



**BATCH** : BATCH 85  
**LESSON** : Docker  
**DATE** : 26.09.2022  
**SUBJECT** : Docker Introduction



techproeducation



techproeducation



techproeducation



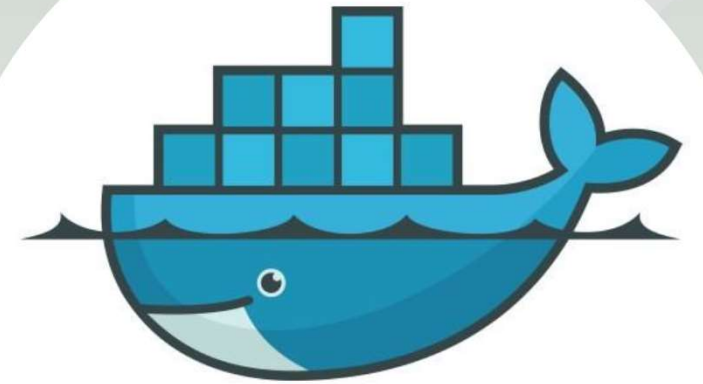
techproeducation



techproedu



# Introduction to Docker



docker



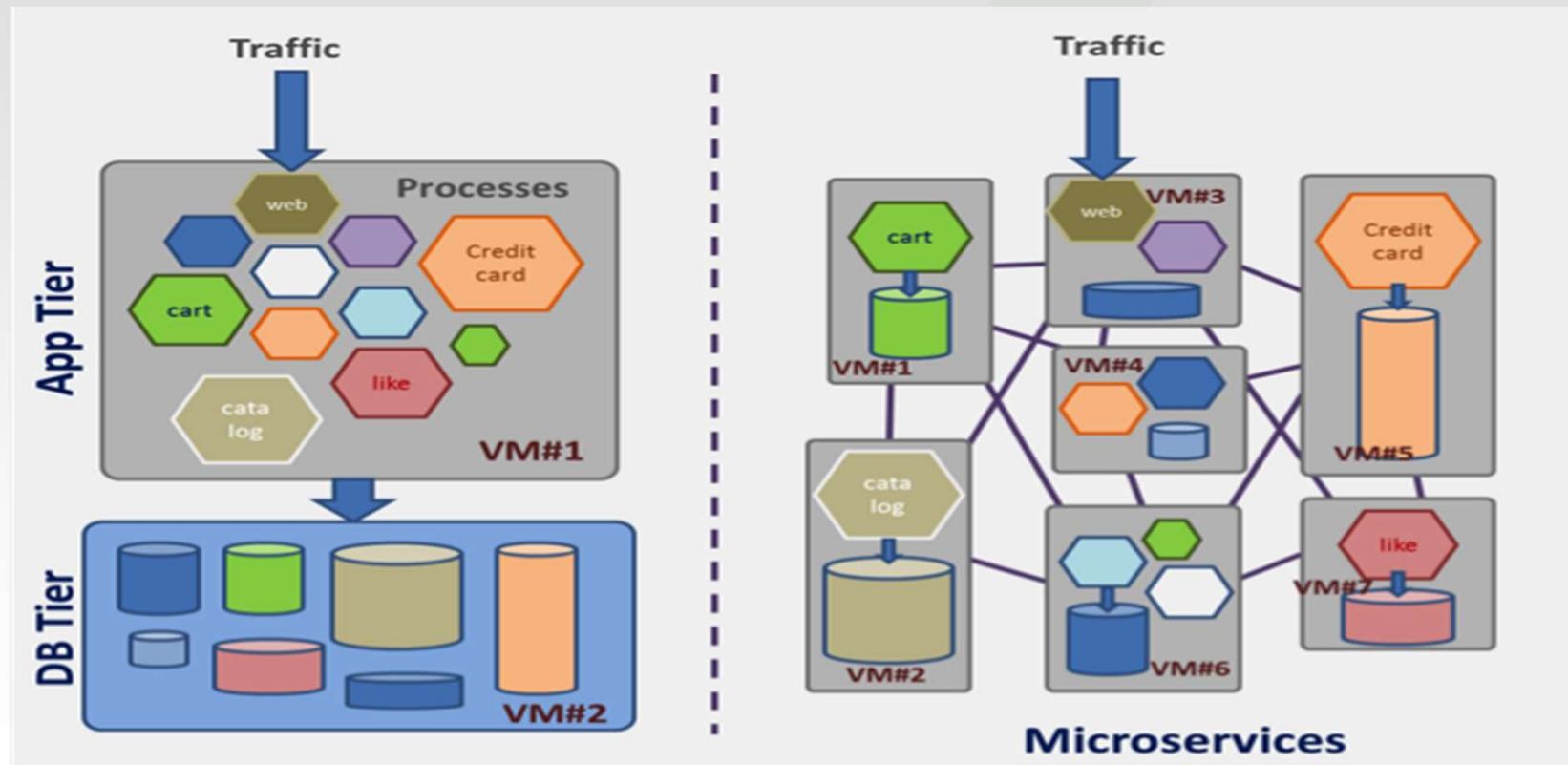
