Kubernetes Interview Q&A; Guide

Core Kubernetes Concepts

What is Kubernetes, and why is it used?

Kubernetes (K8s) is an open-source container orchestration platform that automates deployment, scaling, and management of containerized applications. It ensures high availability, self-healing, and automated rollouts.

Key Components of Kubernetes Architecture

Control Plane (API server, etcd, controller manager, scheduler) and Worker Nodes (kubelet, kube-proxy, container runtime).

Difference between Pod and Node

Pod is the smallest deployable unit; Node is the machine that runs Pods.

What is a Cluster?

A group of nodes managed by the control plane where containerized applications run.

Role of kube-apiserver

The central management component that exposes the Kubernetes API and communicates with all components.

Namespace

Logical partition within a cluster for isolating resources and avoiding conflicts.

kubectl

CLI tool used to manage Kubernetes resources through API calls.

Labels and Annotations

Labels identify/select objects; annotations store metadata.

Role of etcd

Key-value store for cluster state and configuration.

ReplicaSet vs Deployment

ReplicaSet ensures pod count; Deployment manages ReplicaSets and enables rollbacks.

Pods & Workloads

What is a Pod?

Smallest deployable unit that can contain one or more containers sharing network and storage.

Restart a Pod

Use 'kubectl rollout restart deployment ' or delete the Pod if managed by a controller.

Scale a Pod

Pods scale via Deployments/ReplicaSets using 'kubectl scale' or Horizontal Pod Autoscaler.

DaemonSet

Ensures a copy of a Pod runs on all or selected nodes.

StatefulSet vs Deployment

StatefulSet manages stateful apps with stable storage and identity; Deployment manages stateless apps.

Multi-container Pod

Multiple containers share the same network and storage, used for sidecars or helpers.

Static Pod vs Regular Pod

Static Pods are created by kubelet directly from node files; regular Pods via API server.

Pod Deletion in Deployment

Deployment automatically recreates deleted Pods to maintain replicas.

Resource Limits

Configured using resources.requests and resources.limits in YAML to manage CPU/memory.

Role of Node

Worker machine running Pods with kubelet, kube-proxy, and container runtime.

Services & Networking

What is a Service?

An abstraction that provides stable networking to access a set of Pods.

Types of Services

ClusterIP, NodePort, LoadBalancer, and ExternalName.

ClusterIP Service

Default type; exposes service internally within the cluster.

NodePort Service

Exposes the service externally via a port on each node's IP.

ExternalName Service

Maps a service to an external DNS name.

DNS Resolution
CoreDNS provides internal DNS; services are accessible via service.namespace.svc.cluster.local.

NetworkPolicies
Control Pod-to-Pod and external traffic rules.

Ingress
Manages HTTP/HTTPS access to services using host/path-based routing.

LoadBalancer vs NodePort
LoadBalancer provides an external cloud load balancer; NodePort exposes service via node IP.

Service Discovery
Kubernetes provides discovery using environment variables and internal DNS.

Storage & Configuration

PersistentVolume (PV)
Cluster resource representing actual storage.

PersistentVolumeClaim (PVC)
User request for storage that binds to a PV.

PV vs PVC
PV is the resource; PVC is the request for it.

StorageClass
Defines dynamic provisioning rules for storage backends.

Mount Volume
Define 'volumes' and 'volumeMounts' in Pod YAML.

Volume Types
emptyDir, hostPath, persistentVolumeClaim, configMap, secret, NFS, CSI, etc.

PVC Deletion
PVC remains after Pod deletion; data persists.

Reclaim Policy
Default is 'Delete'; can be 'Retain' or 'Recycle'.

ConfigMap

Stores non-sensitive configuration data in key-value pairs.

Secret
Stores sensitive data (passwords, tokens) securely in base64 format.

Advanced Kubernetes Concepts

Helm
Package manager for Kubernetes; simplifies deployments using charts.

Helm Chart vs Release
Chart is a template package; Release is a deployed instance of that chart.

Kubernetes Operator
Extends Kubernetes to automate complex application lifecycles using CRDs and controllers.

Horizontal Pod Autoscaler (HPA)
Automatically scales Pod replicas based on metrics like CPU or memory usage.

Rolling Updates
Gradually replaces old Pods with new ones to avoid downtime.

Recreate vs RollingUpdate
Recreate deletes all old Pods before creating new; RollingUpdate updates gradually.

Taints and Tolerations
Control which Pods can run on which nodes by restricting or allowing scheduling.

High Availability
Achieved via redundant control planes, ReplicaSets, and self-healing mechanisms.

Custom Resource Definitions (CRDs)
Extend Kubernetes API by defining new resource types.

Admission Controllers
Intercept API requests to enforce validation, modification, or policies.