Why we need docker?

What is docker?

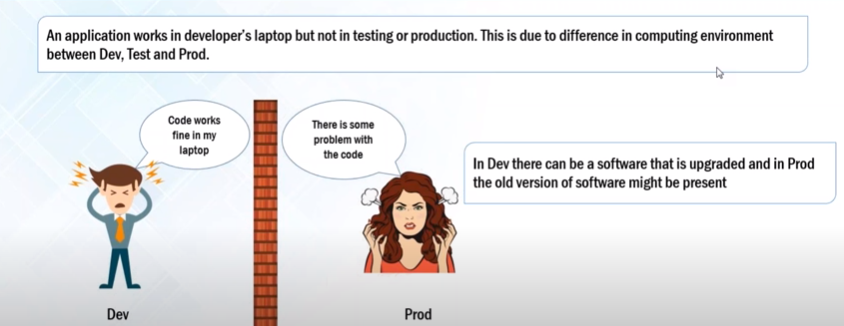
Components of docker

Docker example

Day 1:

**Problems before Docker.**

1.

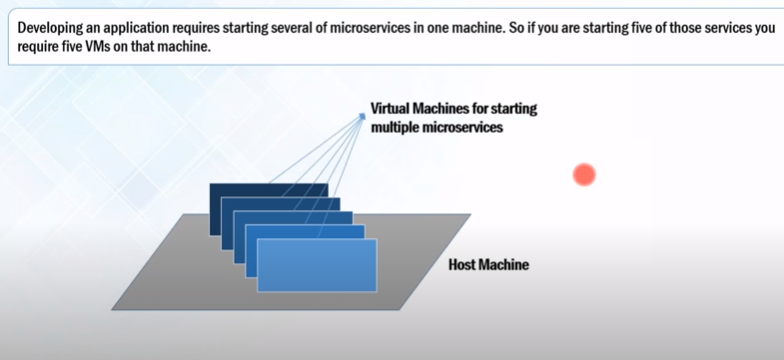


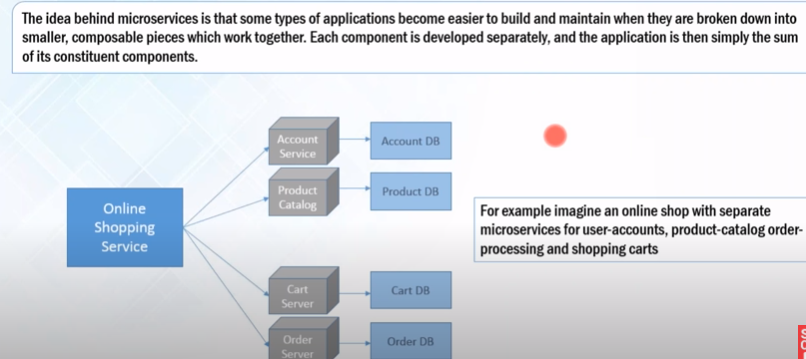
2. fb ,amazon are adpting MS architecture

1.loosely coupled

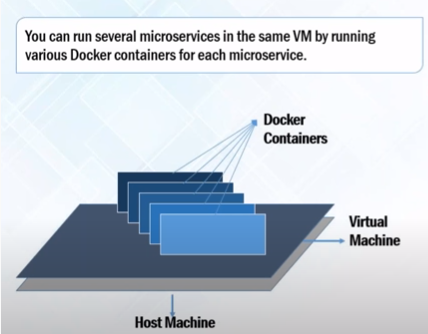
2.modular approach

3.easy to maintain and test



Each virtual machine containing the dependencies for MS .In VM there is a lot of wastage of resources such as RAM, processor ,disk space are not utilized completely by the MS which are running on the VM.

Consider that we have 50 MS then we have to use 50 VM and using this many VM doesn’t make any sense becoz of the wastage of resources.  
how we can rsolve this.



1 host mc🡪1 VM🡪docker container containing dependencies for 1 MS

Docker containers are lightweight version of VM (no need to pre allocate any RAM or disk space it will take RAM and disk space according to the requirement.)

Docker container works on linux/unix OS . and help to isolate the usage of VM form the machine use.

