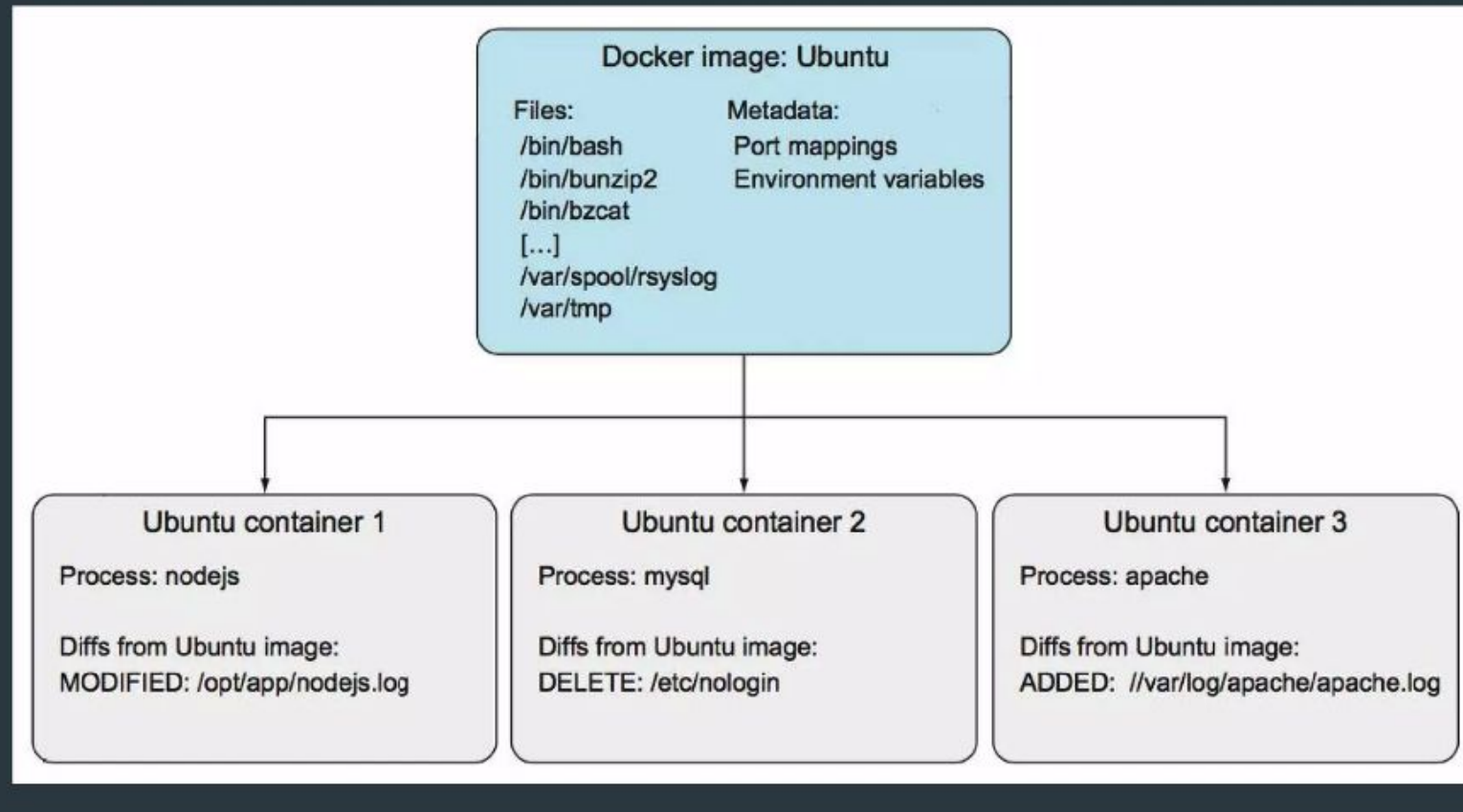


**Images
Containers**



Docker Architecture

- Image and Container





Docker Architecture

Dockerfile

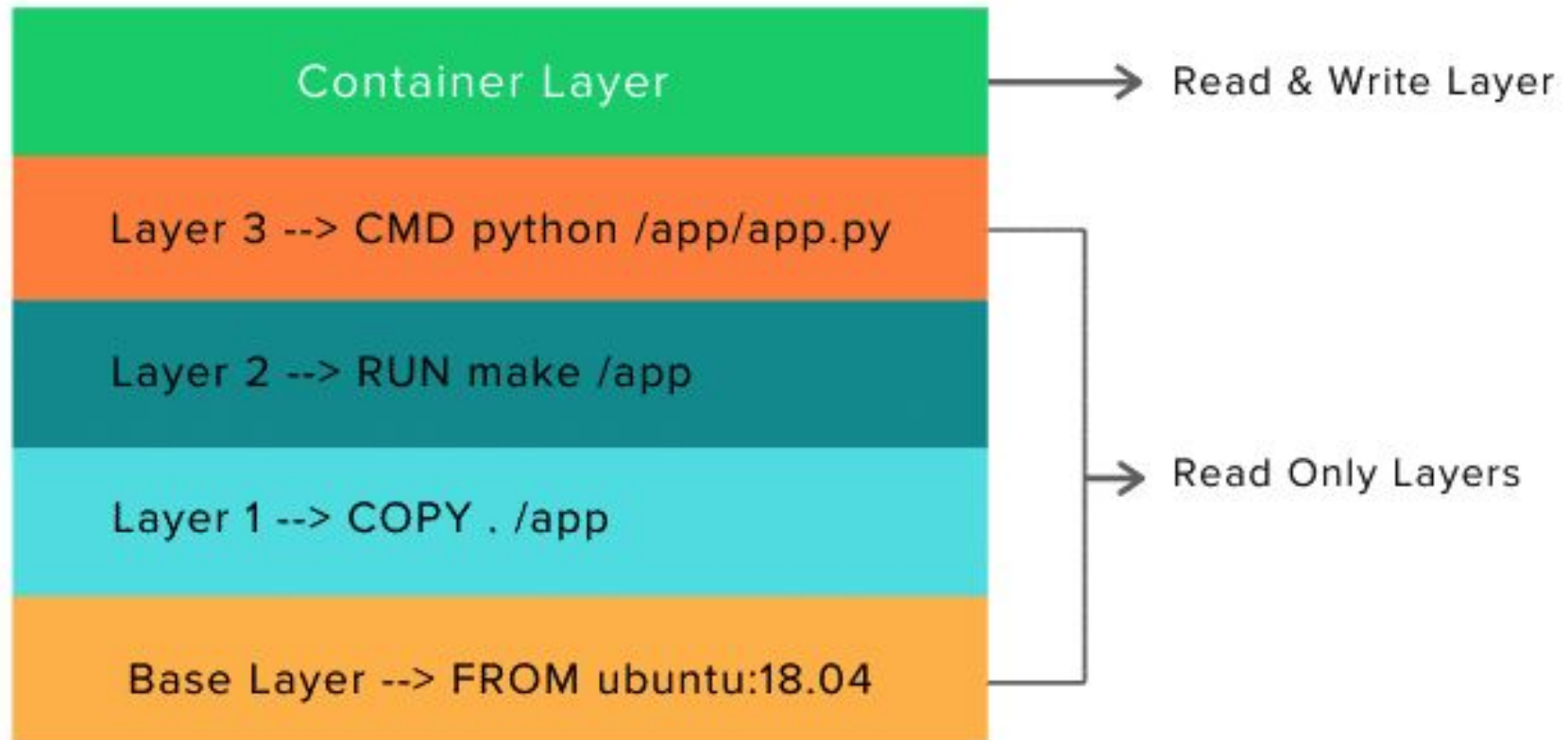
- A textfile
- List of commands to build a Docker image

```
FROM tomcat:9.0
COPY . /app
WORKDIR /app
RUN ./mvnw package -DskipTests
RUN cp /app/target/*.war /usr/local/tomcat/webapps/ROOT.war
CMD ["catalina.sh", "run"]
```



