**Docker:** docker is a containerization platform that allows developers to package applications and their dependencies into lightweight, portable containers that can run consistently across different environment.

**Why is docker used**

* The application run the same way in development, testing, and production environments
* Each container runs independently, preventing conflicts between dependencies or applications.

**Key docker components in DevOps**

1. **Docker Engine**

The core component that runs and manages containers

1. **Docker Images**

A **blueprint** for containers that includes everything needed to run an application

1. **Docker Containers**

A running instance of a docker image

1. **Docker file**

A script that defines how to build a docker image

1. **Docker Compose**

A tool to define and run multi-container applications using a docker-compose.yml file

1. **Docker Hub**

A cloud-based registry where you can store and share Docker image

**Use of dockers**

Docker is widely used in DevOps, cloud computing, and software development because it simplifies the process of building, shipping, and running applications across different environments

* Docker packages applications and their dependences into containers, ensuring that they run the same way on the system
* Docker is widely used to deploy microservices, where each service runs in its own container
* Developers use Docker to set up development environments quickly, reducing the time spent on configurating dependencies
* Docker ensure that applications run in isolated environments, preventing dependency conflicts

**Install Docker**

**Step1:** create a docker hub account

**Step2:** create 2 repository in docker hub 1.dev & 2.qa

**Step3:** create 1 instance in aws console naming as docker & connect with it

**Step4:** switch to root user

Sudo su

**Step5:** To check the detailed information about all network interfaces, whether they are active or inactive

ifconfig -a;

**step4:** install docker with command

yum install -y docker

**step5:** check the version of dockers

docker –version

**Main drawback of VMware**

* VMware runs a full OS for each VM, consuming more CPU, RAM, and disk space
* Running multiple VMs on a single machine slows down performance
* VMware takes 1-2 minutes to start
* Moving a VM form one server to another can be complex

**Difference between Docker and VMware**

|  |  |
| --- | --- |
| **Docker** | **VMware** |
| Resource usage low | Resources usage high |
| Docker takes seconds to start | VMware takes 1-2 minutes to start |
| Runs anywhere | Hard to move |
| Easily scalable | Requires more hardware |
| Free and open source | Expensive |

**Commands of docker**

**Check the docker version**

Docker –version

**Check docker system info**

Docker info

**Pull an image from docker hub**

Docker pull <image\_name>

**List download images**

Docker images

**Remove an image**

Docker rmi <image\_id>

**Create container**

Docker run -itd –name Madhuri -p 40:80 <image id>

**List running containers**

Docker ps

**List all containers**

Docker ps -a

**Stop a running container**

Docker stop <container\_id>

**Restart a stopped container**

Docker start <container\_id>

**Remove a container**

Docker rm <container\_id>

**Run a Container in Interactive Mode (with Bash)**

docker run -it ubuntu /bin/bash

**Create an Image from a Dockerfile**

docker build -t my-image

**Tag an Image**

docker tag my-image myrepo/my-image:v1

**Push an Image to Docker Hub**

docker push myrepo/my-image:v1

**List Docker Networks**

docker network ls

**Create a New Network**

docker network create my-network

**Connect a Container to a Network**

docker network connect my-network <container\_id>