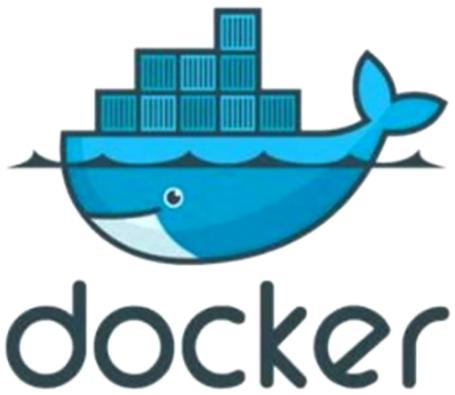


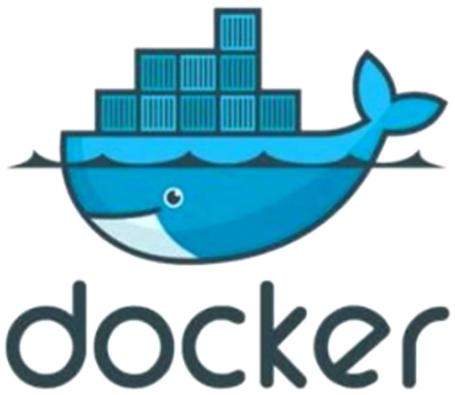
# Docker Infrastructure & Components



# Containerization – Docker

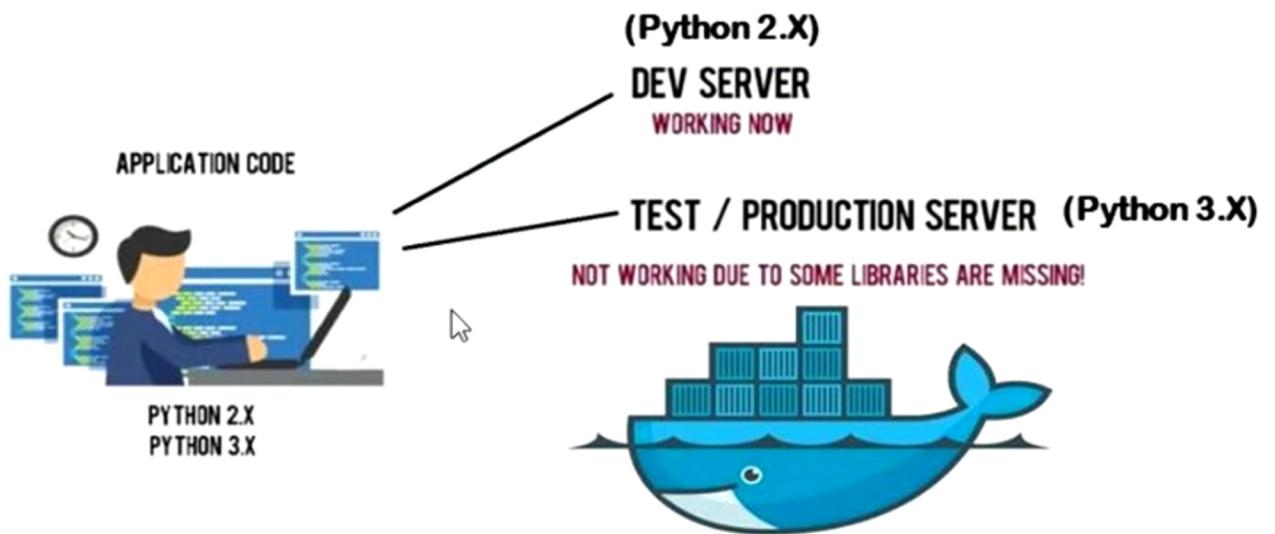
## Topics:

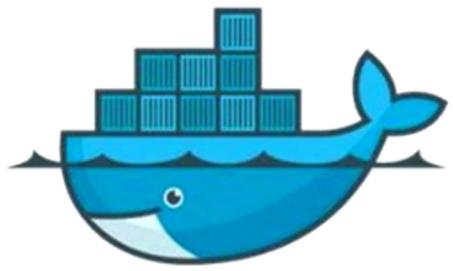
- 1. Introduction to Containers**
- 2. Containerization vs Virtualization**
- 3. Advantages of Containers**
- 4. Introduction to Docker**
- 5. Docker Architecture**



