

# Docker Scenario-Based Interview Questions & Answers

## 1. Scenario: Pod CrashLoopBackOff

Answer: Check logs: `kubectl logs <pod-name>`

Describe pod: `kubectl describe pod <pod-name>`

Check liveness/readiness probes and app startup behavior.

## 2. Scenario: Exposing an Application

Answer: Use a Service (ClusterIP, NodePort, or LoadBalancer).

For external access, use Ingress with an Ingress Controller.

## 3. Scenario: Zero Downtime Deployment

Answer: Use Deployments with rolling update strategy.

Specify `maxUnavailable` and `maxSurge`.

## 4. Scenario: Accessing Secrets

Answer: Store secrets in Kubernetes Secret resources.

Mount as environment variables or volumes in Pods.

## 5. Scenario: Scaling Applications

Answer: Use `kubectl scale` or set replicas in Deployment.

Use HorizontalPodAutoscaler for CPU-based scaling.

## 6. Scenario: Debugging a Pod

Answer: Use `kubectl exec -it <pod-name> -- /bin/sh`

Use ephemeral containers or init containers for deeper debugging.

## 7. Scenario: Resource Limits

Answer: Set requests and limits in Pod specs to control resource usage.

## 8. Scenario: Persistent Storage

Answer: Use PersistentVolume and PersistentVolumeClaim.

Use dynamic provisioning with StorageClasses.

## **Docker Scenario-Based Interview Questions & Answers**

### **9. Scenario: Configuration Management**

Answer: Use ConfigMaps to store config data.

Mount them as env variables or volumes.

### **10. Scenario: Rolling Back a Deployment**

Answer: Use `kubectl rollout undo deployment/<deployment-name>` to revert to the previous version.