Docker Question and Answers

1. **What is Hypervisor?**
   * **Simple Answer:** Hypervisor is software enabling virtualization, creating virtual environments on a single host system. Two types: Type 1 (directly on hardware), Type 2 (uses host operating system).
2. **What is Virtualization?**
   * **Simple Answer:** Virtualization creates virtual versions of hardware or software. Hypervisor software divides a single physical system into multiple virtual environments, allowing various operating systems on one machine.
3. **What is Containerization?**
   * **Simple Answer:** Containerization bundles applications, dependencies, and configurations into containers. It ensures consistency and portability, solving compatibility issues when deploying software on different machines.
4. **Difference between Virtualization and Containerization:**
   * **Simple Answer:** Containers provide isolated environments for applications, while virtualization uses hypervisors to create full virtual machines. Containers are more lightweight, efficient, and share the host OS, while VMs mimic entire systems.
5. **What is Docker?**
   * **Simple Answer:** Docker is a containerization platform that packages applications and dependencies into containers. It ensures consistent performance across different environments, making applications portable and easy to deploy.
6. **What is a Docker Container?**
   * **Simple Answer:** A Docker container includes an application and its dependencies, sharing the host OS kernel. It's an isolated process running on the host system and ensures consistency in application execution.
7. **What are Docker Images?**
   * **Simple Answer:** Docker images are the source of Docker containers. They contain application code, runtime, libraries, and other essential components. Images serve as the basis for creating containers when executed.
8. **What is Docker Hub?**
   * **Simple Answer:** Docker Hub is a registry for Docker images, acting as a repository where users can share and access pre-built Docker images. It allows users to pull images for use in their own containerized applications.
9. **Explain Docker Architecture:**
   * **Simple Answer:** Docker Architecture consists of a Docker Engine, which has a server (daemon), REST API, and a CLI client. Docker commands interact with the daemon through the API, managing containers and other aspects.
10. **What is a Dockerfile?**
    * **Simple Answer:** A Dockerfile is a text document containing instructions for building a Docker image. It specifies the steps to create an automated build, defining the application, dependencies, and configurations.
11. **Tell us something about Docker Compose:**
    * **Simple Answer:** Docker Compose is a YAML file specifying services, networks, and volumes for a Docker application. It allows defining a multi-container application, making it easy to manage and deploy complex setups.
12. **What is a Docker Namespace?**
    * **Simple Answer:** A Docker Namespace is a Linux feature providing isolation for containers. It includes various types such as PID, Mount, IPC, User, and Network, ensuring separation between containers.
13. **What is the lifecycle of a Docker Container?**
    * **Simple Answer:** Docker containers go through stages like creation, running, pausing (optional), unpausing (optional), starting, stopping, restarting, killing, and destruction.
14. **What is Docker Machine?**
    * **Simple Answer:** Docker Machine is a tool that installs Docker Engine on virtual hosts. It facilitates managing Docker hosts, enabling users to provision Docker Swarm Clusters.
15. **How to check for Docker Client and Docker Server version?**
    * **Simple Answer:** Use the command **docker version** to get information about Docker Client and Server versions.
16. **How do you get the number of containers running, paused, and stopped?**
    * **Simple Answer:** Use the command **docker info** to get detailed information, including the number of running, paused, and stopped containers.
17. **If you vaguely remember the command and you’d like to confirm it, how will you get help on that particular command?**
    * **Simple Answer:** Use the command **docker --help** to get help on all Docker commands. For specific commands, use **docker <command> --help**.
18. **How to log into the Docker repository?**
    * **Simple Answer:** Use the command **docker login** to log into hub.docker.com. Enter your username and password when prompted.
19. **If you wish to use a base image and make modifications or personalize it, how do you do that?**
    * **Simple Answer:** Use the command **docker pull <image\_name>** to pull a base image, and then use **docker run -it -d <image\_name>** to create a container and make modifications.
20. **How do you list all the running containers?**
    * **Simple Answer:** Use the command **docker ps** to list all running containers.