# Content

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





# Monolithic vs Microservice







# Monolithic



Everything is integrated



# Modular



Each part is independent

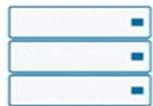


# Microservice

*Monolithic Architecture*

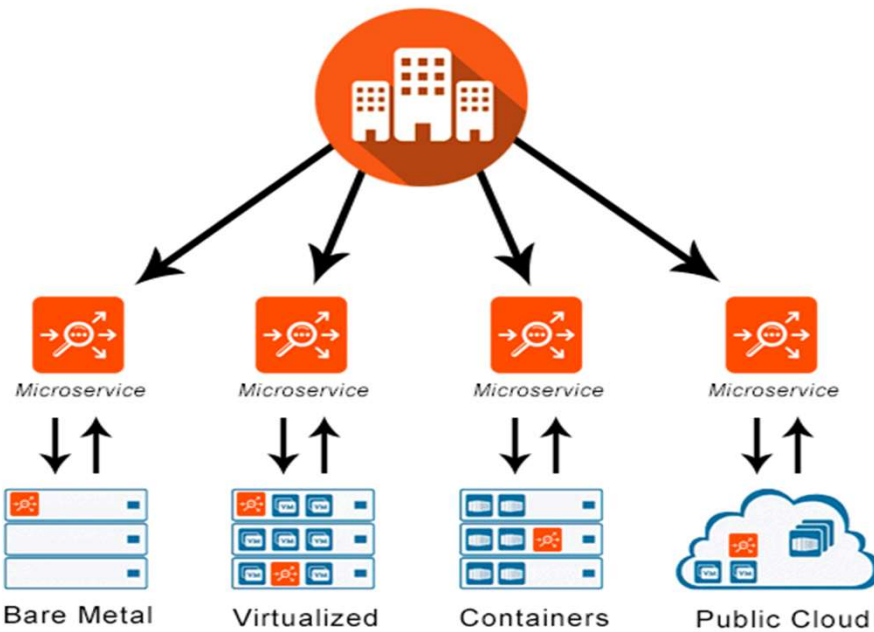


*App Services*



