**Docker Tutorial**

## Docker Introduction:

### What is Docker?

A platform building, running and shipping applications

### What is the issue before Docker

Once the developer developed the code, he will hand over it to other team like testing, deploy for them to complete their works. The problem faced here is software worked very well in developer’s system, but not working in other’s system.

### The various reasons are

1. One or more files are missing
2. Software version mismatch
3. Different configuration settings

This is where docker comes in to picture. In docker, you can package the application with all required software’s and deliver it to others team. Now there is no way for the application not to run.

### Container vs Virtual Machine



### Hypervisor

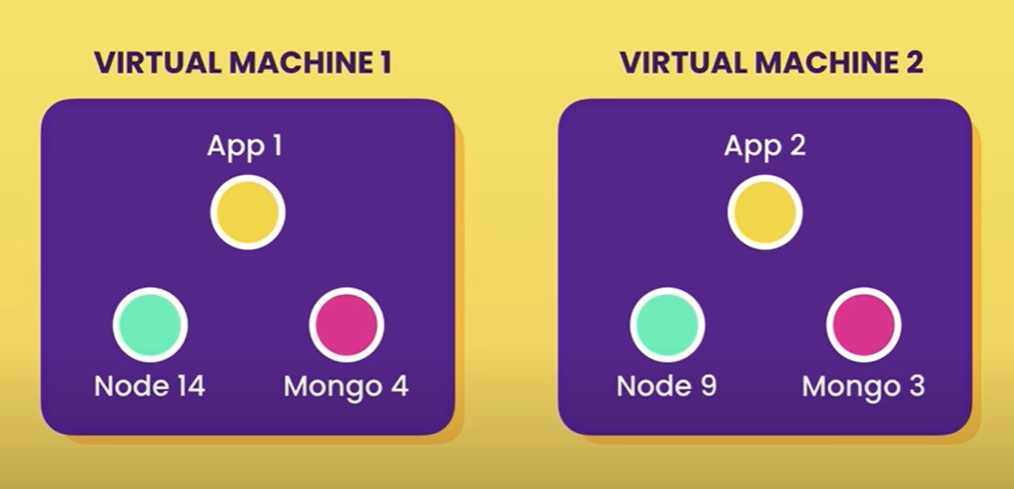
In a mac machine, we can create two virtual machine one containing Window and another containing Linux. This can be done with the help of Hypervisor



### Different Hypervisor

1. VirtualBox
2. VMWare
3. Hyper-v (Windows only)

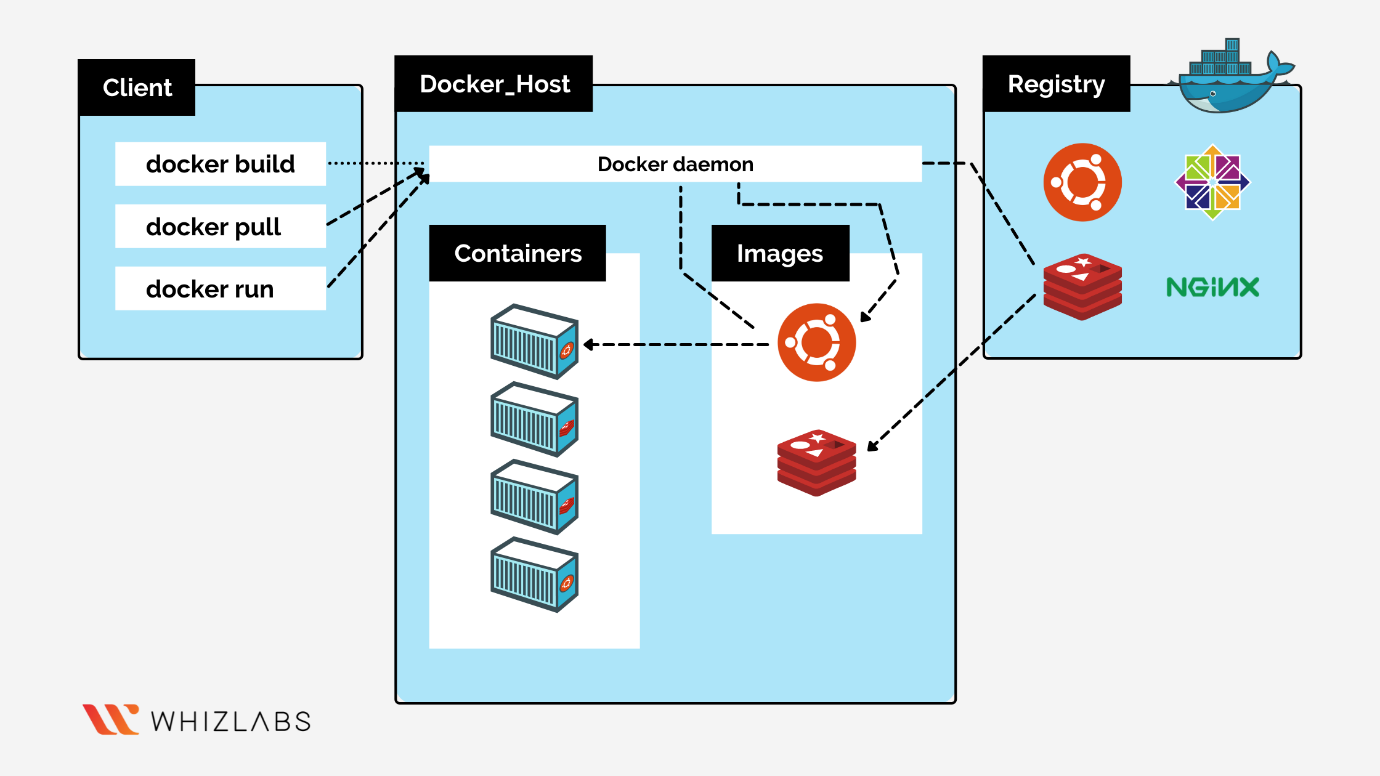
### Advantages of Virtual Machine



### Problems in Virtual Machine

1. Each VM needs a full-blown OS
2. Slow to start
3. Resource intensive

## Docker Architecture



## C**ontainer**

The running version of image is called Container. The Container address the problems in Virtual Machine. The advantages of Containers are

1. Allow running multiple apps in isolation
2. Are lightweight
3. Use the OS of host
4. Start quickly
5. Need less hardware resources

## Container Architecture

## Installing Docker

## Development workflow

Code is developed by Dev Team. Then they create image and put it in Docker Registry. Now any team (Test / Prod) can access this image and run the application. It would run same as how it happened in Dev.

## Docker in Action

## Docker Commands

### Docker Image Commands

| **Purpose** | **Docker Commands** |
| --- | --- |
| Build Image | docker build -t login-name/hello-world-python:0.0.1.Release . |
| Run Image | docker run -p 5000:5000 hello-world-java:0.0.1.Release  -p - publish  5000:5000 – mapping the container port to host port to access the application in localhost  Hello-world-java – Repository  0.0.1.Release – version of specific repository, image file which contains all the files needed, software versions, all the libraries and dependency files.  -d – detached mode (optional) |
| To see console | docker log -f |
| List all images | docker images |
| Push images to docker hub | docker push senthilnata1/hello-world-python:0.0.1.Release |
| Pull Image | docker pull <any image from docker hub> |
| Search image | docker search <any image> |
| History | docker image history rbainfotech/hello-world-java:0.0.1.Release |
| Version | docker –version |
| Terminate Container | Ctrl + C |

### Docker Container Commands

| List all containers | docker ps  docker ps -a |
| --- | --- |
| Stop Container | docker stop <container id> |