**Basic Docker Interview Questions (1–20)**

1. **What is Docker?**  
   Docker is an open-source platform that automates the deployment of applications inside lightweight, portable containers.
2. **What is a Docker container?**  
   A Docker container is a runnable instance of an image, encapsulating the application and its dependencies.
3. **What is a Docker image?**  
   A Docker image is a read-only template used to create containers, containing everything needed to run an application.
4. **How is a container different from a virtual machine?**  
   Containers share the host OS kernel and are more lightweight, while VMs run a full OS.
5. **What are the main components of Docker?**  
   Docker Engine, Docker Images, Docker Containers, Docker Hub, Docker Compose.
6. **What is Docker Hub?**  
   A cloud-based registry for sharing container images.
7. **What is the command to list all running Docker containers?**  
   docker ps
8. **How do you stop a running Docker container?**  
   docker stop <container\_id>
9. **How can you remove an image from Docker?**  
   docker rmi <image\_id>
10. **What is the purpose of a Dockerfile?**  
    It defines the instructions to build a Docker image.
11. **What is the difference between CMD and ENTRYPOINT in a Dockerfile?**  
    CMD provides default arguments, ENTRYPOINT defines the executable.
12. **What is a Docker volume?**  
    A mechanism for persisting data generated by and used by Docker containers.
13. **What is the purpose of the docker exec command?**  
    To run a command inside an already running container.
14. **How do you copy files into a Docker container?**  
    docker cp <src> <container\_id>:<dest>
15. **How can you expose a port from a Docker container?**  
    Use the -p flag: docker run -p 8080:80 <image>
16. **What is the difference between COPY and ADD in Dockerfile?**  
    ADD has more features (e.g., supports remote URLs and auto-extraction), while COPY is more predictable.
17. **How do you view logs of a running container?**  
    docker logs <container\_id>
18. **How can you check the Docker version?**  
    docker --version or docker version
19. **What is a base image?**  
    The starting point for building a Docker image.
20. **Can you run multiple processes in a container?**  
    It's not recommended, but possible using process managers like supervisord.

**🔹 Intermediate Docker Questions (21–60)**

1. **What is Docker Compose?**  
   A tool to define and manage multi-container Docker applications using docker-compose.yml.
2. **How do you build an image using Dockerfile?**  
   docker build -t <image\_name> .
3. **How do you start containers defined in a Compose file?**  
   docker-compose up
4. **How do you scale services in Docker Compose?**  
   docker-compose up --scale <service>=<count>
5. **What is the purpose of .dockerignore?**  
   Specifies files/folders to exclude from the build context.
6. **What is the difference between a bind mount and a volume?**  
   Bind mount maps to a host path; volume is managed by Docker.
7. **How do you remove all stopped containers?**  
   docker container prune
8. **How to see details of an image?**  
   docker inspect <image\_id>
9. **What is a Docker network?**  
   A way for containers to communicate with each other.
10. **What types of Docker networks are there?**  
    Bridge, host, overlay, none.
11. **How do containers communicate in a Docker network?**  
    By default, containers in the same bridge network can communicate by name.
12. **What is the default network driver in Docker?**  
    Bridge.
13. **How do you attach a container to a network?**  
    docker network connect <network> <container>
14. **Can you run Docker inside Docker?**  
    Yes, using the Docker-in-Docker (dind) image.
15. **What are image layers in Docker?**  
    Images are composed of layers that represent filesystem changes.
16. **What is the difference between docker run and docker start?**  
    run creates and starts a container; start only starts an existing one.
17. **How do you commit changes in a container to a new image?**  
    docker commit <container\_id> <new\_image\_name>
18. **What is a health check in Docker?**  
    A mechanism to monitor the health of a container using the HEALTHCHECK instruction.
19. **What does docker system prune do?**  
    Cleans up unused images, containers, volumes, and networks.
20. **How to limit CPU and memory for a container?**  
    Use --memory and --cpus options.
21. **What is multi-stage build in Docker?**  
    A way to reduce image size by using multiple FROM statements in a Dockerfile.
22. **What is the difference between up, start, and run in Docker Compose?**  
    up builds and starts, start restarts existing, run executes one-off commands.
23. **How do you push an image to Docker Hub?**  
    docker push <username>/<image\_name>
24. **What does docker tag do?**  
    Tags an image with a repository and version name.
25. **What’s the difference between docker attach and exec?**  
    attach connects to main process; exec runs a new process inside the container.
26. **What is docker save and docker load?**  
    Used to save and load images to/from tar archives.
27. **How do you troubleshoot networking issues in Docker?**  
    Use docker network inspect, check DNS resolution, container logs, etc.
28. **What is a dangling image?**  
    An image with no tag, often leftover from builds.
29. **How can you clean up dangling images?**  
    docker image prune
30. **How do you update a running container’s environment variable?**  
    You can’t directly — recreate the container with updated env vars.
31. **How do you create a custom Docker network?**  
    docker network create <network\_name>
32. **What is an orphan container in Docker Compose?**  
    A container defined in a previous version of Compose file that’s no longer referenced.
33. **How does Docker caching work during image builds?**  
    Docker caches each layer and reuses unchanged layers to speed up builds.
34. **What’s the purpose of --detach (-d) flag in Docker?**  
    Runs container in the background.
35. **What is the lifecycle of a Docker container?**  
    Created → Running → Paused/Stopped → Removed
36. **What are labels in Docker?**  
    Metadata used to organize and manage containers/images.
37. **What’s the difference between docker pause and stop?**  
    pause suspends all processes; stop terminates them.
38. **How do you monitor Docker container resource usage?**  
    docker stats
39. **How do you configure Docker to start on boot?**  
    Use systemctl enable docker
40. **How do you debug a container not starting?**  
    Use docker logs, check entrypoint errors, check image, environment vars, etc.