# Containerization – Docker

**Introduction – Why we need  
Containerization (Containers)?**

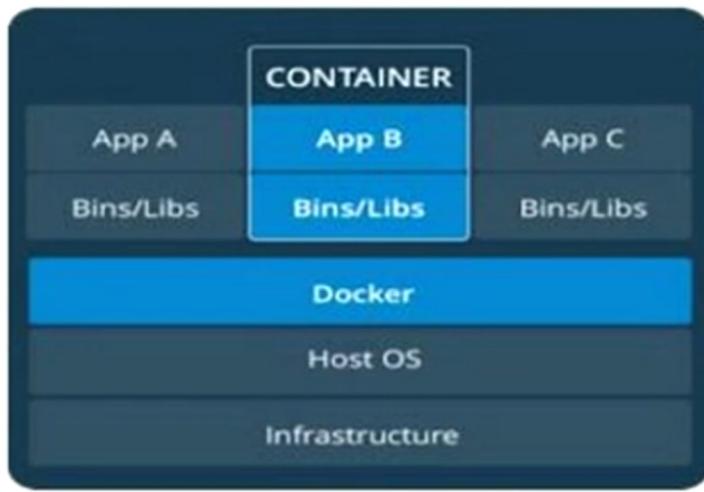




docker

# Containerization – Docker

**What is Containerization (Containers)?**

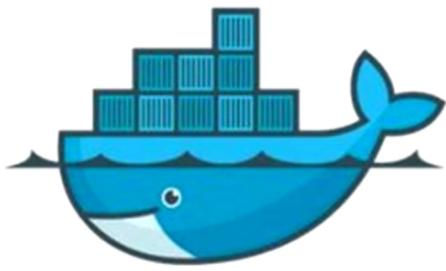


**Containerization is a lightweight virtualization technology alternative to hypervisor virtualization. Any application can be bundled in a container can run without any worries about dependencies, libraries and binaries.**

**Because container creates the isolated environment with all the required dependencies, libraries and binaries to run your application with out any issue.**

**Hence you can build the packages, ship the application to any environment and run it.**

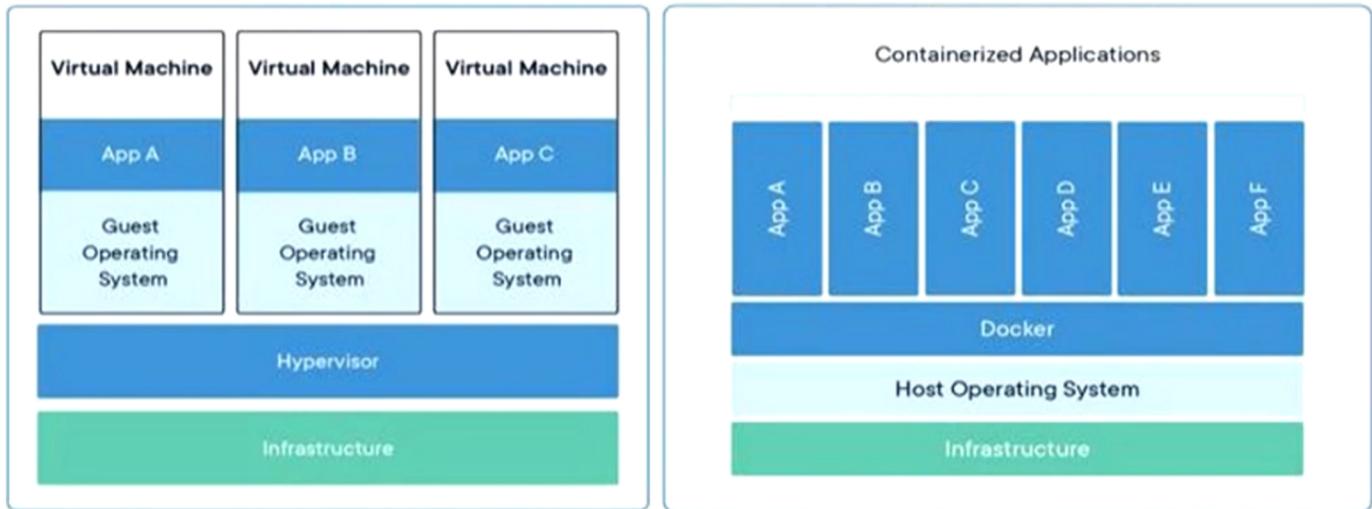
**So Containers are designed to run on physical servers, virtual machines and any cloud instances.**



