DOCKER INTERVIEW QUESTIONS

1. What are the main drawbacks of Docker?

- Docker doesn’t provide storage options.
- There is no automatic rescheduling of no active nodes.
- There is no monitoring facility (implicitly)
- In Docker automatic horizontal scaling is complicated.

2. What is Docker Image?

- Docker Image is an order collection of layers.
- It is essentially a snapshot of system.
- Docker Image is used to create containers.
- Image is a wrapper that will group all of it’s application dependencies.

3. What is Docker Daemon?

Docker Daemon is also called as Docker Engine, it is a background process which runs on the host system responsible for building and running of containers.

4. What command you should run to see all the running containers?

$ sudo docker ps displays all the running containers.

5. Which command you use to stop the docker container?

$ sudo docker stop containerid is used to stop the container.

6. What is Containerization?

Containerization is basically packaging or wrapping up the software code with just the operating system libraries and dependencies required to run the code to create a single lightweight executable package called as container that runs consistently on any infrastructure.

7. What are the advantages of Containerization?

- Quick software development
- Easy software delivery
- Hassle free running and building of containers in different environments like Dev, testing and production environment.

8. How to remove container upon exit?

By using comman **$ sudo docker run –rm imagename** we can remove container upon exit.

9. What is the command used to run the image as a container?

$ sudo docker run it –name containername imagename is used to run the image as a container.

10. Which command is used to view list of exited containers?

```
$ sudo docker ps –a –f status=exited
```

11. How to remove all exited containers?

```
$ sudo docker rm $(docker ps –a –f status=exited –q)
```

12. How to remove containers having more than one filter?

```
$ sudo docker rm $(docker ps –a –f status=exited –f status=created –q)
```

13. What is the difference between VM and Containers?

Virtual machines	Docker containers
1. VM follows Hardware level virtualization.	1. Containers follow OS level virtualization.
2. Virtual machines use Operating system to provide isolation.	2. Containers use kernel features to provide isolation.
3. Virtual machines consume more space.	3. Containers consume less space.
4. Virtual machines take huge time to boot. So, it takes long time to start.	4. Containers take little time to boot. So, it takes less time to start a

	container.
5. Virtual machines have dedicated memory.	5. But, Docker doesn’t have certain dedicated memory.

14. What are the common instructions in Dockerfile?

FROM, RUN, ENTRYPOINT, MAINTAINER, EXPOSE, ADD, COPY, ONBUILD, USER, ENV, WORKDIR, LABEL etc….

15. What is Docker Swarm?

Docker Swarm is a kind of Orchestration tool used to handle the group of Docker hosts into single and provides the virtual docker host to manage more than one container in an orchestrated way.

16. How can we monitor docker in the production environment?

By using docker stats and docker events commands we could monitor how the resources are consumed by the containers.

17. What are the Docker Container states or What is the Container Life Cycle?

Create, start, stop, restart, pause, unpause, kill and remove the containers.

18. What is the Docker Hub?

Docker Hub is a cloud based registry which offers private and public registries that helps us to run, build, test our images and helps us to link with different repositories and it also provides a facility to automatically push the images from different repositories to the Docker Hub.

19. What is a Registry?

Registry is nothing but collection of repositories.

20. What are the Docker Object Labels?

Images, Containers, Volumes, Networking, Swarm nodes, Services.

21. Where the Docker Volumes are stored?

/var/lib/docker/volumes

22. How the communication happens between Docker Client and Docker Daemon?

The communication between Docker Client and Docker Daemon is done through Rest API, Socket.io and TCP.

23. What is your view about Docker in CI/CD?

We are containerizing Micro-Service applications, as a part of containerization development we integrated the container with the continuous integration tool called as Jenkins or Code Build.
Whenever the source code has been completed by the developers , then we build the Dockerfile as the part of configurations and we integrate Github with the Jenkins and whenever any changes are made to the repository , by integrating Webhooks we download the source code from the repository and we develop some automated scripts in the groovy pipeline as a separate stage and that particular stage is going to download the source code and it will start running the Dockerfile and once the Dockerfile creates the image, then we will try to push that images to the Docker Hub, once the images are stored in the Docker Hub or any other private repositories, through other stages in the CI/CD pipeline it downloads the latest images via Ansible Playbook and starts creating the container and deploys containers into respective environment, then we finish all kinds of tests from Unit testing to the User Acceptance Test (UAT) and once we are done with the tests based on manual approval deployment we release the code to the production environment and from next time onwards we are going to automate this process as a Deployment.

24. What are the commands used to control Docker with systemd?

service docker start/stop/restart/status systemctl start/stop docker

25. How to push the Image to Docker Registry?

sudo docker push dockerhubusername/image:tag

26. What is CNM in Docker?

CNM stands for Container Networking Model. Using CNM we can develop, maintain and monitor the communication between different containers. Using CNM we can form the networking in our Docker environment.

27. Does the Docker support IPV6?

If we want to setup IPV6 we need to change the file
/etc/docker/daemon.json
There will be a key called IPV6. We need to set this as True. Then the Docker will support the IPV6.

28. Do we lose the data when the container exits?

Yes, when we don't have volumes.

&

No, when we apply volumes the data will still persist even after exiting the container.

29. How to configure default logging driver?

To configure the Docker Daemon to default to a specific logging driver, set the value of log-driver to the name of the logging driver in the daemon.json configuration file.

30. Explain or What is DTR?

DTR stands for Docker Trusted Registry. It is an Enterprise Image storing tool for Docker.

31. What are the name spaces in Docker?

When we create a container, Docker utilises a concept called Name Spaces. Name Spaces are useful for creating isolated environment for Docker Containers.

32. What are the Key components of Docker Architecture?

Docker Client, Docker Daemon, Docker Host and Docker Registry.

33. What is the purpose of Docker Host?

Docker host contains Docker Daemon along with Containers and Images. It's purpose is to provide an isolated environment to run and execute our applications as containers.

34. What is Docker Client?

Docker Client is a CLI tool that let's us to interact with the Docker Daemon. It means it accepts the commands from the users and sends to the Docker Daemon.

35. What is Docker Engine/Daemon?

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.

36. Do all containers share same host operating system in Docker?

Yes, all containers share same host operating system in Docker as it utilise OS level virtualization.

37. How to reduce Docker Image size?

Use .dockerignorefile

Use Caching process

Use trusted official images

Prevent updating unnecessary packages and dependencies.

38. Why we should reduce the size of the Docker Image?

- By reducing the size of the image we can run the container faster.
- By reducing the size of the image we can remove unnecessary data, that way we can keep our docker image simple and staraight which increase the performance.
- By reducing size of the image we can reduce the span of downloading , creating, running or pushing the image.

39. What is the Build Context?

- When we start running the Docker build command, the very first line is called Build Context.
- Build Context means it is extracting the layers or files where we specify the Dockerfile path or specific URL.
- Build context works at the path level and URL level.
- sudo docker build –t imagename .
- sudo docker build –t imagename:tag URL

40. What is the use of .dockerignore file in the docker?

- When we don't want to keep the unnecessary files while building an image .dockerigonre file will help us.
- It is used to specify some kind of files or directories that have to be ignored while building the images.
- We can reduce the size of the image too.

41. What is the Multi-stage Build in Docker?

- Multi-stage Build is a technique to reduce the Image size.
- To organize the Docker commands or Docker files and to improve performance.
- To avoid unnecessary files or artifacts.

42. How to run our own index and Registry?

- Need to deploy the index components and the registry from the Github
- Configuration of our server with Docker Registry.
- Setup SSL on the webserver for secure communication.
- Finally push the images to newly created Docker Registry.