**🔹 Advanced Docker Interview Questions (61–100)**

1. **What are some best practices for writing Dockerfiles?**  
   Use small base images, multi-stage builds, minimize layers, use .dockerignore.
2. **How does Docker ensure image layer integrity?**  
   Uses SHA256 checksums.
3. **What’s the difference between bridge and host networking?**  
   Bridge uses virtual interfaces; host shares the host’s network stack.
4. **How do you secure Docker containers?**  
   Use minimal base images, non-root users, read-only file systems, security scanning.
5. **What is Docker Swarm?**  
   A native clustering and orchestration tool for Docker.
6. **How does Docker Swarm work?**  
   It manages a cluster of Docker nodes as a single virtual system.
7. **What is a service in Docker Swarm?**  
   A scalable group of containers running the same image.
8. **What is a secret in Docker Swarm?**  
   A secure way to store sensitive information like passwords, used in services.
9. **What is overlay network in Docker?**  
   A virtual network that spans multiple Docker daemons.
10. **What’s the difference between Docker and Kubernetes?**  
    Docker handles containers; Kubernetes orchestrates large-scale container deployments.
11. **Can Docker containers have static IPs?**  
    Yes, via custom bridge networks.
12. **How do you handle environment-specific configuration in Docker?**  
    Use environment variables or external config files.
13. **What is user namespace remapping in Docker?**  
    A security feature that maps container UIDs to non-root UIDs on the host.
14. **What is the difference between build, create, run, start?**  
    Build = image, Create = container, Run = create + start, Start = run existing.
15. **What are the security risks in Docker?**  
    Privileged containers, untrusted images, running as root, network exposure.
16. **How can you scan Docker images for vulnerabilities?**  
    Use tools like Docker Scout, Trivy, Clair.
17. **How do you optimize Docker images?**  
    Use Alpine base image, remove temp files, combine RUN statements.
18. **How does Docker handle persistent storage?**  
    Through volumes and bind mounts.
19. **What’s the role of cgroups in Docker?**  
    They manage resources like CPU, memory, disk I/O.
20. **What is the Docker registry?**  
    A storage and distribution system for named Docker images.
21. **What’s the default location of Docker volumes on Linux?**  
    /var/lib/docker/volumes/
22. **How can you share data between containers?**  
    Use shared volumes or a shared network.
23. **How do you roll back to a previous Docker image version?**  
    Use the image tag or pull an older version.
24. **What happens when you delete a container?**  
    The container is removed, but the image and volumes may persist.
25. **How does Docker ensure container isolation?**  
    Through namespaces (PID, NET, IPC, etc.) and cgroups.
26. **How does Docker handle DNS?**  
    Docker provides an internal DNS to resolve container names.
27. **What is the use of --link option?**  
    Deprecated; previously used for container-to-container communication.
28. **Can you run GUI apps in Docker?**  
    Yes, using X11 forwarding or VNC.
29. **How do you make Docker containers restart automatically?**  
    Use --restart=always or --restart=on-failure.
30. **How to manage Docker credentials securely?**  
    Use Docker secrets or credential stores.
31. **How do you create a custom Docker image?**  
    Write a Dockerfile and build it using docker build.
32. **What is docker diff?**  
    Shows changes to the container’s filesystem.
33. **What are the risks of running containers as root?**  
    Escalation of privileges if exploited.
34. **What is init system in Docker containers?**  
    Minimal init systems like tini are used to handle reaping zombie processes.
35. **What is the difference between alpine and ubuntu base images?**  
    Alpine is minimal and lightweight; Ubuntu is feature-rich and heavier.
36. **How to mount a host directory as a volume?**  
    docker run -v /host/path:/container/path
37. **What is Docker context?**  
    Configuration of Docker CLI to connect to different Docker environments.
38. **What is build cache in Docker?**  
    Stores intermediate layers to speed up subsequent builds.
39. **How do you debug Docker build issues?**  
    Use --progress=plain, --no-cache, print statements.
40. **What’s the future of Docker with Kubernetes dominance?**  
    Docker remains crucial for containerization, while orchestration moves to Kubernetes.