1. What is Docker?

ANS) Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

2. What is the difference between Docker image and Docker container?

Ans) The key difference between a Docker image vs a container is that a Docker image is a template that defines how a container will be realized. A Docker container is a runtime instance of a Docker image.

3. How will you remove an image from Docker?

Ans) docker rmi <your-image-id>

4. How is a Docker container different from a virtual machine?

Ans) Now I'll tell you the significant differences between docker containers and virtual machines. Well, the significant differences are their operating system support, security, portability, and performance.

5. Explain basic Docker usage workflow? Docker DocumentationV1.pdf

Ans) The Dockerfile is the source code of the Image. Once the Dockerfile is created, you build it to create the image of the container. The image is just the "compiled version" of the "source code" which is the Dockerfile. Once you have the image of the container, you should redistribute it using the registry.

6. What is the most popular use of Docker?

Ans) Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services. Containers

are great for continuous integration and continuous delivery (CI/CD) workflows.

Ex: Fast, consistent delivery of your applications

7. What is a Docker Image?

Ans) A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template. Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments.

8. What is a Docker Container?

Ans) A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

9. Can we lose our data when a Docker Container exits?

Ans) Do I lose my data when the container exits? 📎 Not at all! Any data that your application writes to disk gets preserved in its container until you explicitly delete the container.

10. How do you make sure our data exists even though you deleted container?

Ans)

11. How can you do a Volume Mapping?

Ans) 1.Display all the existing Docker volumes...

2Creating a Volume

3.inspecting docker volumes..

4.mouting docker volumes

5.creating another container and mount the volume.

12. Can I do a Read-only Mapping?

Ans)

13. How to mount files from physical server to docker container?

Ans)

14. Can we run more than one process in a Docker container?

Ans)

15. What is Docker Hub?

Ans) Docker hub is a hosted repository service provided by Docker for finding and sharing container images with your team.key features include:private Repository: Push and pull container images.Automated Builds : Automatically build container images from GITHUB and BITBUCKET and push them to docker Hub.

16. What are the main features of Docker Hub?

Ans) The main features of docker Hub

✓Repositories: push and pull container images.Team& Organization: Manages access to private Repository of container images.Docker official images:pull and use high-quality container in mages provided by Docker.

17. How can we check the status of a Container in Docker?

Ans)

1.Obtain the container ID by running the following command :

EX: docker ps ….

2.Access the docker container by running the following command:

EX: docker execute – it <container _id> / bin/bash

18. What is the difference between docker ps and dockerps-a commands?

Ans)Docker ps is using only running containers only showing and docker ps –a is all continer ids and continer names are showing.

19. How do you login to the running container?

Ans) docker exec -it <container_id> /bin/bash.

20. What are the various states that a Docker container can be in at any given point in time?

Ans) created A container that has been created (e.g. with docker create) but not started. restarting A container that is in the process of being restarted. running A currently running container. paused A container whose processes have been paused.

ex : Running. Paused. Restarting.

21. What are the main benefits of using Docker?

Ans) Portability Across Machines.
Rapid Performance.
Lightweight.
Isolation.
Scalability.
Docker Hub.
Docker Store.
Docker Image.

22. What are the popular tasks that you can do with Docker Command line tool?

Ans)

- **docker –version**
- **docker pull**
- **docker run**
- **docker ps**
- **docker ps –a**
- **docker exec**
- **docker stop**
- **docker kill**
- **docker commit**
- **docker login**

- **docker push**
- **docker images**
- **docker rm**
- **docker rmi**
- **docker build**

23. What is the user of Dockerfile?

Ans) The default user in a Dockerfile is the user of the parent image. For example, if your image is derived from an image that uses a non-root user example: swuser , then RUN commands in your Dockerfile will run as swuser .

24. What is the difference between Add and Copy command in a Dockerfile?

Ans) COPY is a docker file command that copies files from a local source location to a destination in the Docker container. ADD command is used to copy files/directories into a Docker image.

25. What is the difference between RUN and CMD command in a Dockerfile?

Ans) RUN is an image build step, the state of the container after a RUN command will be committed to the container image. A Dockerfile can have many RUN steps that layer on top of one another to build the image. CMD is the command the container executes by default when you launch the built image.

26. What is Docker Entrypoint?

Ans) An ENTRYPOINT setting specifies both the command and parameters to be run when a container is initialized. All arguments passed when running a container will be included in ENTRYPOINT command and they will override any options specified by CMD.

27. What is the difference between CMD and ENTRYPOINT command in a Dockerfile?

Ans) CMD - The CMD describes the default container parameters or commands. The user can easily override the default command when you use this. ENTRYPOINT - A container with an ENTRYPOINT is preferred when you want to define an executable.