Resource and Memory Utilization

======================================

Host os 16 GB Ram

==================

In Case of virtual Macine

=============================

Allocation => VMs => Memory Used => Memory Un used

6 GB => VM 1 => 4 GB => 2 GB

4 GB => VM 2 => 1 GB => 3 GB

6 GB => VM 3 => 2 GB => 2 GB

==========================================================

16 GB 9 GB 7 GB

7 GB RAM is blocked

7 GB RAM cannot be alloted to new vm.

Host os 16 GB Ram

==================

In Case of Docker

=============================

Allocation => Docker => Memory Used

=> Container1 => 4 GB

=> Container2 => 1 GB

=> Container3 => 2 GB

==========================================================

16 GB 9 GB 7 GB

No Ram is allocated for unused

7 GB RAM is unused

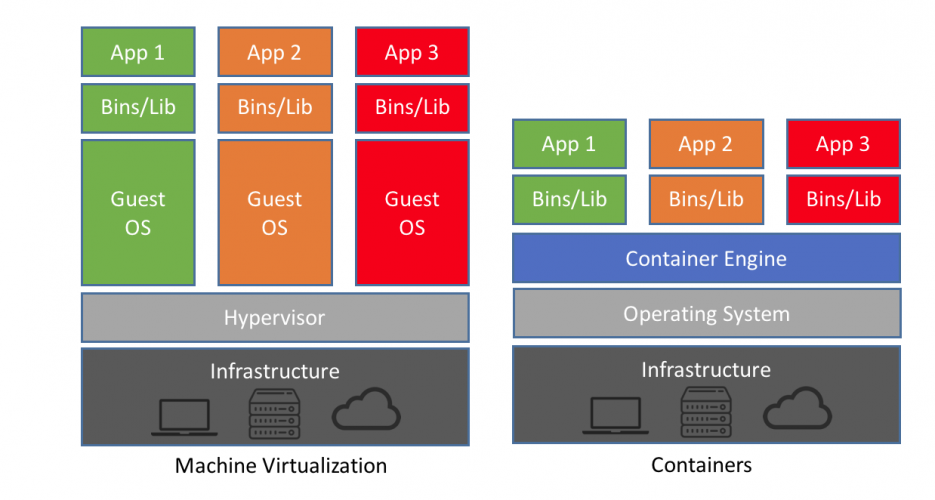
7 GB RAM can be alloted to new containerization.

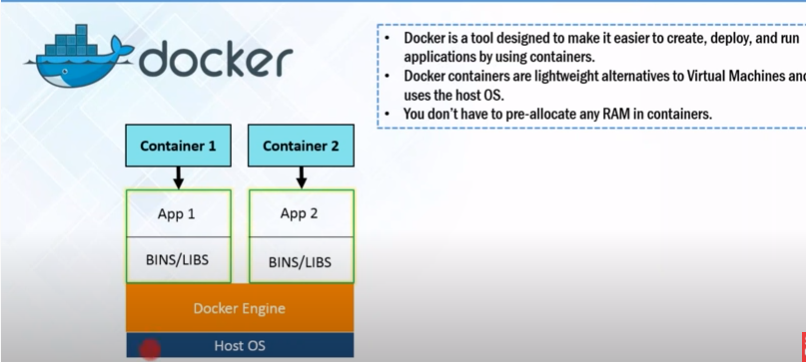
How docker solves the issue of consistence computing environment.

Docker container is created by developer. Same environment is available in dev as well as in operation environment and providing consistant computing enviroment.

.