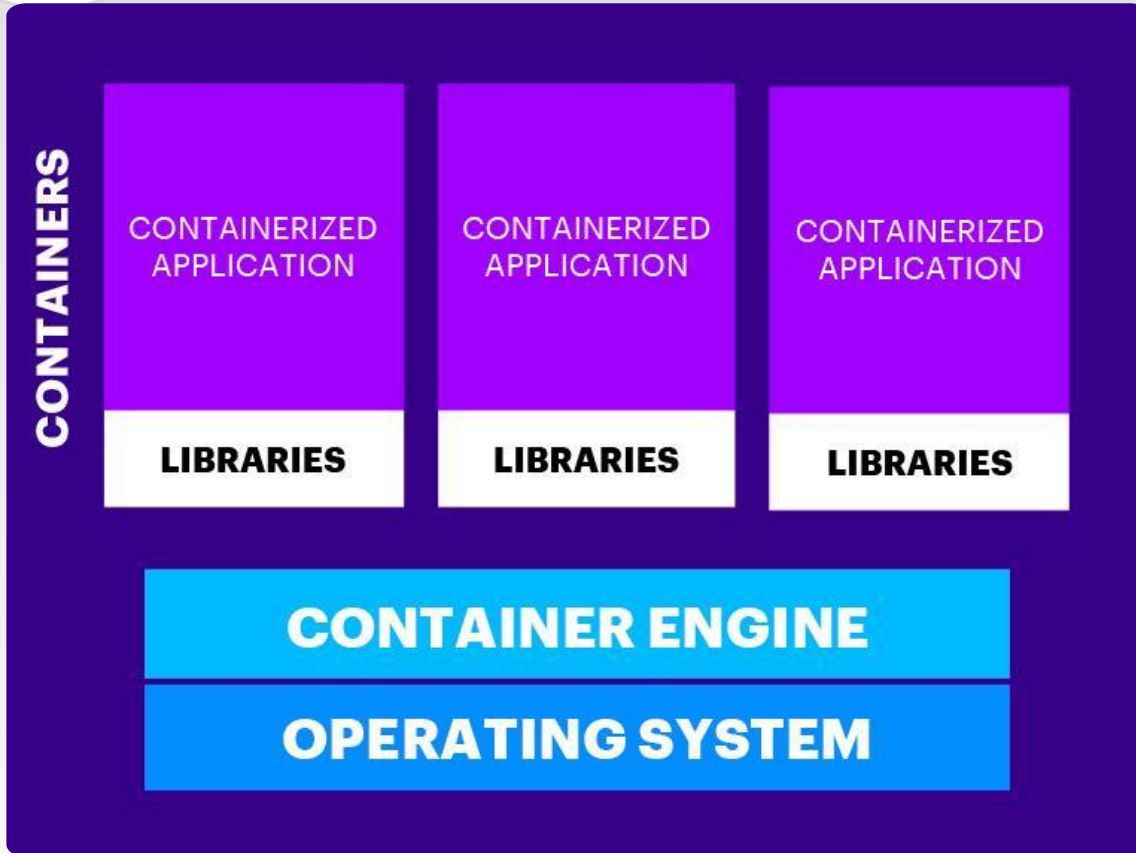


Docker Architecture





Docker Engine



- Container execution and admin
- Uses Linux Kernel
- Linux Namespaces and Control Groups
- Namespaces provide for isolated workspace



Docker CLI Commands

Docker start: Start one or more stopped containers

Docker stop: Stop one or more running containers

Docker run: Run a command in a new container

Docker restart: Restart one or more containers

Docker rm: Remove one or more containers

Docker rmi: Remove one or more images

Docker ps: List containers

Docker image: Manage images

Docker images: List images

Docker exec: Run a command in a running container

Docker inspect: Return low-level information on Docker objects

Docker logs: Fetch the logs of a container

Docker network: Manage networks

Docker volume: Manage volumes

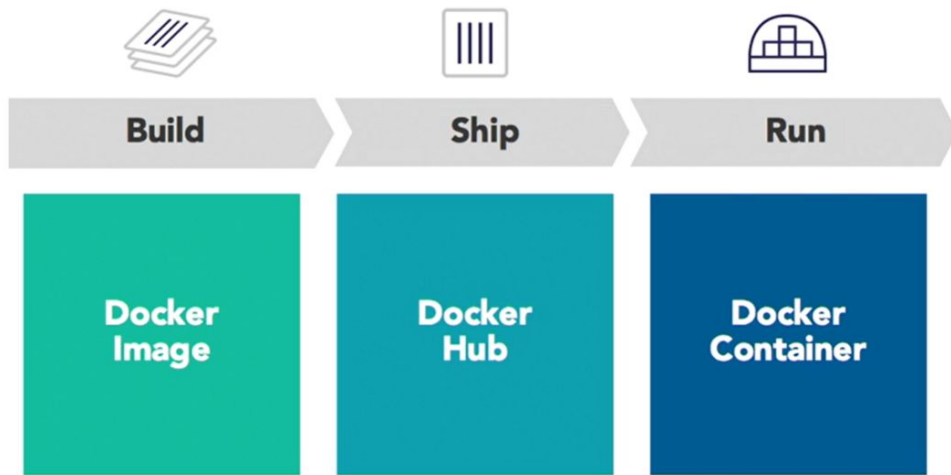
Docker pull: Pull an image or a repository from a registry

Docker push: Push an image or a repository to a registry



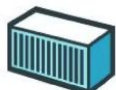
Summary

Docker Concept



Build

Develop an app using Docker containers with any language and any toolchain.