28. What is Build cache in Docker?

Ans) Building images should be fast, efficient, and reliable. The concept of Docker images comes with immutable layers. Every command you execute results in a new layer that contains the changes compared to the previous layer.

29. What are the most common instructions in Dockerfile?

Ans) FORM,LABEL,RUN,CMD.

- **FROM: We use FROM to set the base image for subsequent instructions. In every valid Dockerfile, FROM is the first instruction.**
- **LABEL: We use LABEL to organize our images as per project, module, licensing etc. We can also use LABEL to help in automation.**
  In LABEL we specify a key value pair that can be later used for programmatically handling the Dockerfile.
- **RUN: We use RUN command to execute any instructions in a new layer on top of the current image. With each RUN command we add something on top of the image and use it in subsequent steps in Dockerfile.**
- **CMD: We use CMD command to provide default values of an executing container. In a Dockerfile, if we include multiple CMD commands, then only the last instruction is used.**

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

ANS) 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. These ports can be either TCP or UDP, but it's TCP by default.

31. Which kind of network you are using in docker?

Ans) Bridge networking is the most common network type. It is limited to containers within a single host running the Docker engine. Bridge networks are easy to create, manage and troubleshoot.

32. What is Overlay network in Docker swarm?

Ans) The overlay network driver creates a distributed network among multiple Docker daemon hosts. This network sits on top of (overlays) the host-specific networks, allowing containers connected to it (including swarm service containers) to communicate securely when encryption is enabled.

33. What are the main features of Docker-compose?

Ans) Docker Compose allows you to host multiple isolated environments on one host. Running everything on a single piece of hardware lets you save a lot of resources. Its features that enable it to cache a configuration and re-use existing containers also contribute to resource efficiency.

- **Start, stop, and rebuild services.**
- **View the status of running services.**
- **Stream the log output of running services.**
- **Run a one-off command on a service.**

34. How can we control the start-up order of services in Docker compose?

Ans) You can control the order of service startup and shutdown with the depends_on option. Compose always starts and stops containers in dependency order, where dependencies are determined by depends_on , links , volumes_from , and network_mode: "service:..." .

35. How will you customize Docker compose file for different environments?

Ans)

36. Which version of docker compose you have used?

Ans)

37. In compose file what is the difference between Links and depends on?

Ans) depends_on tells Docker the order of running containers while links, or the network in newer versions of Docker Compose, sets a connection for containers over a network.

38. How to get docker container logs?

Ans) docker container exec, Run a command in a running container ; docker container export, Export a container's filesystem as a tar archive ; docker container inspect

39. What is Docker Swarm?

Ans) A Docker Swarm is a group of either physical or virtual machines that are running the Docker application and that has been configured to join together in a cluster. Docker swarm is a container orchestration tool, meaning that it allows the user to manage multiple containers deployed across multiple host machines.

40. What are the features of Docker Swarm?

Ans) Features of Docker Swarm

- **High Scalability. The Swarm environment is transformed into a highly scalable infrastructure through load balancing.**
- **High-level Security. Any communication between the Swarm's manager and client nodes is encrypted.**
- **Automatic Load Balancing. ...**
- **Decentralized Access. ...**
- **Reverse a Task.**

41. Command to remove all unused 'volumes' from Docker?

Ans)docker prune <coninor name>

42. Can the docker memory be increased? if yes can it done without down time?

Ans)

43. Where do you set the 'http proxy' for Docker in linux with systemd?

Ans) In order to the set the proxy for Docker, you will need to create a configuration file for the Docker service. No configuration files exist by default, so one will have to be created. All Systemd service configuration are stored under /etc/systemd/system.

44. Write a Docker file for running the tomcat service from centos image as root user with port 8080?

Ans)

45. What is the use of 'Docker wait'?

Ans) The 'docker wait' is a command that is used to wait or block until one or more containers stop, and then it outputs their exit codes which means you cannot use your terminal if you are running the command on the terminal.

46. What is your roles & responsibilities in Docker?

Ans) Docker is a software platform that allows you to build, test, and deploy applications quickly.

1. **Managing .NET apps and migrating them off Windows Server**
   2. How networking with containers work and how to build an agile and secure network for containers
   3. How to achieve a secure and compliant application environment for any industry
   4. Integrating Docker with monitoring and logging tools

47. What is Docker compose?

Ans) Docker Compose is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down.