# docker

## Containerization - Docker

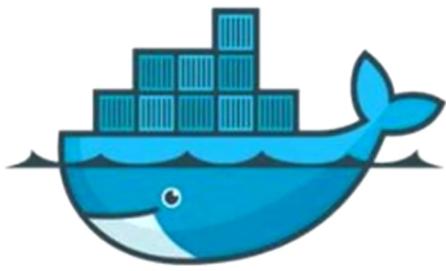
### Difference of Containerization and Virtualization



**Virtualization technology that allow us to have multiple operating systems to share a single hardware processor.**

**Containerization is an application-specific virtualization, because it provides application with dedicated environments in the form of container to run on which can be deployed and run anywhere without a dedicated virtual machine with Operating system for each application.**

**Also Container was designed to solve modern problems and application management issues. so it's not a replacement for virtualization, but it's complementary to it.**



# docker

## Containerization – Docker

### Advantages of Containers

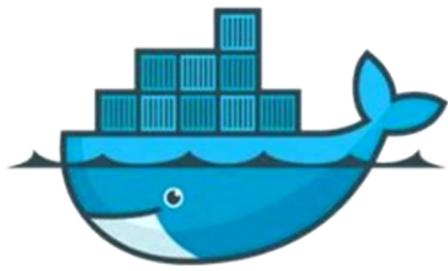
**Containers are isolated, doesn't require operating system and it shares a host kernel. So Containers run on the same server and use the same resources, they do not interact with each other because its isolated. If one application crashes, other containers with the same application will keep running without any issues.**

**It's a Portable and light weight operating system and it contains only the required binaries, dependencies and libraries to run the application. so it can be move anywhere easily and can run without worrying about compatibility, dependencies kind of issues.**

**Faster and Resource Efficiency** – Its Very fast to boot, because containers are lightweight and start in less than a second since they do not require an operating system boot.

**Resource efficiency** Since containers do not require a separate operating system, they use less resources.

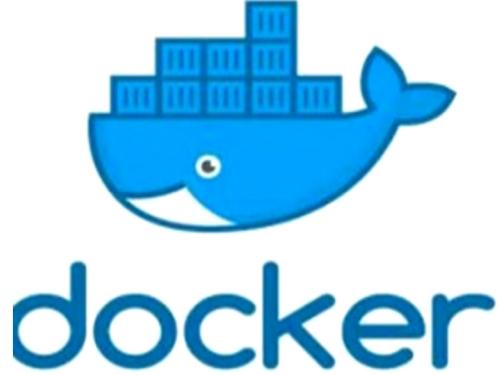
**Improve Scalability and lower costs** – By allowing more containers in the environment without the need for more servers, containerization increases the scalability anywhere from 10 to 100 times that of traditional VM Environments



# docker

## Containerization - Docker

**Introduction to Docker - What is docker  
on Container**



# docker



kubernetes



Azure Kubernetes Service



OPENSIFT

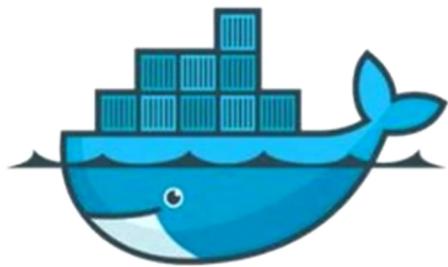


Amazon EKS



Google  
Container Engine

**Docker is a opensource platform tool designed to manage the containers, which allow us to build the application in a container with required libraries, binaries, dependencies to run the application, ship the container and run anywhere.**