*Bare Metal*

*Microservices Architecture*



*Applications*

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

# What is Container?



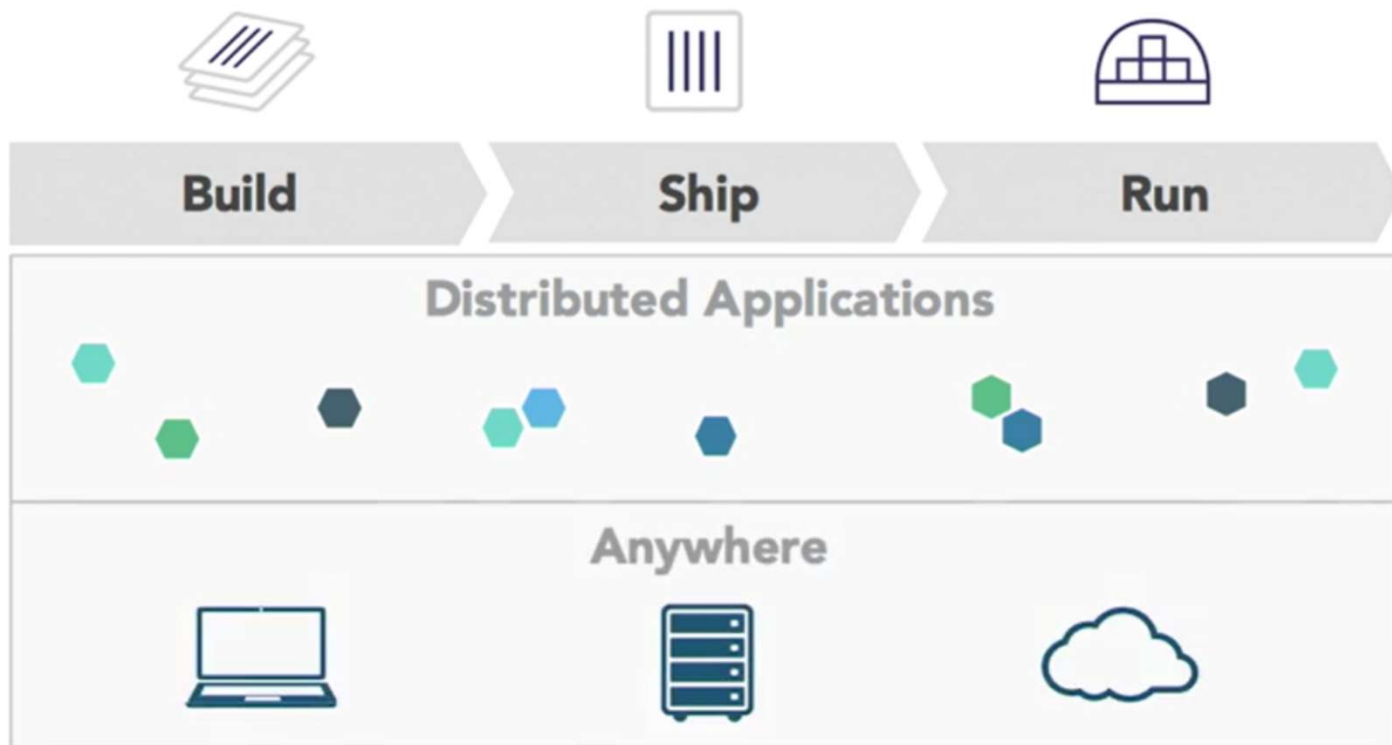
**Containers** are easily packaged, lightweight and designed to run anywhere. Multiple **containers** can be deployed **in a** single host.