48. What is Docker server version?

Ans) docker version: By default, this will render all version information in an easy ... true Server: Engine: Version: 19.03.8 API version

49. What are the advantages of Docker?

Ans) The benefits of Docker in building and deploying applications are many: Caching a cluster of containers. Flexible resource sharing. Scalability - many containers can be placed in a single host.

50. How do create Docker image from Docker file?

Ans)

1. **Write a Dockerfile for your application.**
2. **Build the image with docker build command.**
3. **Ans)Host your Docker image on a registry.**
4. **Pull and run the image on the target machine.**

51. What is Docker data centre?

Ans) Docker Datacenter is a complete, integrated, enterprise solution for container deployment and management, tailored for production environments with a strong emphasis in availability, multi-tenancy and security.

52. What is Docker hub & uses?

Ans) Docker Hub is a Docker Registry, a cloud-hosted version, open-source, scalable server-side application, and stateless. It can manage the sharing and storage of Docker images.

53. What are the types of Docker networks?

Ans) There are three common Docker network types – bridge networks, used within a single host, overlay networks, for multi-host communication, and macvlan networks which are used to connect Docker containers directly to host network interfaces.

54. How do you define network in the Docker compose file?

Ans) Docker Compose sets up a single network for your application(s) by default, adding each container for a service to the default network. Containers on a single network can reach and discover every other container on the network.

55. What are the basic parameters required in the Docker compose file?

Ans)

56. What is overlay networking?

Ans) An overlay network is a virtual or logical network that is created on top of an existing physical network. The internet, which connects many nodes via circuit switching, is an example of an overlay network. An overlay network is any virtual layer on top of physical network infrastructure.

57. How to communicate between 2 containers present in separate network?

Ans)

1. **Check that the bridge network is running: You can check it's running by typing docker network ls . ...**
2. **Start your containers: Start your containers as normal, with docker run . ...**
3. **Address another container by its IP address: Now one container can talk to another, by using its IP address.**

58. How to store the data present in the Docker container in the AWS?

Ans)

59. If we define the Docker volume in the docker compose file is it possible to share data with the EFS, NFS?

Ans)

60. Difference between image and container?

Ans) Image is a logical entity. Container is a real world entity. Image is created only once. Containers are created any number of times using image.

61. How to Run containers?

Ans) Docker can run your container in detached mode or in the background. To do this, we can use the --detach or -d for short. Docker will start your container the same as before but this time will "detach" from the container and return you to the terminal prompt.

62. Why we need to mention dual ports(8080:8080) in docker run command?

Ans)

63. How to write a docker file to deploy a war file i. We have private repository and don't have base images .how can deploy a war file? ii. Write a simple docker file to deploy a war file by using base images?

Ans)

64. Difference between docker compose and docker swarm?

- **Ans) The few differences between Docker Swarm and Docker-Compose: Docker Swarm is used to scale your web app across one or more servers. Where as Docker-compose wi What is Docker Swarm? In a previous tutorial we had deployed services in**

multiple docker containers and then had these services interact with each other using Docker Networking.

65. Why we need to use docker compose?

Ans

66. Tell me about Docker Network?

- **Ans) Docker networking initiates communication between Docker containers and the external world through the host machine. This is achieved through the use of a container network model (CNM). The model outlines the required steps to provide networking for containers using drivers. Docker relies on several drivers for networking.**

67. Explain about name space in docker?

Ans) Namespace : Docker uses a technology called namespaces to provide the isolated work space called the container. When you run a container, Docker creates a set of namespaces for that container. These namespaces provide a layer of isolation.

68. What is docker why we are using docker difference between vm?

Ans) Another relevant Docker vs Virtual Machine difference is about portability: VMs are isolated from their OS, and so they are not ported across multiple platforms without incurring compatibility issues. At the development level, if an application is to be tested on different platforms, then Docker containers must be considered.

69. What is docker cloud & how it is different from docker hub what is the features over docker hub ?

- **Ans) The Docker Cloud has more added features and so it is more extensive than Docker Hub. It seems like Docker Cloud is built on top of Docker Hub to provide a more comprehensive solution in the Docker eco-system.**

70. What is docker compose ?

- **Ans) Docker Compose is software used for defining and running multi-container Docker applications. It can handle multiple containers simultaneously in the production, staging, development, testing, and CI environment. Therefore, use Docker Compose to manage the whole software development lifecycle (SDLC).**

72. What is the difference between docker cloud and docker swarm?

- **Ans) The difference between Docker and Docker-Swarm is as follows:- The purpose of designing Docker was to create an environment for running containers. Whereas Docker Swarm runs on top of many Docker hosts to orchestrate containers on many machines.**