Docker Assignment 1

1. What is Docker?

Docker is an open platform for developing, shipping, and running applications.By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, we can significantly reduce the delay between writing code and running it in production.

2. What is a Container?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

3. What are Docker Images?

A Docker image is a read-only template that contains a set of instructions for creating a container that can run on the Docker platform. It provides a convenient way to package up applications and preconfigured server environments, which we can use for your own private use or share publicly with other Docker users.

4. What is Docker Hub?

Docker Hub is a cloud-based repository in which Docker users create, test, store and distribute container images. Through Docker Hub, a user can access public, open source image repositories, as well as use a space to

create their own private repositories, automated build functions and work groups.

5. Explain Docker Architecture?

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.

Docker client uses commands and REST APIs to communicate with the Docker Daemon (Server). When a client runs any docker command on the docker client terminal, the client terminal sends these docker commands to the Docker daemon. Docker daemon receives these commands from the docker client in the form of command and REST API's request.

Docker Host is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks, and storage.

Docker Registry manages and stores the Docker images.

There are two types of registries in the Docker -

Public Registry - Public Registry is also called as Docker hub.

Private Registry - It is used to share images within the enterprise.

Docker objects:

Docker images are the read-only binary templates used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses public container registry to share container images within the whole world.

Docker Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources.

In other words, we can say that the image is a template, and the container is a copy of that template.

6. What is a Dockerfile?

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

7. What is the purpose of the EXPOSE command in Dockerfile?

The EXPOSE instruction exposes a particular port with a specified protocol inside a Docker Container. In the simplest term, the EXPOSE instruction tells Docker to get all its information required during the runtime from a specified Port.

8. Why is docker monitoring necessary?

Docker monitoring helps you to: Detect and solve issues early and proactively to avoid risks in production. Implement changes safely as the entire environment is monitored.

9. Explain the implementation method of continuous integration (CI) and continuous deployment (CD) in Docker.

- First we need to build a docker file.
- We run this docker file on a container.
- Now, we need to setup continuous integration tool to test and deploy the code.
- We need to setup a production server to serve the code.
- At last, we deploy our code in our production server.

10. What is a Docker Engine?

Docker Engine is an open source containerization technology for building and containerizing your applications. Docker Engine acts as a client-server application with:

- A server with a long-running daemon process dockerd.
- APIs which specify interfaces that programs can use to talk to and instruct the Docker daemon.
- A command line interface (CLI) client docker.