Docker Preparation: Detailed Outline

1. What are Containers?

Definition:

Containers are lightweight, portable, and self-sufficient units that bundle an application along with its dependencies (libraries, binaries, and configuration files).

They run consistently across various environments (development, testing, production).

How They Differ:

Virtual Machines (VMs): Containers share the host OS kernel, making them faster and more resource-efficient compared to VMs.

Isolation: Containers are isolated from each other and the host OS.

2. What is Docker?

Definition:

Docker is an open-source platform for automating the deployment of applications inside containers.

It simplifies the development, testing, and deployment process by standardizing how software is packaged.

Components:

Docker Engine: Core runtime to build and run containers.

Docker CLI: Command-line interface to interact with Docker.

Docker Hub: Registry for sharing and downloading images.

3. Why Do You Need Docker?

Benefits:

Portability: Consistent environment from development to production.

Efficiency: Reduced overhead compared to VMs.

Speed: Faster startup times for containers.

Isolation: Ensures one app doesn’t interfere with another.

Scalability: Easily scale services horizontally.

Use Cases:

Simplifying microservices deployment.

Streamlining CI/CD pipelines.

Testing in isolated environments.

4. What Can Docker Do?

Create, run, and manage containers.

Automate application deployment across multiple environments.

Enable seamless collaboration through shared Docker images.

Build multi-container applications using Docker Compose.

Orchestrate containerized applications using Docker Swarm or Kubernetes.

5. Run Docker Containers

Pull an Image and Run a Container:

docker pull nginx

docker run -d -p 8080:80 nginx

Basic Commands:

List running containers:

docker ps

Stop and remove containers:

docker stop <container\_id>

docker rm <container\_id>

6. Create a Docker Image

Dockerfile Basics:

Example of a Dockerfile:

Dockerfile

FROM ubuntu:20.04

RUN apt-get update && apt-get install -y python3

COPY app.py /app.py

CMD ["python3", "/app.py"]

Build and Run the Image:

docker build -t my-app .

docker run my-app

7. Networks in Docker

Default Networking Modes:

Bridge: Default network for standalone containers.

Host: Shares the host’s network stack.

None: No network connectivity.

Create a Custom Network:

docker network create my-network

docker run --network=my-network nginx

Inspect Networks:

docker network ls

docker network inspect my-network

8. Docker Compose

Purpose:

Tool for defining and running multi-container applications using a single YAML configuration file.

Example docker-compose.yml:

version: "3.9"

services:

web:

image: nginx

ports:

- "8080:80"

db:

image: mysql

environment:

MYSQL\_ROOT\_PASSWORD: password

Commands:

docker-compose up

docker-compose down

9. Docker Concepts in Depth

Image Layers:

Images are made up of multiple layers, which Docker caches to optimize builds.

Volumes:

Persist data generated by containers:

docker volume create my-volume

docker run -v my-volume:/data ubuntu

Entrypoint vs CMD:

CMD sets the default command.

ENTRYPOINT configures the main command while allowing arguments.

10. Docker for Windows/Mac

Installation:

Use Docker Desktop for Windows/Mac.

WSL 2 Integration (Windows):

Ensures Linux containers run natively on Windows.

11. Docker Swarm

Definition:

Native orchestration tool for managing container clusters.

Key Features:

High availability.

Load balancing.

Service scaling.

Commands:

docker swarm init

docker service create --name my-service --replicas 3 nginx

12. Docker vs Kubernetes

Feature Docker Swarm Kubernetes

Ease of Use Simple setup, suitable for beginners Complex but highly scalable

Scalability Limited scalability Handles thousands of nodes

Ecosystem Focuses on Docker tools Extensive ecosystem and plugins

Load Balancing Built-in Advanced load balancing options

Community Support Smaller community Larger community and adoption