Docker doesn’t use guest OS it uses the host os





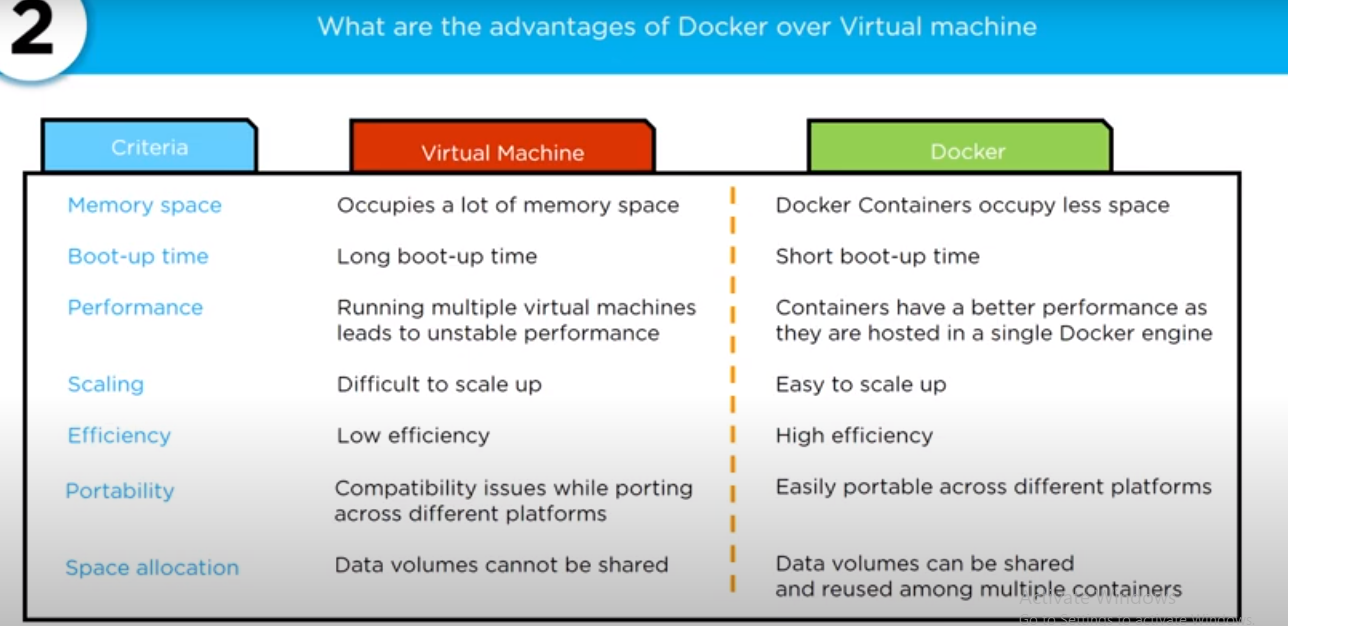
* Docker doesn’t use guest OS it uses the host os
* On the top of host OS docker engine is there . with the help of docker engine , docker containers are created
* Containers have applications running in them
* Requirement for the application such as binaries/libraries are also packaged in the same container.

Container Engine

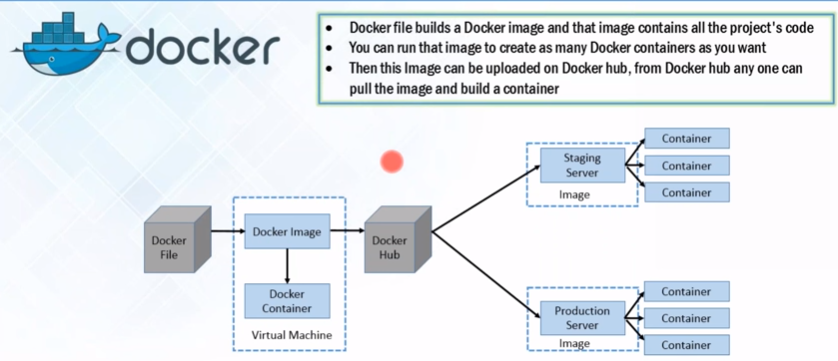
1.Docker

2.Container D

3.RKT

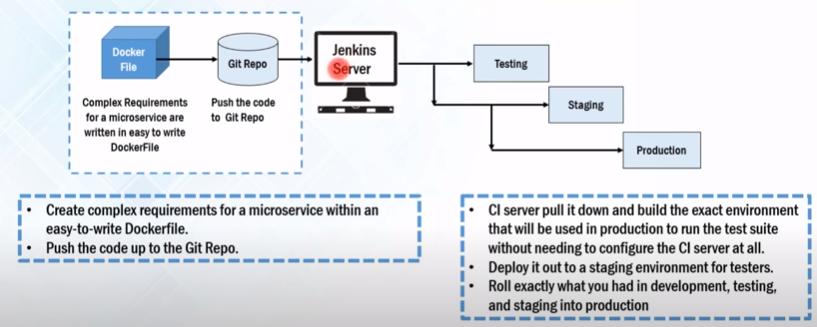
* 

**WorkFlow of docker:**

****

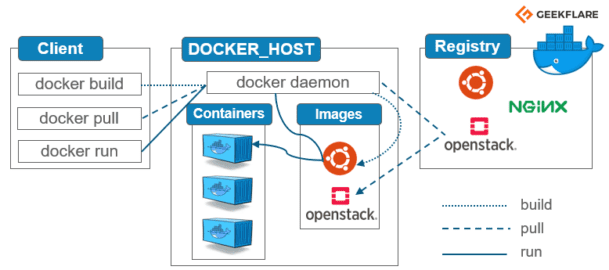
* Devloper defines a code that specifies the dependencies and requirement in a docker file.
* Docker file produces docker images(dependencies +requirements)
* When we run the docker image it create an instance ie, Docker container
* Image uploaded on docker hub (git repo) which is a git repository for docker image both public and private repo
* From hub various team such as QA or prod team will pull the image and prepare their own container.
* Same environment through out the sdlc life cycle ie on dev,test an dprod server.
* Docker hub is a cloud hosted service provided by docker. Where we can push and pull the images.