21. **Suppose you have 3 containers running, and out of these, you wish to access one of them. How do you access a running container?**
    * **Simple Answer:** Use the command **docker exec -it <container\_id> bash** to access a running container.
22. **How to start, stop, and kill a container?**
    * **Simple Answer:** Use **docker start <container\_id>** to start, **docker stop <container\_id>** to stop, and **docker kill <container\_id>** to kill a container.
23. **Can you use a container, edit it, and update it? Also, how do you make it a new and store it on the local system?**
    * **Simple Answer:** Yes, use the command **docker commit <container\_id> <username/imagename>** to create a new image with changes.
24. **Once you’ve worked with an image, how do you push it to Docker Hub?**
    * **Simple Answer:** Use the command **docker push <username/image\_name>** to push the image to Docker Hub.
25. **How to delete a stopped container?**
    * **Simple Answer:** Use the command **docker rm <container\_id>** to delete a stopped container.
26. **How to delete an image from the local storage system?**
    * **Simple Answer:** Use the command **docker rmi <image\_id>** to delete an image from the local system.
27. **How to build a Dockerfile?**
    * **Simple Answer:** Use the command **docker build <path to docker file>** to build an image from a Dockerfile.
28. **Do you know why docker system prune is used? What does it do?**
    * **Simple Answer:** **docker system prune** is used to remove all stopped containers, unused networks, dangling images, and build caches, freeing up space.
29. **Have you used Kubernetes? If you have, which one would you prefer amongst Docker and Kubernetes?**
    * **Simple Answer:** Express your preference based on experience. If you have used both, highlight strengths and weaknesses of Docker and Kubernetes.
30. **What is Docker Swarm?**
    * **Simple Answer:** Docker Swarm is native clustering for Docker, turning a pool of Docker hosts into a single, virtual Docker host. It provides high availability and scales applications transparently.
31. **If you wish to use a base image and make modifications or personalize it, how do you do that?**
    * **Simple Answer:** Use the command **docker pull <image\_name>** to pull a base image, and then employ **docker run -it -d <image\_name>** to create a container and make modifications.
32. **How do you list all the running containers?**
    * **Simple Answer:** Use the command **docker ps** to list all running containers.
33. **Suppose you have 3 containers running, and out of these, you wish to access one of them. How do you access a running container?**
    * **Simple Answer:** Utilize the command **docker exec -it <container\_id> bash** to access a running container.
34. **How to start, stop, and kill a container?**
    * **Simple Answer:** Use **docker start <container\_id>** to start, **docker stop <container\_id>** to stop, and **docker kill <container\_id>** to kill a container.
35. **Can you use a container, edit it, and update it? Also, how do you make it a new and store it on the local system?**
    * **Simple Answer:** Yes, use the command **docker commit <container\_id> <username/imagename>** to create a new image with changes.
36. **Once you’ve worked with an image, how do you push it to Docker Hub?**
    * **Simple Answer:** Use the command **docker push <username/image\_name>** to push the image to Docker Hub.
37. **How to delete a stopped container?**
    * **Simple Answer:** Use the command **docker rm <container\_id>** to delete a stopped container.
38. **How to delete an image from the local storage system?**
    * **Simple Answer:** Use the command **docker rmi <image\_id>** to delete an image from the local system.
39. **How to build a Dockerfile?**
    * **Simple Answer:** Use the command **docker build <path to docker file>** to build an image from a Dockerfile.
40. **What changes are expected in your docker-compose file while moving it to production?**
    * **Simple Answer:** Remove volume bindings, change port bindings, specify a restart policy, and add services like a log aggregator.
41. **Have you used Kubernetes? If you have, which one would you prefer amongst Docker and Kubernetes?**
    * **Simple Answer:** Express your preference based on experience. If you have used both, highlight strengths and weaknesses of Docker and Kubernetes.
42. **Are you aware of load balancing across containers and hosts? How does it work?**
    * **Simple Answer:** Load balancing across containers and hosts is achieved using tools like HAProxy. It distributes incoming traffic among available containers, ensuring high availability and fault tolerance.