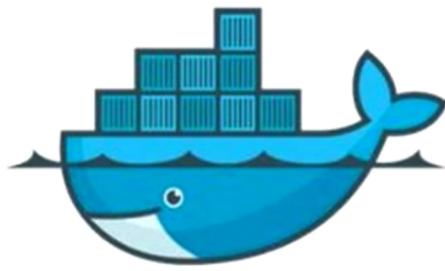


**docker**

# Containerization - Docker

## Introduction to Docker - Why we do use Docker

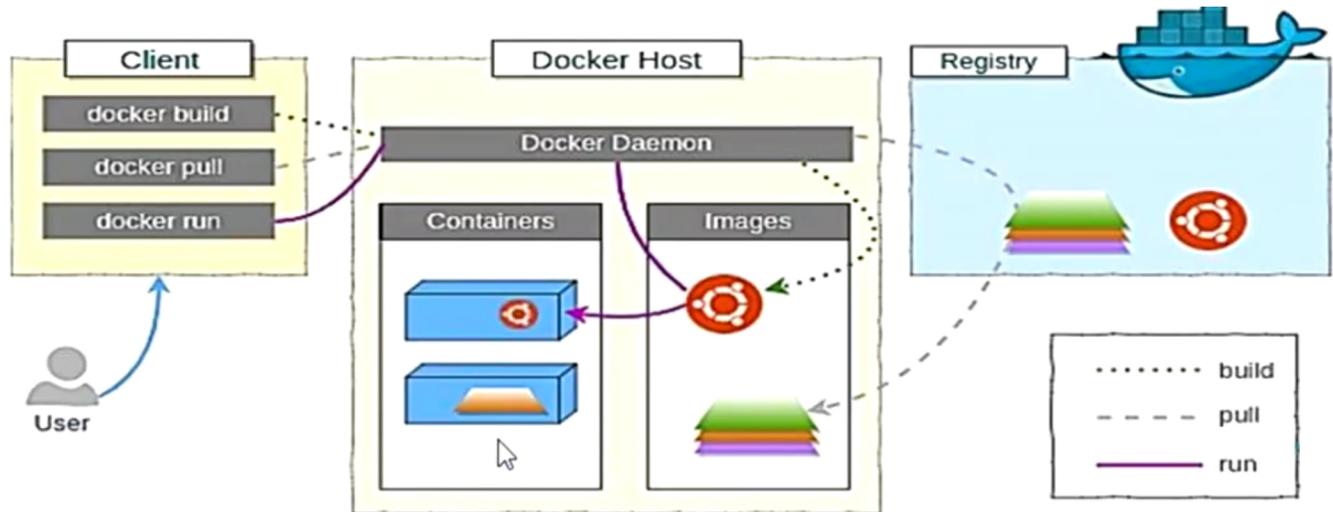
- 1. Portability** - An Applications can be build in to a single unit and same unit can be deployed to various environments such as dev server, testing server and production server without making any changes to the container.
- 2. Light Weight** - Docker containers are pretty lightweight, So it provides a smaller footprint of the operating system via containers.
- 3. Fast Delivery and Scalable** - Since Docker containers are pretty lightweight, So they can be deployed faster and they are very easily scalable.
- 4. Docker used for Continuous Deployment and Testing.** So with the help of containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- 5. Docker also provides you, ability to run multiple isolated OS on single host.**
- 6. Resource Optimization** - Docker enable you to utilize the maximum resources and reduce the resource wastages of your hardware.



# docker

## Containerization - Docker

### Docker Architecture and Components

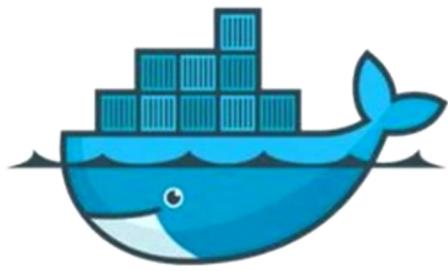


**Docker Client** – It is used to manage our environment,  
Docker client is run not only inside in the Docker Host, It will run in any different (Over the Network) location.

**Docker Host** – Docker Host is the main of the Docker Engine Where we run the Docker Host, there Docker Daemon also run.