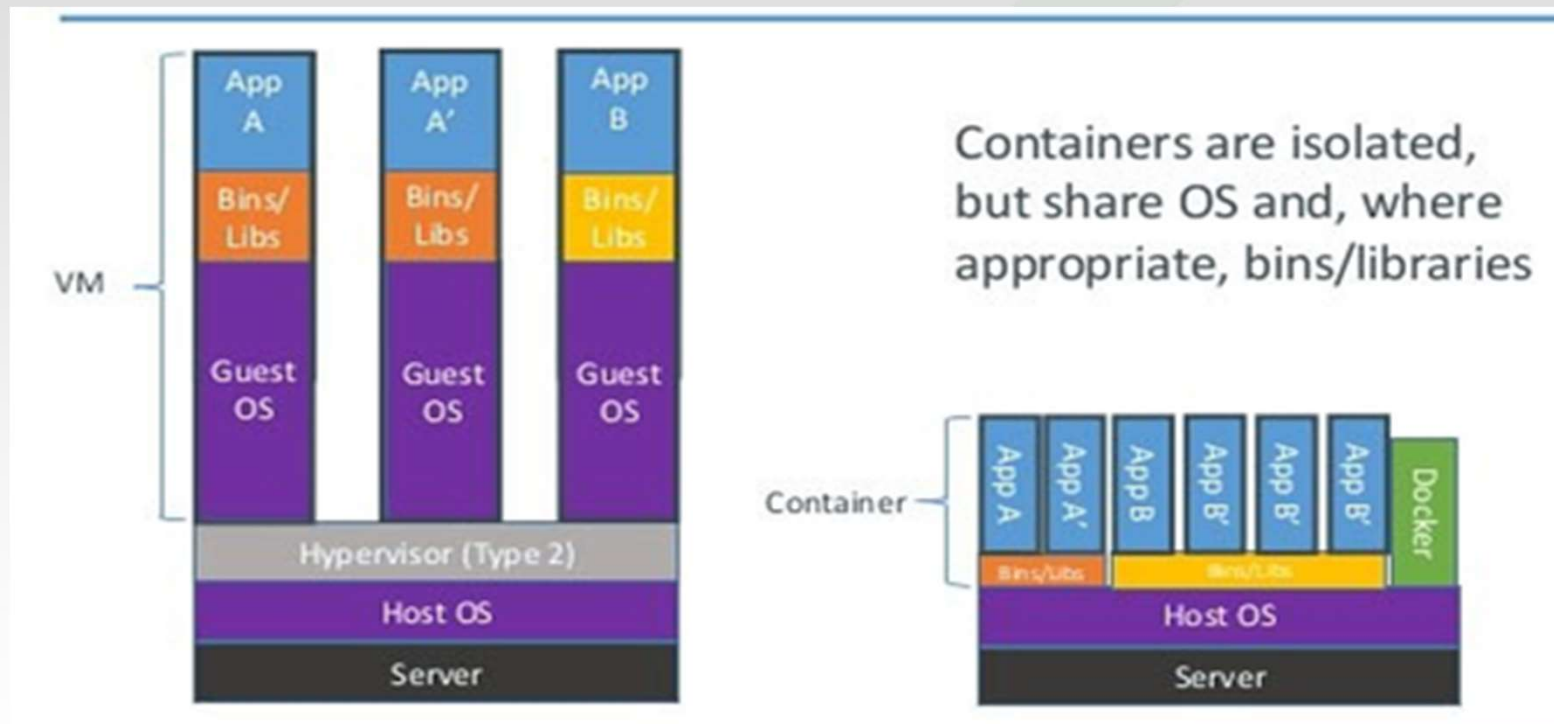


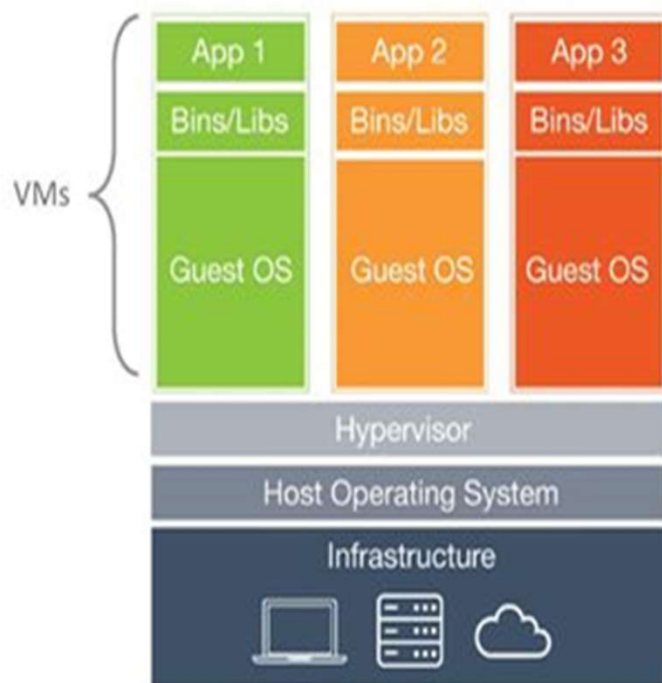
# Container's Mission





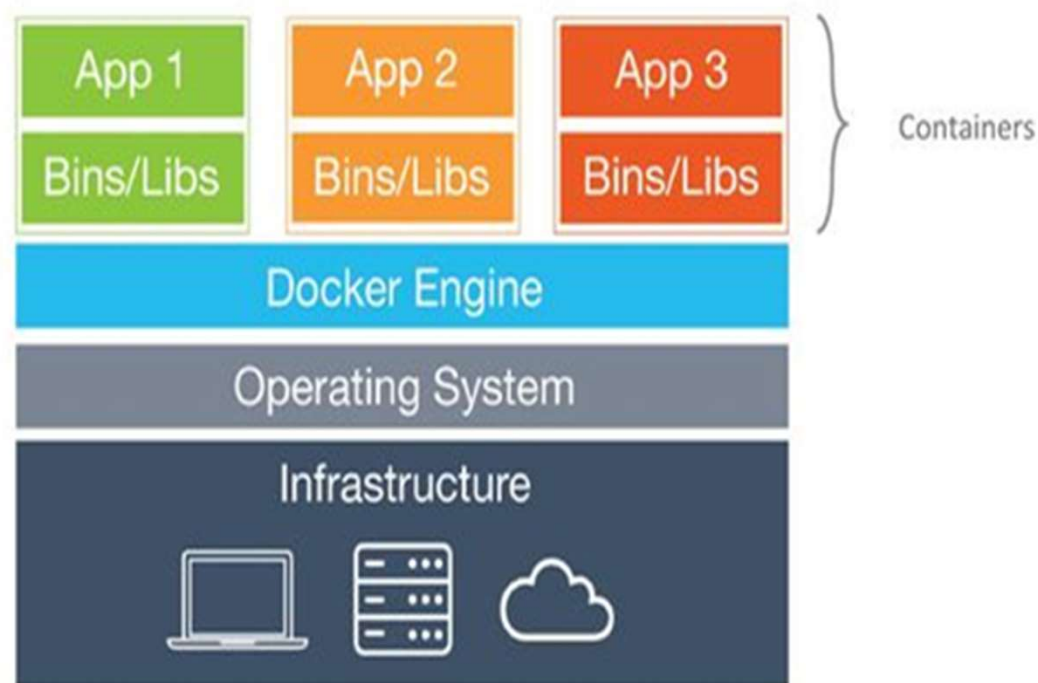
# Virtual Machines vs Containers





## Virtual Machines

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