Ship

Ship the "Dockerized" app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.



Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

- Docker is an application delivery technology
- Build an application as an image
- Ship your image(application) by using Docker Hub
- Run the application as a Docker Container on any host environment
- Avoid single point of failure that crashes whole the application by using microservices



Tips

- Containers are stateless, they do not store your data inside.
- Each container gets an IP address at creation.
- namespaces: running isolated processes
- cgroup: assign resources to namespaces
- container: running processes with dedicated resources
- Docker runs on Linux, on platforms like MacOS, Windows, it uses a tiny Linux environment
- Containers are used for a single application. They are the basic of microservices.
- Docker is made up of
 - a CLI
 - a background daemon (service)
 - REST API



Docker Environment



Docker Desktop

Developer productivity tools
and a local Kubernetes
environment.

Download for
Windows



Docker Hub

Cloud-based application
registry and development
team collaboration services.

Signup



Play with Docker

Cloud-based docker
environment to try out
docker and learn the ropes.

Play with Docker

- Docker-Desktop
- Docker-Hub
- Play with Docker



Docker in Real Life

“Each week we launch more than 2 billion container instances across our global data centers, ...”

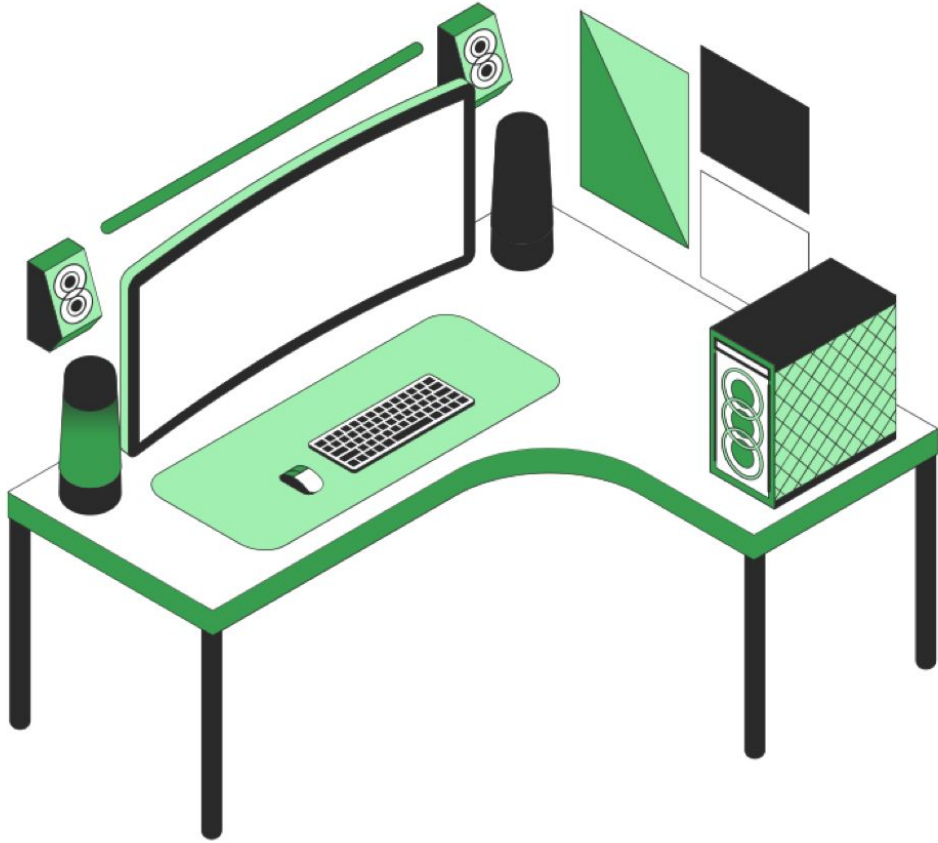
Eric Brewer, VP of Google Infrastructure

“At Instagram, we deploy our backend code 30–50 times a day... whenever engineers commit changes to master... with no human involvement in most cases.”

Michael Gorven, Production Engineer at Facebook

“Everything at Google, from Search to Gmail,
is packaged and run in a Linux container.”

Eric Brewer, VP of Google Infrastructure



Do you
have any
questions?

Send it to us! We hope you learned
something new.