If the Docker Daemon is run only means, Docker Images and Containers We easily manage to them.

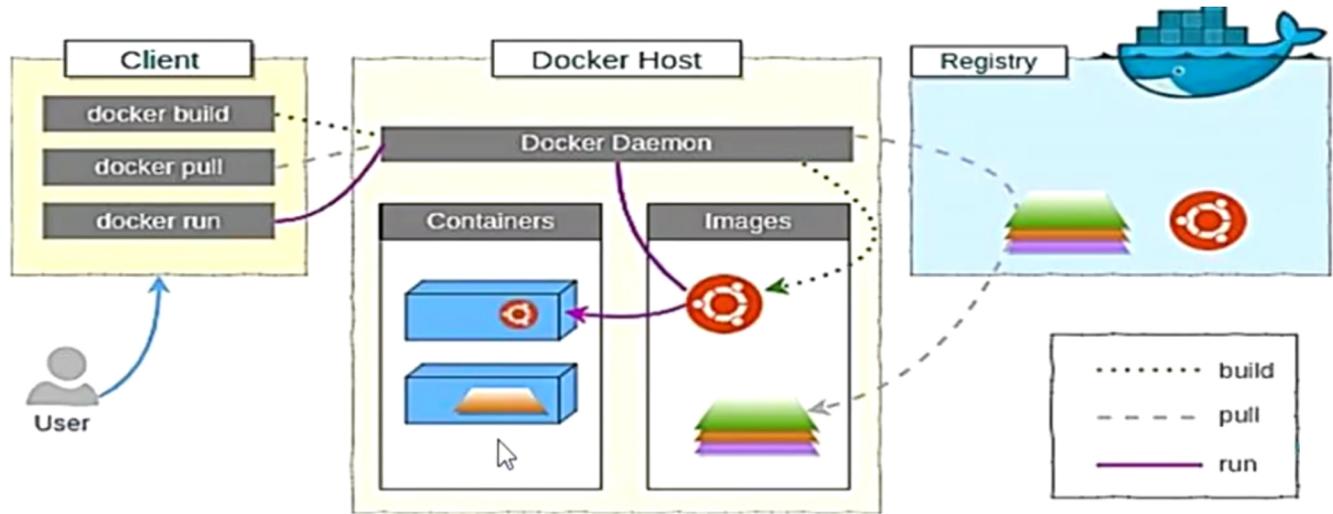
If we want to run the container means, We must need the Docker Images,  
[For ex: VM -> We need OS ISO File]



# docker

## Containerization - Docker

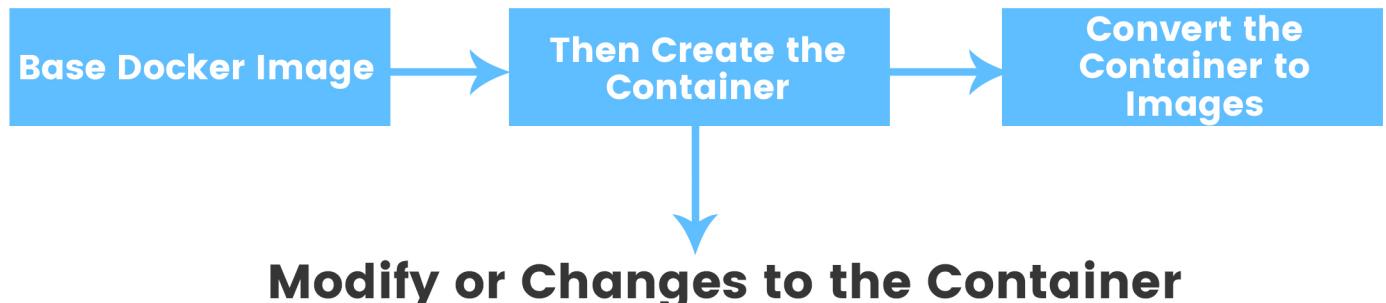
### Docker Architecture and Components



We store the Docker Images in 2 Places

1. Docker Host [local Docker]
2. Public Registry [Docker Hub]

How to create the Docker Images



**Note:** If we need to create the container means we must need Docker Images