Docker Study Guide

Basics

- What is Docker and why is it used?
- What is a Docker container?
- What is a Docker image?
- What is a Dockerfile and what is its purpose?
- Describe the function of Docker Hub and image registries.
- Run `docker run hello-world` and explain what happens when the image isn't present locally.
- Run an `alpine` container with `echo "Hello Basics"`. Check `docker ps` and `docker ps -a`. What's the difference between the image and container?
- Run `nginx` on port 8080, access it, then stop/restart/remove the container. What happens?
- Use `docker logs` and `docker inspect` on a stopped container. What info can you extract?
- What happens to data stored inside a container when it is removed?

Intermediate

- What is the difference between Docker run and Docker start?
- What are Docker volumes and how are they used for data persistence?
- Where are Docker volumes stored, and how can you inspect them?
- Demonstrate passing environment variables to a container (inline `-e` and `.env` file).
- Create a Docker network, attach two containers, and use one to `curl` the other by name. Why does that work?
- Explain Docker networking and list different network drivers.
- What is Docker Compose and how does it simplify multi-container app deployment?
- Write a Dockerfile for a simple Python script, build it, and explain how caching works.
- What is the difference between CMD and ENTRYPOINT?
- Why is `docker system prune` used, and what are the risks?

Advanced

- What is Docker Swarm and how does it enable container orchestration?
- How do namespaces and cgroups provide container isolation?
- How would you limit CPU and memory usage of a container?

- What is the purpose of the `docker checkpoint` command?
- How do you debug issues inside a container?
- What are multi-stage builds in Dockerfiles, and why are they important?
- How can you manage secrets securely within a Docker environment?
- Explain Docker content trust and its significance.
- Describe best practices for optimizing Docker image size and build times.
- Can you implement CI/CD pipelines with Docker? If yes, how?

Scenario-Based

- Demonstrate how to attach a volume to a container and persist data.
- Connect two containers on a custom bridge network and show communication.
- Write a Dockerfile to build an image for a simple Python Flask application.
- Create a `docker-compose.yml` file to deploy a web app with a separate database service.
- Outline the steps to scale a service in Docker Swarm mode.
- You have a container consuming too much memory. How do you diagnose and fix without restarting?
- Your team has multiple environments (dev/stage/prod). How do you manage environment-specific configs?
- You need to deploy a microservices app. How would you structure Dockerfiles and Compose files?
- You need to secure sensitive data inside containers. What strategies do you use?
- How would you scale a web app horizontally using Swarm or Kubernetes?