**DOCKER**

1. **What is Docker?**

Docker is Open Source software. It provides the automation of Linux application deployment in a software container.

We can do operating system level virtualization on Linux with Docker.

Docker can package software in a complete file system that contains software code, runtime environment, system tools, & libraries that are required to install and run the software on a server.

1. **What is the difference between Docker image and Docker container?**

Docker container is simply an instance of Docker image.

A Docker image is an immutable file, which is a snapshot of container. We create an image with **build** command.

When we use run command, an Image will produce a container.

In programming language, an Image is a Class and a Container is an instance of the class.

1. **How will you remove an image from Docker?**

We can use docker rmi command to delete an image from our local system.

Exact command is:

% docker rmi <Image Id>

If we want to find IDs of all the Docker images in our local system, we can user docker images command.

% docker images

If we want to remove a docker container then we use docker rm command.

% docker rm <Container Id>

1. **How is a Docker container different from a hypervisor?**

In a Hypervisor environment we first create a Virtual Machine and then install an Operating System on it. After that we deploy the application. The virtual machine may also be installed on different hardware configurations.

In a Docker environment, we just deploy the application in Docker. There is no OS layer in this environment. We specify libraries, and rest of the kernel is provided by Docker engine.

In a way, Docker container and hypervisor are complementary to each other.

1. **Can we write compose file in json file instead of yaml?**

Yes. Yaml format is a superset of json format. Therefore any json file is also a valid Yaml file.

If we use a json file then we have to specify in docker command that we are using a json file as follows:

% docker-compose -f docker-compose.json up

1. **Can we run multiple apps on one server with Docker?**

Yes, theoretically we can run multiples apps on one Docker server. But in practice, it is better to run different components on separate containers.

With this we get cleaner environment and it can be used for multiple uses.

1. **What are the common use cases of Docker?**

Some of the common use cases of Docker are as follows:

1. **Setting up Development Environment**: We can useDocker to set the development environment with the applications on which our code is dependent.
2. **Testing Automation Setup**: Docker can also help increating the Testing Automation setup. We can setup different services and apps with Docker to create the automation testing environment.
3. **Production Deployment**: Docker also helps inimplementing the Production deployment for an application. We can use it to create the exact environment and process that will be used for doing the production deployment.

1. **What are the main features of Docker-compose?**

Some of the main features of Docker-compose are as follows:

1. **Multiple environments on same Host**: We can use it tocreate multiple environments on the same host server.
2. **Preserve Volume Data on Container Creation**: Dockercompose also preserves the volume data when we create a container.
3. **Recreate the changed Containers**: We can also usecompose to recreate the changed containers.
4. **Variables in Compose file**: Docker compose also supportsvariables in compose file. In this way we can create variations of our containers.

1. **What is the most popular use of Docker?**

The most popular use of Docker is in build pipeline. With the use of Docker it is much easier to automate the development to deployment process in build pipeline.

We use Docker for the complete build flow from development work, test run and deployment to production environment.

1. **What is the role of open source development in the popularity of Docker?**

Since Linux was an open source operating system, it opened new opportunities for developers who want to contribute to open source systems.

One of the very good outcomes of open source software is Docker.

It has very powerful features.

Docker has wide acceptance due to its usability as well as its open source approach of integrating with different systems.