**18/03/2025**

**Docker**

* Docker is a containerization tool
* Virtualization
* Docker is a server level virtualization tool
* Docker is an open source containerization platform by which you can pack your application and all its dependencies into a standardized unit called container
* Docker is a client server architecture
* Developer will write the docker file not only developers, operations team can also write the docker file

**Creating the docker image is done in 2 ways**

* From docker file
* Using docker commands

Docker image will be stored in docker containers

Containers will stored in docker hub

Docker images will be pulled into multiple environments and using one container will be created

For n number of images having the n number of containers

**Docker Hub:**

Docker hub is used to take the backup of both images and containers

From here we will start all the environments

**VMware:** VMware used to operate multiple operating systems like windows, Linux, ubuntu

VMware as virtualizing the entire hardware in your pc/laptop

VMware is called as hypervisor

Docker overcomes the drawbacks of the VMware

**Container:**

Container is one Operating system

Container is one server

Container is one process

We can create n number of the containers on hardware of the system

A container is like a small, portable box that holds everything on application needs to run. This includes

* The app itself
* All its dependencies (libraries, tools, and configurations)

Docker remove the gap between developers and operations

**VMware:**

VMware uses full virtualization, which virtualizes at the hardware level. Each virtual machine runs its own OS with a dedicated portion of hardware resource

Each Virtual machine has its own OS, requiring more RA, CPU and storage

VMware provides full OS-level isolation, making it more secure but at the cost of higher resource usage

VMware less portable since VMs depends on the underlying hypervisor and may require adjustments when migrating across platforms

VMware suitable for running multiple Oses, legacy applications

VMware uses virtual switches, shared storage, and traditional disk management for VMs

**Docker:**

Docker uses containerization, which virtualizes at the operating system level. Containers share the host OS kernel and are lightweight

Docker is lightweight because containers share the OS kernel, leading to faster startup times and less resource consumption

Docker provides process level isolation but shares the OS kernel, making it less secure if a container is compromised

Docker highly portable, containers can run on any system with docker installed, making it ideal for microservice and cloud deployments

Docker ideal for micro services, cloud native applications, and CI/CD pipelines where lightweight, scalable environments are needed

Docker uses overlay networks and volumes for container communication and storage

**Steps to create docker Hub**

Step1: Install docker

Step2: create a docker hub account

Step3: create 2 repositories in docker hub 1. Dev 2. qa

Step4: create one instance in aws console naming as docker and connect with git bash

Step5: run sudo su command 🡪to switch to root user

Step6: install docker with command **yum install -y docker**

Step7: docker --version 🡪 to check the version

Step8: ifconfig -a: no info

Step9: service docker start

Step10:ifconfig -a: it shows some info

Step11: change the directory to docker

Docker home directory is /var/lib/docker

Step12:docker images or docker image ls 🡪this command is used to see how many images are created

**Parameters of docker image:**

**REPOSITORY TAG IMAGE ID CREATED SIZE**

Step13: docker ps: to see containers that are currently running

Step14: docker ps -a: to see all the running and not running containers

Step15: pull image

* Pulling a docker image from online(official website)
* By writing the docker file
* I am pulling the docker image from online
* To pull an image called-nginx(web server)
* Docker pull nginx: to pull an image
* Docker run -itd --name name -p <cid:apn> imageID: to create a container for the image
* Docker login: to login into docker hub
* Docker tag nginx:latest dockerrepo:jen(image name)
* Docker push dockerrepo:jen: to push into docker repo
* Docker exec -it container\_id 🡪 to change docker directory to container