Docker Introduction

(Document No - 1)

By Mr.Shiva

What is SDLC?

SDLC(Software Development Life Cycle) is a methodology that industry follows to develop and application.

What is DevOps?

DevOps is the combination of both development and operations.
Development : It doesn't mean application development. Knowing the functionality of the application and developing the scripts to create infrastructure(servers).
Operations : Maintaining the applications and infrastructure(eg: Providing the security updates.)

Docker and Kubernetes position in SDLC?

Docker and Kubernetes fit in the deployment phase in the SDLC process.

Importance of Docker and Kubernetes in IT industry

Because of the evolution of microservice docker and kubernetes has great importance in the industry. Most of the companies follow the docker way of deploying applications on the servers.

Importance of DevOps in SDLC?

DevOps plays a major role in the SDLC process. Below is its importance in the SDLC process.

- It provides a way to automate the deployment of infrastructure and applications on the servers.

- DevOps helps to deliver the software products to the customers faster and with high performance and reliability.

- DevOps helps us to solve the issues faster and reduces complexity.

- DevOps provides scalable and disaster recoverable infrastructure.

- Better resource utilization.

What are Physical Servers?

A server with the motherboard, CPU, Hard Drive, Network Connection and Operating System for running programs and applications.

In simple terms, The server which we can touch and interact with is called a physical server.

Problems that we have when deploying applications on physical servers.

- Resource wastage.
- High cost.
- High maintenance.

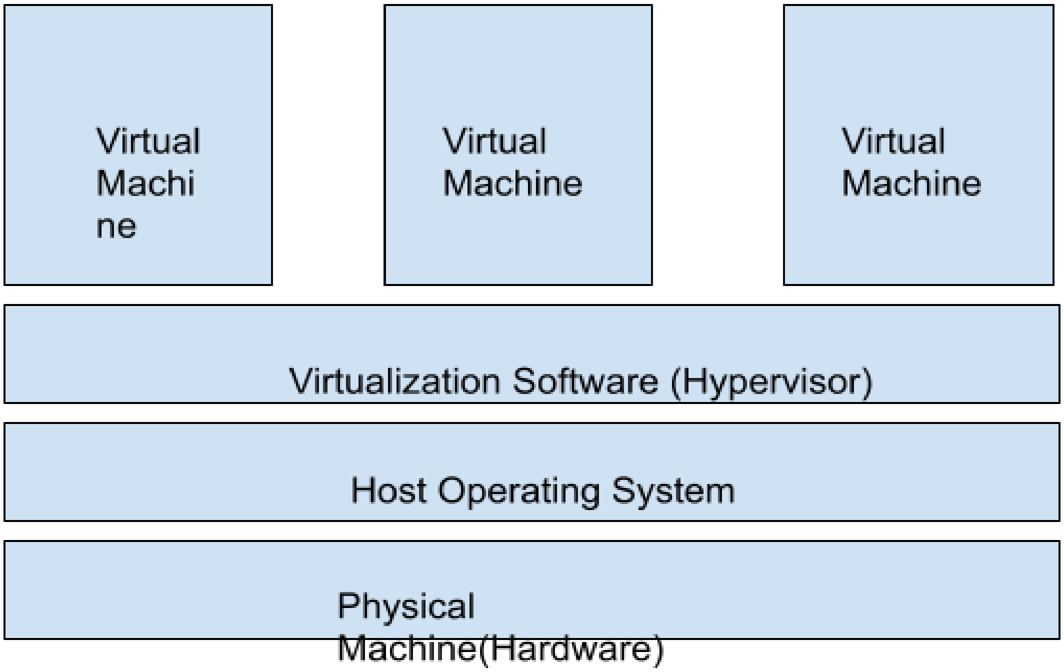
What is virtualization?

Creating multiple virtual machines on top physical machines is called Virtualization.

We have a bare metal(hardware or physical machine) on top of which there will be an OS(Operating System). On this OS there will be virtualization software(eg: hypervisor). On this we will create virtual machines/servers.

What is Virtual Server?

A virtual server recreates the functionality of the physical server.
In simple terms, The servers which we can only interact with are called virtual servers.
We can create multiple virtual servers in one physical



We will install applications on virtual machines.

Problems we have in installing applications on virtual machines:

1) Resource wastage.
2) Increasing cost.
3) Applications have to pass through many layers in order to access hardware resources.

Because of these problems containerization has evolved.

