Unqu' Shed fy Number.

DOCKER

Need of docker :
To overcome the limitation of resources faced in virtualization
Docker is a tool used for containerization.

- Open source centralized platform design to create, deploy and run applications

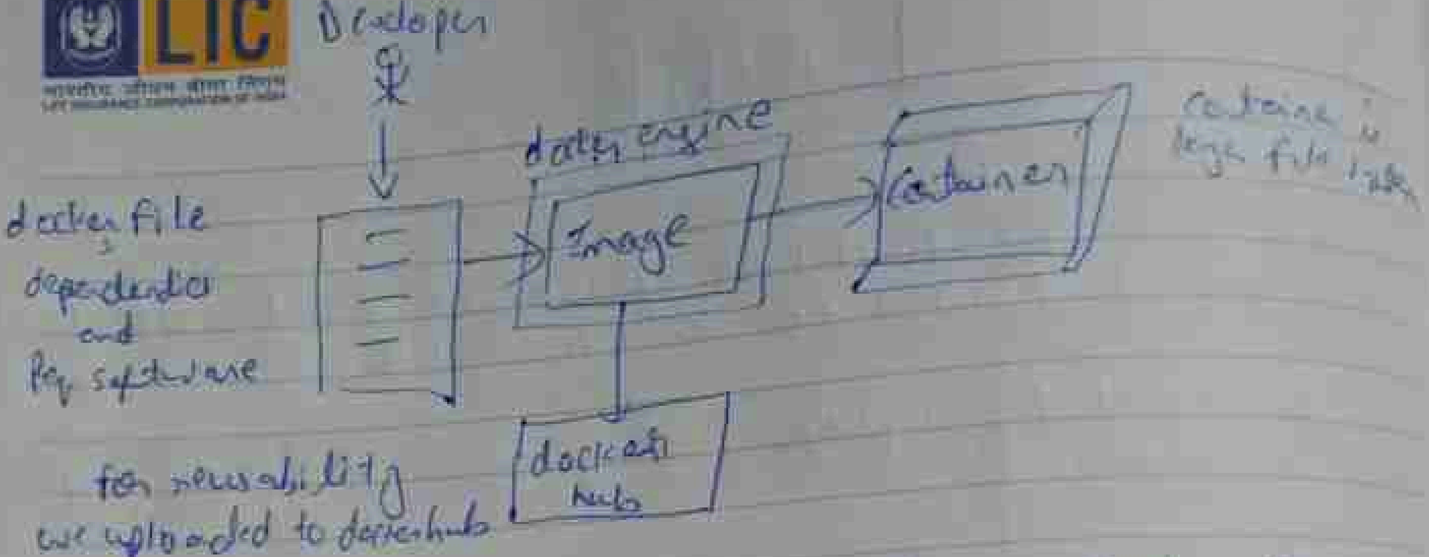
- Docker use containers on the host O.S to run applications.
- Docker ~~xxx~~ engine runs natively on linux distribution.
- it perform OS level virtualization known as containerization
- Docker is a PAAS
- Docker use OS level virtualization while VM ware use hardware level virtualization.

Advantages,
NO pre allocation of RAM
CI efficiency
less cost, light in weight
docker can run on physical H/w / virtual H/w / Cloud

[We cannot modify images but can change Container
Container is running instance of Image.]

Limitation of Docker
- do not support cross platform
- not a good soln for app that require rich GUI
- Testing and development should be same (ubuntu = ubuntu and not/ubuntu ≠ centos)

Docker Architechture



container is large file [illegible]

docker file
dependencies and Req software

for reusability we uploaded to dockerhub

Docker is a ecosystem because docker containers requires a lot of thing.

Docker ecosystem contents :—

1) Doc Client – where we create one docker file
2) doc daemon/ server – where doc engine resides, it is responsible for creating an image into container
3) doc hub – It is registry
4) doc image – template
5) doc Compose – To run multiple docker.

doc engine = doc daemon

* ways to create an image
a) image from docker hub
b) create image from docker file
c) create image from existing docker container

Container hold the entire package that is needed to run the application.

Basic Commands in Docker

Ps = Process status
CNTR = Container

```
yum install docker
1) docker images                                   (To see present images)
2) docker search <jenkins/chef/nginx>              (To search)
3) docker pull <jenkins/chef/etc>                  (To download)
4) docker run -it --name akbar ubuntu /bin/bash    (To give name to Container)
5) service docker status  or docker info           (To check)
6) docker start <name of container>                (To start)
7) docker attach <name of container>               (To enter a container)
8) docker ps -a                                    (To see all containers)
9) docker ps                                       (To see only running containers)
10) docker stop <name of container>
11) docker rm <name of container>                  (remove)
which docker | docker -v | docker --version
12) cat /etc/os-release                            (To check OS version in container)
```

Dockerfile Components

Steps: ~~Create a container~~
: ~~update the data make them into image~~
: ~~new/[illegible]~~

To find differences,
docker diff <container name>

To create container of image,
docker commit new container update image
name of container name of image

DOCKERFILE

- A Text file which contains set of instruction
- Automation of docker image creation

COMPONENTS

FROM - Top command, it tells base image
RUN - To execute
MAINTAINER - Author/owner
COPY - copy from local host
ADD - copy from internet also extract
axPOSE - To expose port number
WORKDIR - To set working directory for container
CMD - execute command during creation of container
ENTRYPOINT - high priority than CMD
ENV - Environment variable.
ARG -

To create image ~~to~~ from Dockerfile

```
docker build -t myimage .
```