Docker Scenario-Based Interview Questions & Answers

1. Scenario: Application Crashes Frequently

Answer: Run the container with --rm and -it to observe logs:

docker run --rm -it myapp

Check logs:

docker logs <container_id>

Use docker inspect <container_id> and try running with interactive shell.

2. Scenario: Application Needs a Secret Key

Answer: Use environment variables or Docker secrets (Swarm mode). Avoid hardcoding secrets in images.

3. Scenario: Application Needs to Scale

Answer: Use Docker Compose: docker-compose up --scale web=3

Or use Docker Swarm/Kubernetes for orchestration.

4. Scenario: Reduce Image Size

Answer: Use smaller base images like alpine. Use multi-stage builds. Clean up unnecessary files and apt cache.

5. Scenario: Data Persistence

Answer: Use Docker volumes: docker volume create mydata

Use bind mounts for local file sharing.

6. Scenario: Network Communication

Answer: Use user-defined bridge networks. Containers can communicate by name.

7. Scenario: Image Vulnerability

Answer: Scan images with docker scan, Trivy, Clair, or Anchore.

8. Scenario: High Availability Setup

Answer: Use Docker Swarm or Kubernetes. Define replicas, health checks, and load balancers.

9. Scenario: Rolling Update Without Downtime

Answer: Use docker service update in Swarm or Deployment strategy in Kubernetes.

10. Scenario: Docker Container Consumes High CPU

Answer: Limit CPU with --cpus flag. Use docker stats for monitoring.

11. Scenario: Application Needs Host Machine Files

Answer: Use bind mounts: docker run -v /host/data:/app/data

12. Scenario: Automating Docker Image Build in CI/CD

Answer: Use GitHub Actions or Jenkins. Include build, tag, and push steps in the pipeline.

13. Scenario: Multi-Environment Configurations

Answer: Use .env files or override Compose files with -f option.

14. Scenario: Debugging a Running Container

Answer: Use docker exec -it <container_id> /bin/bash or /bin/sh.

15. Scenario: Application Needs to Restart on Failure

Answer: Use --restart=on-failure or --restart=always when running containers.

16. Scenario: Service Discovery Between Containers

Answer: Use Docker networks. Containers in the same network can communicate via hostname.

17. Scenario: Clean Up Unused Images and Containers

Answer: Use docker system prune -a or specific prune commands.

18. Scenario: Security Hardening of Docker Image

Answer: Use minimal base images, remove unnecessary packages, avoid root user, and scan with tools like Trivy.