Containerization

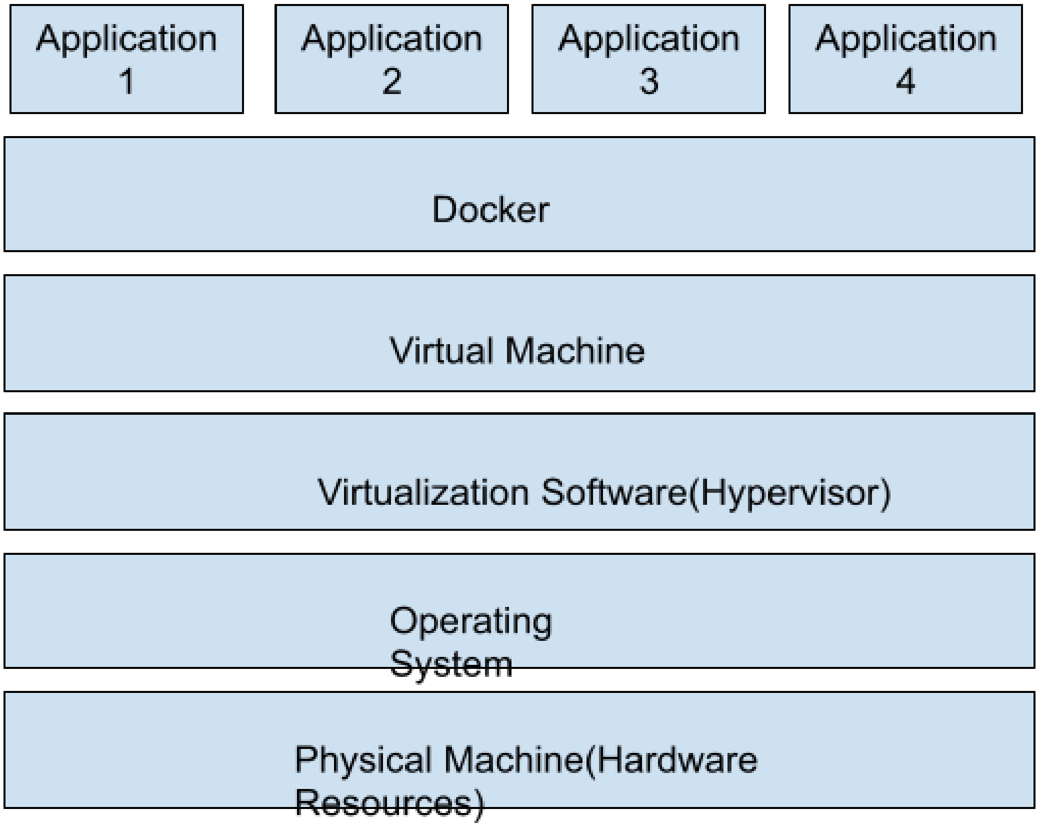
Here we have a physical machine on which we have an operating system. We will be having virtualization software. Upon which there will be virtual machines.

We will install containerization software(docker) on virtual machines. On top of docker we will run applications as containers.

There are many containerization technologies available in the market. Popular one is Docker. Containerd is the other one.

Docker

Docker performs process isolation i.e., it removes the dependencies of an application on an host OS and it allows that application to run directly on docker engine.



Docker comes with two types.

a) Community Addition.
b) Enterprise Addition.

Docker can be installed in multiple operating systems. Below is the link that will navigate you to the docker installation page.

https://docs.docker.com/engine/install/

As we are using a ubuntu machine to install docker, Below are the commands that are used to install docker on ubuntu.

```
sudo apt update -y
sudo apt install apt-transport-https ca-certificates curl
software-properties-common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable" -y
sudo apt update -y
apt-cache policy docker-ce
sudo apt install docker-ce -y
sudo systemctl status docker
sudo chmod 777 /var/run/docker.sock
```

Docker images and containers

A docker image is a collection of binaries and libraries which are necessary for software applications to run.

Running instance of an image is called a container. We can run any number of containers from one docker image.

Technical terms that were used in the docker course

Docker Host

The machine on which docker is installed is called docker host. It can be Windows, Linux or Mac.

Docker Client

This is an application which is part of the docker engine which is responsible for accepting docker commands from the user and passing it to docker daemon.

Docker Daemon

This is the background process which is also part of the docker engine. Responsibility of docker daemon is to accept the commands from the docker client and perform necessary action.

Docker Registry

The place where we store docker images is docker registry. These are two types:

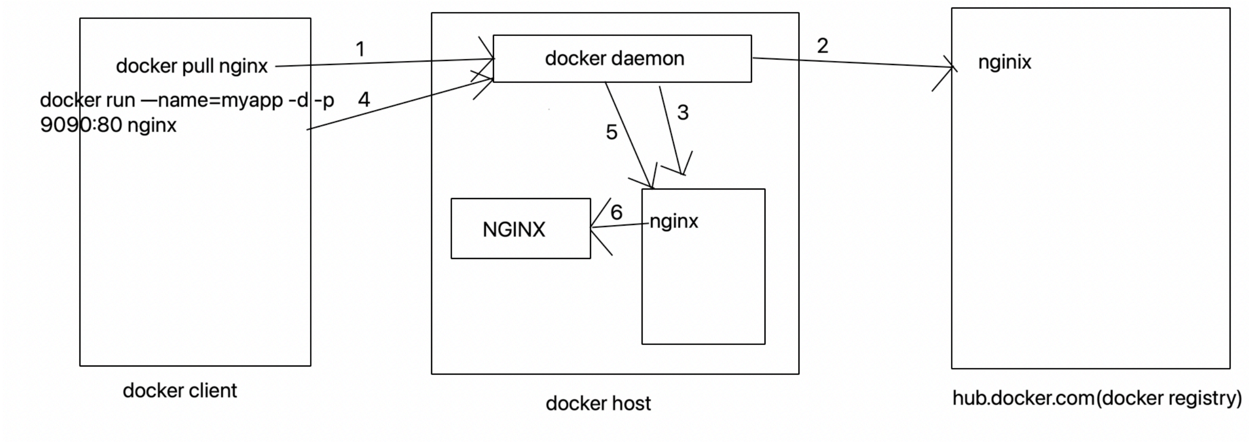
1) Public Docker Registry.

2) Private Docker Registry.

Public is the hub.docker.com images uploaded here will be accessed by anyone.
Private docker registry is private to a particular organization or particular team. Only they can access the images in these registries.

Docker Architecture

Here is the way that docker functions when we issue a command from docker client.



Below were the steps in the above architecture

Step1: We execute the command from the docker client(i.e docker pull nginx). This command will be listened to by docker daemon.

Step2: Docker daemon will be connecting to the docker registry and perform action.

Step3: Docker daemon will be downloading nginx image from docker registry.

Step 4: Here we have executed a command to run a container(docker run –name=myapp -d -p 9090:80 nginx) that will be listened to by docker daemon.

Step5, Step6: In this step docker daemon will be using an already downloaded image to run the application.

—-------Continuation in next document —--------