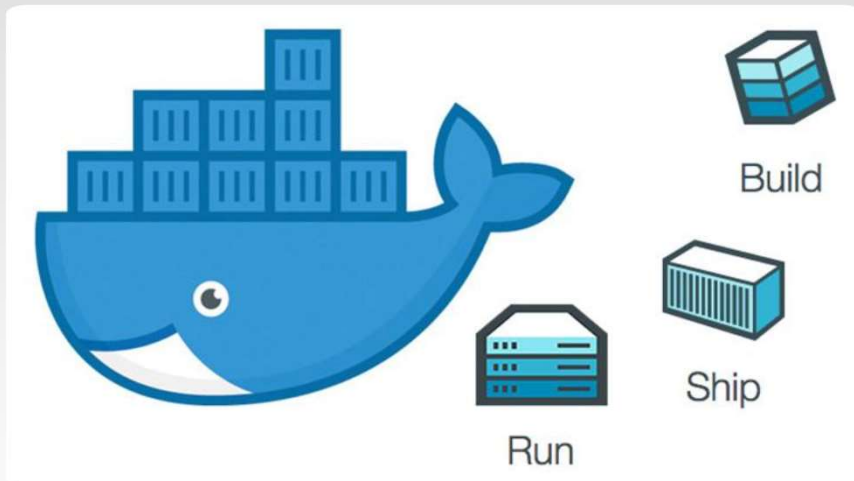


## Containers

- 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.



# What is Docker?



Docker is a open source project. It is 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 on OSs'.