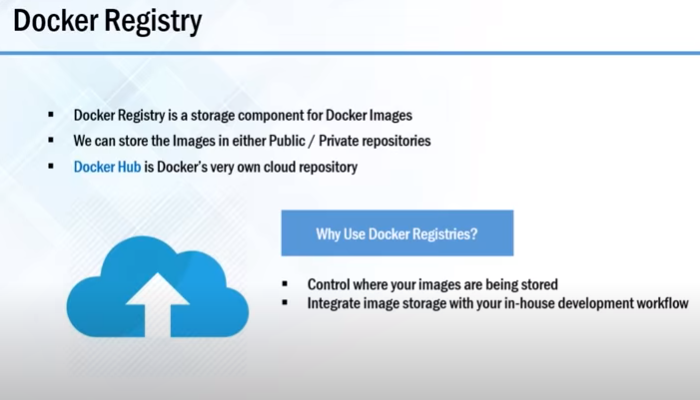
Example



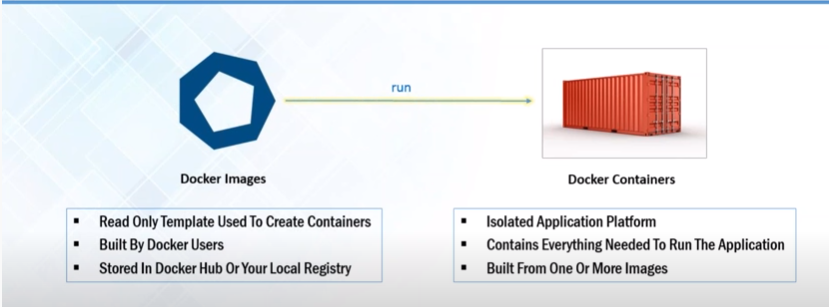
1. Docker File –Push on git repository
2. From git rep container integration service like Jenkins will pull the code and build the environment that contains all the dependencies for the MS which can be deployed on testing,statging and production environment.

**Docker Components:**



****

**Docker Images & Containers**

****

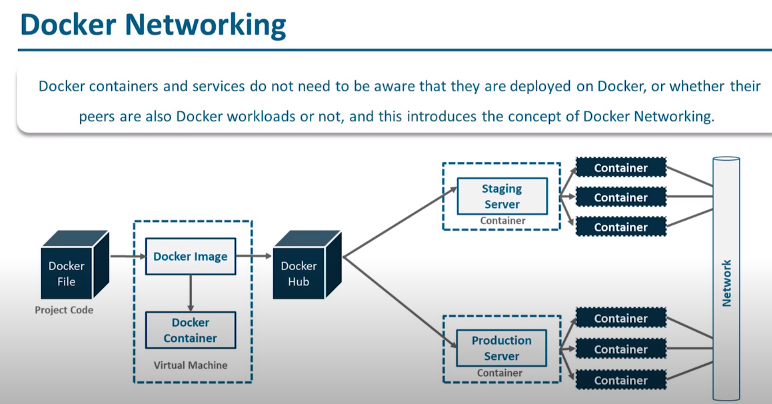
**Day 2:**

**Docker Networking:**

**Orchestration** is the automated configuration, coordination, and management of computer systems and software

**Multipler containers run on same machine and sharing the OS kernel with other containers each running as an isolated process.how this containers communicate with each other? Ie. Docker networking.**

**Docker workflow**

****

* Devloper defines a code that specifies the dependencies and requirement in a docker file.
* Docker file produces docker images(dependencies +requirements)
* When we run the docker image it create an instance ie, Docker container
* Image uploaded on docker hub (git repo) which is a git repository for docker image both public and private repo
* From hub various team such as QA or prod team will pull the image and prepare their own container.
* This various containers communicate with each other over a network

<https://www.edureka.co/blog/docker-networking/>

<https://www.youtube.com/watch?v=cMj_3woGXhM>