Q) What is Docker ?

- Docker is a containerization platform
- The whole idea of Docker is to easily develop applications, ship them into containers which can then be deployed anywhere.
- It is used to package all the softwares and application code in one container for execution.
- Container will take care of everything which is required to run our application.
- This guarantees that the software will always run the same regardless of its environment.

Q) Explain the architecture of Docker

- Docker uses a client-server architecture
- Docker client is a service which runs a command. The command is translated using REST API and sent to the Docker Daemon.
- Docker Daemon accepts the request and interact with the os in order to build Docker Images and run Docker containers.
- A Docker Image is a template of instruction which is used to create containers.

Q) What are the advantages of Docker over virtual machine

| Virtual machine | Docker |
|---|---|
| ① Occupies a lot of memory space | ① Docker containes occupy less space |
| ② long boot-up time | ② short boot-up time |
| ③ Running multiple virtual m/c leads to unstable performance | ③ Containers have a better performance as they are hosted in a single Docker engine. |
| ④ Difficult to scale up | ④ Easy to scale up |
| ⑤ low efficiency | ⑤ High efficiency |

Q) What is Docker images ?

- It is a package that contains code + dependencies
- Docker images are used to create docker containers
- when a user runs a Docker image, an instance of a container is created. These Docker image can be deployed to any Docker env.

Q) what is Docker container ?

- It is the runtime process which runs our application
- Docker container include the appln and all its dependencies.
- They share kernel and system resources with other containers and run as isolated systems in 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 instance of Docker images.

Q) ~~Docker hub~~ Docker image registry [Diff bet$^{w}$ Docker registry and hub]

- A Docker image registry is an area where the docker images are stored. Instead of converting the appln to containers each and every time, a developer can directly use the 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
- That provides a centralized repository for sharing and distributing docker images

Q) what is the lifecycle of Docker container

Q) what are the various state that a Docker container can be at any given point of time

⇒ ① Create a container
② Run the created container
③ Pause the processes running inside the container
④ Unpause the processes running inside the container
⑤ start the container, if exist in a stopped state
⑥ stop the container as well as the running processes
⑦ Restart the container as well as running processes
⑧ Kill the running container
⑨ Destroy the container, only if it exist in stopped state

Q) what platform does docker support

⇒ Docker run on various linux administration:

- Ubuntu 12.04, 13.04 et al
- Fedora 19/20+
- RHEL 6.5+
- Centos 6+
- Gentoo
- Archlinux
- OpenSUSE 12.3+
- CRUX 3.0+

It can also be used in Production with cloud platform with the following services:

- Amazon EC2
- Amazon ECS
- Google Compute Engine
- Microsoft Azure
- Rackspace

Q) will you lose your data, when a docker container exists?

⇒ No, you won't lose any data when container exists. Any data that your application writes to the container get preserved on the disk ~~until~~ until you explicitly delete the container. The file system for the container persist even after the container halts.

Q) Diff between Docker run and the docker create

- Docker run is basically running the command in the container

  $ docker run -it <container-name> /bin/bash

- The above is for creating a bash terminal. And make us use bash commands in the container.
- Docker create is to create a container from a docker image

  $ docker create -d /var/lib:/var/lib --name docker-ubuntu ubuntu

- The above is create a container of the name "docker-ubuntu" from the image "ubuntu".

Q) Can you remove paused container from docker

⇒ Yes, Docker container must be in stopped state before removing from ~~a~~ docker container.

Q) Difference between docker pause and docker stop

Docker Pause: Pauses the Processes in a container, essentially freezing its state. The Processes are stopped but remain in memory and the container can be resumed later with 'docker unpause'.

$ docker Pause <container id>

Docker stop: Gracefully stops the Processes in a container, allowing them to clean up and exit. The container is then in the stopped state and can be restarted with 'docker start', but its state is not preserved, and any changes made during its runtime are saved.

Q) Is there a Possibility that a container can restart all by itself in docker?

- No its not Possible
- The default-restart Flag is set to never restart on its own.

Q) why do we use docker on top of a virtual machine?

- Generally we use docker on top of a virtual mlc to ensure isolation of appln.
- on a virtual mlc we get the advantage of security Provided by hypervisor.
- we can implement diff security levels on virtual mlc.
- Docker can make use of this to run the appln at diff security levels.

Q) Do you think Docker appln is Appln centric or mlc centric

- Docker is an application-centric solution
- It is optimized for the deployment of appln
- It does not replaces the machine by creating the virtual mlc. Rather it focuses on Providing ease of use features to run an appln.

Q) what are the object created by docker cloud in Amazon web services. (AWS) EC2

⇒ Docker created following object in AWS EC2 instance:

VPC: created VPC with tag name dc-vpc. It also created class less inter-Domain Routing (CIDR) with the range 10.78.0.0/16

Subnet: created subnet in each AZ. subnet is tagged with dc-subnet.

Internet Gateway: created IGW with dc-gateway name and attached to vpc created earlier.

Routing Table: create Routing table with dc-route-table in vpc. In this Routing Table docker cloud associates the subnet with the IGW.

Q) what are the main security concern with docker based containers?

⇒ Docker based container has following security-concerns:

* Kernel sharing: In a container-based system, multiple containers share same kernel. If one container causes kernel to go down, it will take down all the containers. In vm envilo virtual mic env we do not have this issue.

* Container leakage: If a malicious user gain access to one container, it will try to access other containers on the same host. If a container has security vulnerabilities it can allow the user to access other containers on same host mic.

Denial of service: If one container occupies the resource of a kernel then other containers will starve for resources. It can create a Denial of service attack like situation.

Tampered image: Sometimes the container image can be tampered. This can lead to further security concerns. An attacker can try to run the tampered image to exploit the vulnerabilities in host mic and other containers

Secret sharing: Generally other containers can access other services. To access a service it require a key or secret. A malicious user can gain access to this secret. Since multiple containers share the secret, it may lead to further security concerns

Q) what is Build cache in Docker?

- when we build an image, Docker will process each line in Dockerfile.
- It will execute the commands on each line in the order that is mentioned in the file. But at each line, before running any command, Docker will check if there is already an existing image in its cache that can reuse rather than creating a new image.
- This method of using cache in Docker is called Build cache in Docker

Q) How do you debug the existed container?

⇒ ① Identify the container id "docker ps -a"

② use docker log <container-id> to check the logs

③ Inspect the container to gather detail info. docker inspect <container-id>

④ Restart the container to see real time log and debug

docker start <container-id>

⑤ docker exec -it <container-id> /bin/bash

Q) Can You use multiple FROM in Dockerfile?

⇒ No

Q) Dockerfile run on which user?

⇒ Dockerfile runs as a root user by default.

- Each instr in dockerfile is executed with root privileges.
- we can consider using user specification in Dockerfile to run Processes with lower privileges if needed.

Q) Can we run container inside the container?

⇒ Yes, But its generally not recommended due to complexities and Potential security concerns. It is called as nested containerization

- It is often better to use alternate approach like using container orchestration tool like kubernates to manage multiple container in more controlled and scalable manner

Q) what is called as "FROM SCRATCH" in docker?

⇒ - FROM SCRATCH in a dockerfile refers to starting with an empty, minimal image.

- Its the smallest and the lightest base image possible, containing only the essential component to run our appln.
- It is often used for building very minimal and specialized docker image to reduce size and attack surface.

Q) what happen when You delete /var/lib/docker/overlay2

⇒ In the directory docker images are stre. Due to which we will loose at the container images and data. which may result in the inability to run existing containers or pull new images.