# 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.”

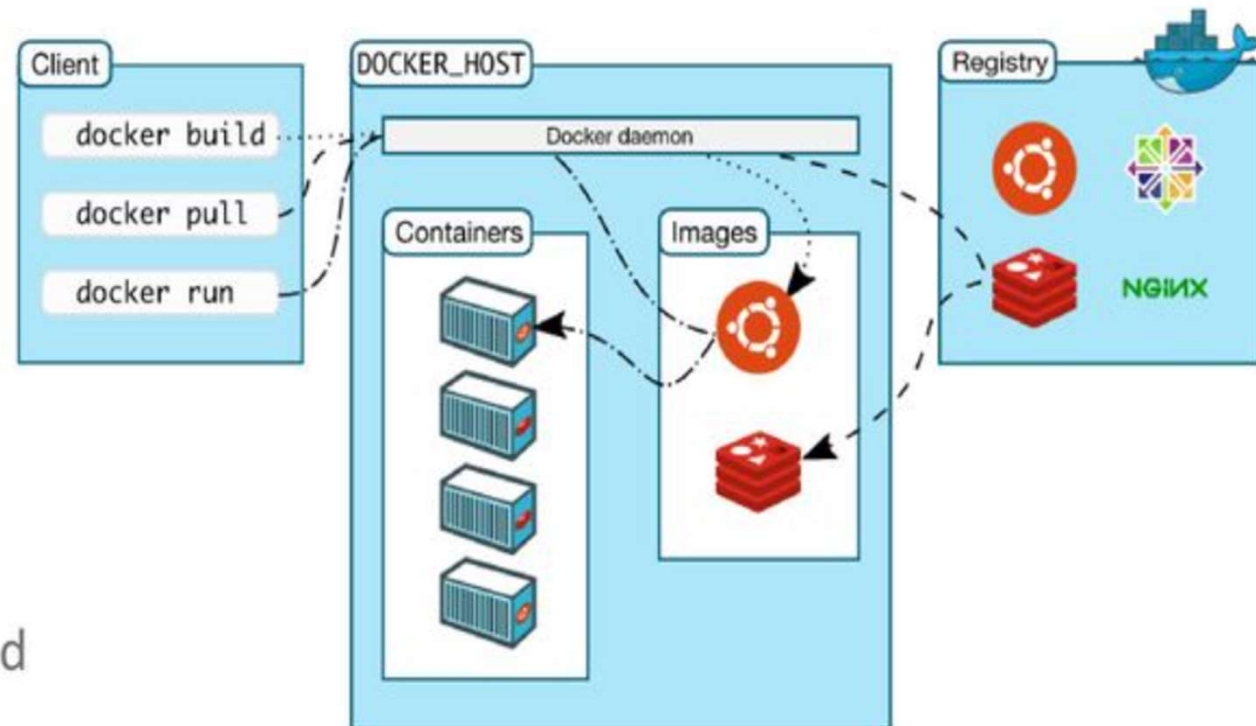
**Speed.** Docker containers are very lightweight and fast. Since containers are just sandboxed environments running on the kernel, they take up fewer resources. You can create and run a Docker container in seconds, compared to VMs which might take longer because they have to boot up a full virtual operating system every time.

**Docker Hub.** Docker users also benefit from the increasingly rich ecosystem of Docker Hub, which 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. With Docker, it’s become easier to link containers together to create your application, 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 IMAGE LAYERS

1. Started Image Layer 3 as a container and accessible by users

1. Started Image Layer v1 as a container.  
2. Installed and Configured https web server.  
3. Committed new layer v2

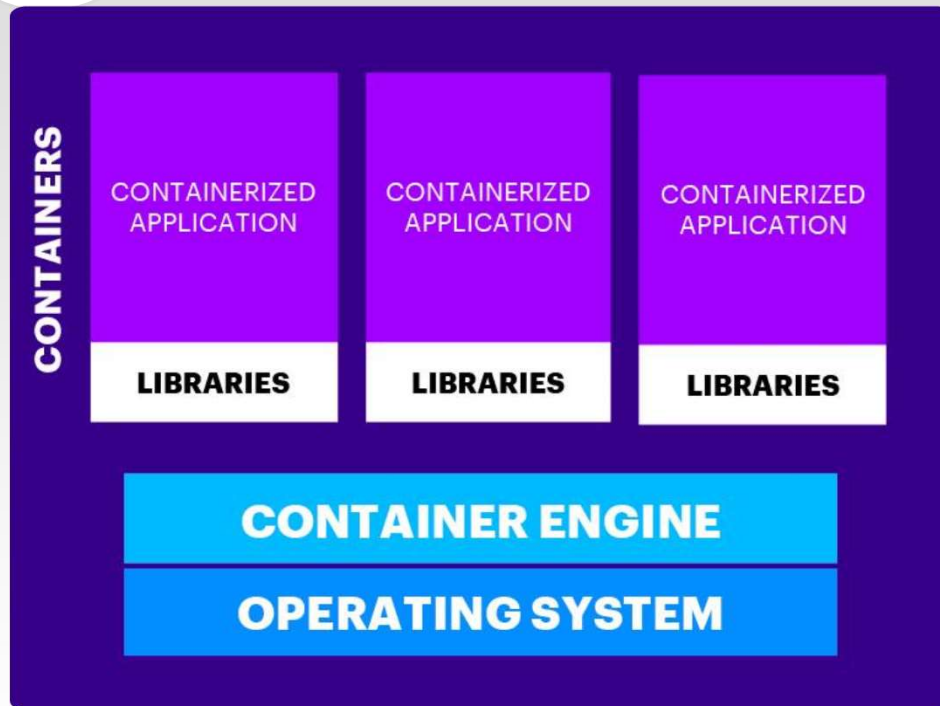
1. Started Base Image (**docker.io/centos**) as a container.  
2. Package Updated on Base Image using "yum update".  
3. Committed new layer v1

