Kubernetes Interview Questions & Troubleshooting Answers

Basic to Advanced

Section 1: Kubernetes Fundamentals (Q1–Q10)

1. What is Kubernetes, and how does it work?
Kubernetes is an open-source container orchestration platform for automating deployment, scaling, and management of containerized applications. It works by managing containers across a cluster of nodes, ensuring high availability and scalability through its master and worker architecture.

2. Explain the difference between a Pod and a Container.
A container is a lightweight unit that packages code and dependencies. A Pod is the smallest deployable unit in Kubernetes and can hold one or more containers sharing the same network namespace and storage.

3. What is a Namespace in Kubernetes, and why is it used?

A Namespace is a way to divide cluster resources between multiple users. It provides isolation and helps manage resources in large environments.

4. How do you expose a Pod to the external world?

By creating a Service of type LoadBalancer or using an Ingress controller, you can expose a Pod externally.

5. What are Labels and Selectors?

Labels are key-value pairs attached to objects. Selectors allow filtering and selecting resources based on those labels.

6. Difference between ReplicaSet and Deployment.

ReplicaSet ensures the specified number of Pod replicas. Deployment manages ReplicaSets and enables features like rolling updates and rollbacks.

7. What is the purpose of kubectl and how do you use it?

kubectl is the CLI tool to interact with Kubernetes clusters. Example: **kubectl get pods** lists all Pods in the current namespace.

rollbacks.

8. How is a Service different from an Ingress?

A Service provides stable internal access to Pods, while an Ingress manages external HTTP/S access to the services via rules.

9. What is a ConfigMap vs Secret?

Both store configuration data. ConfigMap stores plain-text, non-sensitive data; Secret stores sensitive data in base64 encoding.

10. What are the different types of Services in Kubernetes?

- ClusterIP (default)
- NodePort
- LoadBalancer
- ExternalName

Section 2: Deployment & Configuration (Q11–Q20)

11. How do you perform a Rolling Update and Rollback?
Rolling updates are done with Deployments: **kubectl rollout restart deployment <name>**. Rollback with **kubectl rollout undo deployment <name>.**

12. What is a DaemonSet and use case?
DaemonSet ensures that a Pod runs on all (or some) nodes. Useful for log collectors or monitoring agents.

13. Explain StatefulSet vs Deployment.
StatefulSet is used for stateful applications needing persistent identity/storage. Deployment is for stateless apps.

14. What is an InitContainer and when do you use it?

An InitContainer runs before the main container. Used for setup tasks like waiting for a service or copying files.

15. How do you mount a volume to a Pod?

By defining **volumes** and **volumeMounts** in the Pod spec. Example: **emptyDir, hostPath, persistentVolumeClaim**.

16. Difference between emptyDir and hostPath volumes?

emptyDir creates a temporary volume for Pod lifetime. **hostPath** mounts a directory from the host node into the Pod.

17. How do you pass environment variables to containers?

Using env in the container spec:

```
env:
 - name: APP_MODE
   value: "production"
```

18. How does a Pod lifecycle work?

It includes phases: Pending, Running, Succeeded, Failed, Unknown. Lifecycle hooks: postStart, preStop.

19. What is the use of livenessProbe and readinessProbe?

Liveness checks if the app is healthy. Readiness checks if the app is ready to serve traffic.

20. How do you perform zero-downtime deployments in Kubernetes?

Using rolling updates with readiness probes and ensuring availability of old Pods until new ones are ready.

Section 3: RBAC, Security & Networking (Q21–Q30)

21. What is RBAC in Kubernetes?

Role-Based Access Control restricts access to resources. Defined using Roles and RoleBindings or ClusterRoles.

22. Difference between Role and ClusterRole?

Role is namespace-scoped, ClusterRole is cluster-scoped.

23. What is a ServiceAccount?

It provides identity to processes running in a Pod for API access.

24. What is a NetworkPolicy?

Defines rules to allow or block traffic between Pods based on labels, namespaces, and ports.

25. How do you restrict access to a namespace?

Use Roles and RoleBindings scoped to the namespace.

26. How can you secure Secrets in Kubernetes?
Use RBAC restrictions, enable encryption at rest, and avoid printing them in logs.

27. How do you audit access logs?
Enable audit logging in the API server and review logs based on events.

28. What are PodSecurityPolicies (PSP)?
Deprecated in recent versions; use OPA/Gatekeeper or Kyverno for policy enforcement.

29. What is a Kubernetes Ingress Controller?
It routes external HTTP/S traffic to services based on Ingress rules.

30. Explain Ingress vs LoadBalancer.
Ingress offers routing at HTTP layer with rules. LoadBalancer exposes service using cloud provider's load balancer.

Section 4: Scaling, Monitoring & Logging (Q31–Q40)

31. How do you scale Pods manually and automatically?

Manual: kubectl scale deployment <name> --replicas=5. Automatic: using HPA (Horizontal Pod Autoscaler).

32. What is Horizontal Pod Autoscaler (HPA)?

Automatically scales Pods based on metrics like CPU or memory.

33. Explain Cluster Autoscaler.

It adjusts the number of nodes in the cluster based on Pod resource requirements.

34. What metrics do you use to monitor Kubernetes?

CPU, memory, disk usage, network I/O, request latency, Pod restarts, etc.

35. How to set up Prometheus and Grafana?

Use Helm charts or custom YAMLs. Prometheus collects metrics, Grafana visualizes them.

36. How do you log application data?

Using stdout/stderr of containers. Tools: EFK/ELK, Loki, Fluentd, etc.

37. What is Fluentd?

A data collector for unified logging. Used in Kubernetes to ship logs to Elasticsearch or other systems.

38. Tools for monitoring cluster health?

Prometheus, Grafana, Datadog, New Relic, Kube-state-metrics.

39. How to troubleshoot CrashLoopBackOff?

Check **kubectl describe pod <pod>** and **kubectl logs**. Common reasons: misconfigurations, missing files, failing health checks.

40. How to troubleshoot high CPU usage in a node?

Use **kubectl top nodes**, inspect workloads, check for overprovisioned Pods.

Section 5: Advanced Real-Time Scenarios (Q41–Q50)

41. A Pod is not getting an IP. How to troubleshoot?

How to troubleshoot? Check CNI plugin status, kubelet logs, and network policies.

42. Debug networking between services?

Use nslookup, curl, tcpdump, or network debugging Pods like busybox.

43. How to isolate a node temporarily?

Use kubectl cordon <node> to mark unschedulable and kubectl drain <node> to evict Pods.

44. Deployment is stuck. How to debug?

Check rollout status, events, and describe Deployment and Pods.

45. How to do Blue-Green Deployment?

Deploy new version alongside old, switch traffic using Ingress or Service.

46. What are Admission Controllers?

Plugins that intercept API requests to validate or mutate objects. Example: ResourceQuota, LimitRanger.

47. How to configure Kubernetes for multi-tenancy?

Use Namespaces, NetworkPolicies, RBAC, ResourceQuotas.

48. How to perform a canary release?

Use Deployment with multiple versions and route partial traffic using Ingress or Service mesh.

49. Backup and restore of K8s resources?

Tools: Velero, kubectl get -o yaml, custom scripts with etcd snapshots.

50. What is a CRD and when to use it?

Custom Resource Definitions extend Kubernetes with new resource types. Used to manage custom application components declaratively.

Found this useful?

Follow

Let’s connect

💬 Found it useful? Drop your thoughts below and share it with your fellow DevOps engineers!