**Introduction to Docker:**  
Docker is a very popular and powerful open-source containerization platform that is used for building, deploying, and running applications.

**What is a Container?**  
A container is a standard unit of software bundled with dependencies so that applications can be deployed fast and reliably b/w different computing platforms.

* Docker container doesn’t require the installation of a separate operating system. Docker makes use of the kernel’s resources and uses resource isolation for CPU and memory
* The main aim of docker containers is to get rid of the infrastructure dependency while deploying and running applications

**Docker Images:**

They are executable packages (bundled with application code & dependencies, software packages, etc.) for the purpose of creating containers.

**Docker File:**

It is a text file that has all commands which need to be run for building a given image.

**What is the functionality of a hypervisor?**

* A hypervisor is a software that makes virtualization happen. This divides the resources of the host system and allocates them to each guest environment installed.
* This means that multiple OS can be installed on a single host system.

**What can you tell about Docker Compose?**

It is a YAML file consisting of all the details regarding various services, networks, and volumes that are needed for setting up the Docker-based application. So, docker-compose is used for creating multiple containers, host them and establish communication between them. For the purpose of communication amongst the containers, ports are exposed by each and every container.

**Can you tell something about docker namespace?**

Namespaces introduce a layer of isolation amongst the containers. In docker, the namespaces ensure that the containers are portable and they don't affect the underlying host.

**What is docker image registry?**

* A Docker image registry, is an area where the docker images are stored. Instead of converting the applications to containers each and every time, a developer can directly use the images stored in the registry.
* This image registry can either be public or private and Docker hub is the most popular and famous public registry available.

**Differentiate between virtualization and containerization.**

| **Virtualization** | **Containerization** |
| --- | --- |
| This helps developers to run and host multiple **OS** on the hardware of a single physical server. | This helps developers to deploy multiple **applications** using the same operating system on a single virtual machine or server. |
| **Hypervisors** provide overall virtual machines to the guest operating systems. | **Containers** ensure isolated environment/ user spaces are provided for running the applications. Any changes done within the container do not reflect on the host or other containers of the same host. |
| These virtual machines form an **abstraction of the system hardware** **layer**this means that each virtual machine on the host acts like a physical machine. | Containers form **abstraction of the application** **layer** which means that each container constitutes a different application. |

### Can you tell the difference between CMD and ENTRYPOINT?

* CMD commands can be changed at run time (while creating the container), but not at the ENTRYPOINT run time.
* ENTRYPOINT specifies that the instruction within it will always be run when the container starts.   
  - The most commonly used ENTRYPOINT is /bin/sh or /bin/bash for most of the base images.
* Every DockerFile should have at least one of these two commands

### Can you tell the difference between ADD and COPY?

**COPY** provides just the basic support of copying local files into the container whereas **ADD** provides additional features like remote URL and tar extraction support.

**Importance of docker layer in dockerfile**

While creating a dockerfile, our primary goal is to keep the image size as less as possible. A Docker image takes up more space for every layer you add to it. Therefore, the more layers the image has, the more space the image requires.

Docker layers contain the state of the docker image at each milestone, and are saved on the local filesystem, layers act as a cache. If nothing is changed in a layer, we can simply re-use the files without the need to rebuild that particular layer.