Pulled CentOS image from Docker Hub using docker pull command. **Repo: docker.io/centos**





# Docker Engine

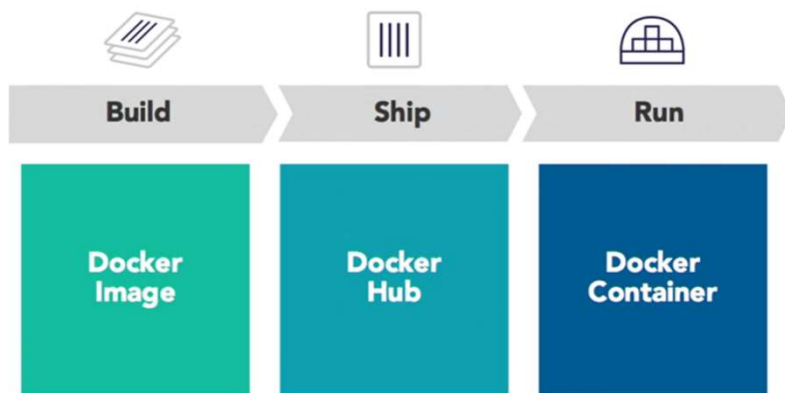


- Container execution and admin
- Uses Linux Kernel
- Linux Namespaces and Control Groups
- Namespaces provide for isolated workspace



# Summary

## Docker Concept



- 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





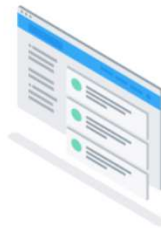
# Docker Environment



## 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

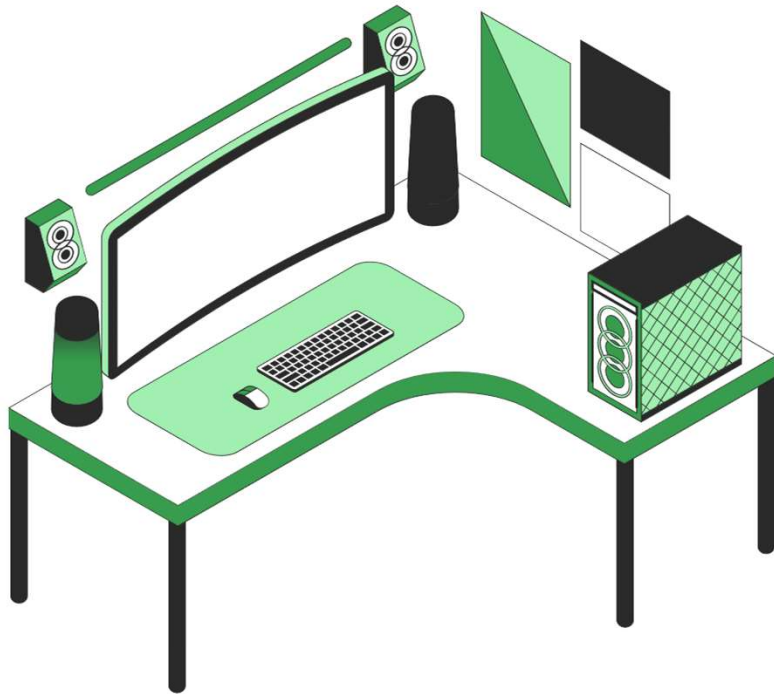


## Play with Docker

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

Play with Docker

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



# Do you have any questions?

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



TECHPROED