1. What is Docker?

Docker is an open-source lightweight containerization technology. It has gained widespread popularity in the cloud and application packaging world. It allows you to automate the deployment of applications in lightweight and portable containers.

1. What is a Container?

Docker containers include the application and all of its dependencies. It shares the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud. Docker containers are basically runtime instances of Docker images.

1. What are Docker Images?

The Docker image help to create Docker containers. You can create the Docker image with the build command. Due to this, it creates a container that starts when it begins to run. Every docker images are stored in the Docker registry.

1. What is Docker Hub?

Docker hub is a cloud-based registry that which helps you to link to code repositories. It allows you to build, test, store your image in Docker cloud. You can also deploy the image to your host with the help of Docker hub.

1. Explain Docker Architecture?

Docker Architecture consists of a Docker Engine which is a client-server application with three major components:

1. A server which is a type of long-running program called a daemon process (the docker command).
2. A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
3. A command line interface (CLI) client (the docker command).
4. The CLI uses the Docker REST API to control or interact with the Docker daemon through scripting or direct CLI commands. Many other Docker applications use the underlying API and CLI.
5. What is a Dockerfile?

Let’s start by giving a small explanation of Dockerfile and proceed by giving examples and commands to support your arguments.

Docker can build images automatically by reading the instructions from a file called Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build, users can create an automated build that executes several command-line instructions in succession.

The interviewer does not just expect definitions, hence explain how to use a Dockerfile which comes with experience. Have a look at [this](https://www.edureka.co/blog/docker-explained/) tutorial to understand how Dockerfile works.

1. 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.

1. Why is docker monitoring necessary?

**Detect and solve issues early and proactively** to avoid risks in production. Implement changes safely as the entire environment is monitored. Fine-tune applications to deliver improved performance and better user experience. Optimize resource allocation.

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

You need to do the following things:

* Runs Jenkins on docker
* You can run integration tests in Jenkins using docker-compose

1. What is a Docker Engine?

Docker daemon or Docker engine represents the server. The docker daemon and the clients should be run on the same or remote host, which can communicate through command-line client binary and full RESTful API.