1. What is Docker?

Docker is an open-source platform that automates the deployment, scaling, and management of applications within isolated containers.

2. What is a container?

A container is a lightweight, standalone, and executable software package that contains everything needed to run a piece of software, including the code, runtime, system tools, and libraries.

3. How does Docker differ from virtualization?

Docker containers share the host OS kernel, making them more lightweight and efficient compared to traditional virtual machines.

4. What is an image in Docker?

An image is a read-only template used to create containers. It includes the application code, runtime, system libraries, and settings.

5. How do you pull an image from Docker Hub?

Use the command: docker pull image-name

6. How do you create a Docker container from an image?

Use the command: docker run image-name

7. How do you list all running containers in Docker?

Use the command: docker ps

8. How do you stop a running container in Docker?

Use the command: docker stop container-id

9. What is a Dockerfile?

A Dockerfile is a text file that contains instructions to build a Docker image. It defines the base image, environment settings, commands, and more.

10. How do you build a Docker image from a Dockerfile?

- Use the command: docker build -t image-name path-to-dockerfile

11. How can you remove a Docker container?

- Use the command: docker rm container-id

12. What is a Docker volume?

- A Docker volume is a directory or data storage mechanism that persists beyond the lifetime of a single container.

13. How do you create and mount a volume in Docker?

- Use the -v flag when running a container: docker run -v /host/path:/container/path image-name

14. What is Docker Compose?

- Docker Compose is a tool for defining and running multi-container Docker applications using a YAML file.

15. How do you define services in a Docker Compose file?

- Define services under the services section in the YAML file.

16. What is Docker Swarm?

- Docker Swarm is Docker's native clustering and orchestration solution for creating and managing a swarm of Docker nodes.

17. How do you initialize a Docker Swarm?

- Use the command: docker swarm init

18. What is the purpose of a Docker registry?

- A Docker registry is a repository for Docker images, allowing you to store, manage, and distribute images.

19. How do you tag a Docker image?

- Use the command: docker tag source-image:tag target-image:tag

20. How do you push a Docker image to a registry?

- Use the commands:

docker login registry-url docker push image-name

21. What is Docker Hub?

- Docker Hub is a cloud-based registry service where you can find, share, and distribute Docker images.

22. How can you expose a port in a Docker container?

- Use the -p flag when running a container: docker run -p host-port:container-port image-name

23. What is the difference between an image and a container in Docker?

- An image is a static blueprint, while a container is a running instance of an image.

24. How can you check the logs of a running Docker container?

- Use the command: docker logs container-id

25. How do you run a command inside a Docker container?

- Use the command: docker exec -it container-id command

26. What is the Dockerfile instruction ENTRYPOINT used for?

- ENTRYPOINT specifies the command that will be executed when the container starts.

27. How can you remove all Docker containers at once?

- Use the command: docker rm $(docker ps -aq)

28. What is Docker's layer caching mechanism?

- Docker uses a layer caching mechanism to optimize image builds. If a layer is unchanged, it can be reused from cache.

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

- CMD provides default arguments for the ENTRYPOINT instruction. It can be overridden when running the container.

30. How can you remove a Docker image?

- Use the command: docker rmi image-id

31. What is Docker's isolation principle?

- Docker uses containerization to provide process isolation, ensuring that each container runs in its own isolated environment.

32. How do you monitor Docker containers' resource usage?

- Use the command: docker stats container-id

33. What is Docker Desktop?

- Docker Desktop is a tool for developing, building, and testing Docker applications on macOS and Windows.

34. What is Docker's network bridge?

- Docker's default network, called the bridge network, allows containers to communicate with each other using their IP addresses.

35. How can you remove all unused Docker resources (containers, images, networks, etc.)?

- Use the command: docker system prune

36. How can you set environment variables in a Docker container?

- Use the -e flag when running a container: docker run -e VAR_NAME=value image-name

37. What is Docker's health check mechanism?

- Docker's health check mechanism monitors the status of a container and can automatically restart it if it becomes unhealthy.

38. How can you pause and unpause a Docker container?

- Use the commands: docker pause container-id and docker unpause container-id

39. What is the purpose of the .dockerignore file?

- The .dockerignore file specifies files and directories that should be excluded from the build context when building Docker images.

40. How do you run a Docker container in the background?

- Use the -d flag when running a container: docker run -d image-name

41. What is Docker's overlay network?

- Docker's overlay network is a network type that allows containers running on different hosts to communicate seamlessly.

42. How can you set a container's restart policy in Docker?

- Use the --restart flag when running a container: docker run --restart policy image-name

43. How do you view the layers of a Docker image?

- Use the command: docker history image-name

44. How can you remove all stopped containers at once?

- Use the command: docker container prune

45. What is Docker's multi-stage build feature?

- Docker's multi-stage build allows you to create smaller and more efficient images by building and copying artifacts in multiple stages.

46. How do you attach to a running Docker container's console?

- Use the command: docker attach container-id

47. How can you set resource constraints on a Docker container (CPU, memory)?

- Use the --cpus and --memory flags when running a container.

48. What is Docker's image layer?

- Each instruction in a Dockerfile contributes to creating a layer in the final image. Images share layers to save disk space and improve build times.

49. How do you create a custom Docker network?

- Use the command: docker network create network-name

50. What is the EXPOSE instruction in a Dockerfile used for?

- The EXPOSE instruction documents which ports the container listens on, but it doesn't actually publish the ports.