Kubernetes Interview Questions for DevOps Engineers

Written by Zayan Ahmed | 7 min read

Basic Kubernetes Questions

1. What is Kubernetes?

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery.

2. What is a Kubernetes cluster?

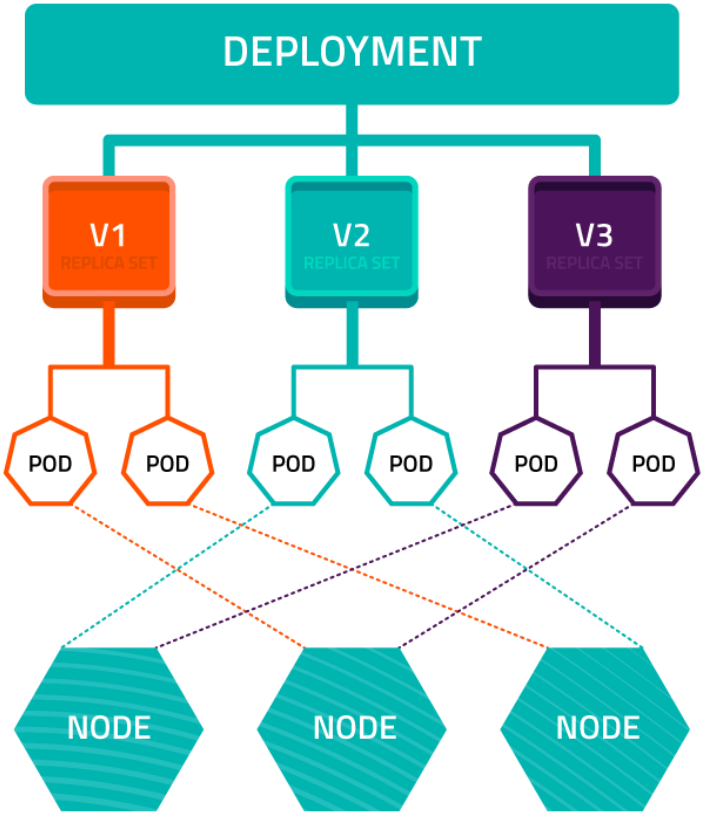
A Kubernetes cluster is a set of nodes (machines) that run containerized applications. A cluster has at least one master node and multiple worker nodes. The master node manages the cluster, while the worker nodes host and run the applications.

3. What are Pods in Kubernetes?

A Pod is the smallest and most basic unit in Kubernetes. It represents a single instance of a running process in your cluster. A Pod can contain one or more containers, such as Docker containers, which share storage and network resources.

4. What is a Namespace in Kubernetes?

Namespaces are virtual clusters within a Kubernetes cluster that allow for separation of resources and organization of workloads. They help in dividing cluster resources among multiple users or teams.



5. What is a Node in Kubernetes?

A node is a machine—either a virtual or physical machine—where Kubernetes runs the Pods. There are two types of nodes:

- **Master Node**: Responsible for the management of the Kubernetes cluster.
- **Worker Node**: Responsible for running the containers and workloads.

6. Explain the Kubernetes Master components.

- **API Server**: Exposes Kubernetes API and acts as the frontend for the Kubernetes control plane.
- **Controller Manager**: Manages controllers such as the replication controller and node controller.
- **Scheduler**: Places the Pods on the worker nodes based on resource availability.
- **etcd**: A distributed key-value store that holds the cluster data and state.

7. What is a ReplicaSet?

A ReplicaSet ensures that a specified number of identical Pods are running at any given time. If a Pod fails, the ReplicaSet will automatically create a new Pod to replace it.

8. What is a Deployment in Kubernetes?

A Deployment is a higher-level abstraction that defines the desired state for an application, such as the number of replicas and update strategies. It manages ReplicaSets and allows for version control and rolling updates of applications.

9. What is a Service in Kubernetes?

A Service is an abstraction that defines a logical set of Pods and a policy to access them. Services enable Pods to communicate with each other and expose the application to external traffic.

10. What is a ConfigMap in Kubernetes?

ConfigMaps are used to inject configuration data into containers. They allow you to decouple configuration details from container images, making the system more flexible and portable.

Intermediate Kubernetes Questions

11. What is a StatefulSet in Kubernetes?

A StatefulSet is a Kubernetes workload API object used to manage stateful applications. It provides unique identities and stable, persistent storage for each Pod, ensuring order and persistence across Pod restarts and scaling.

12. What are Kubernetes Secrets?

Secrets are objects in Kubernetes used to store sensitive information such as passwords, OAuth tokens, and SSH keys. Secrets can be mounted as volumes or exposed as environment variables to Pods.

13. What is Horizontal Pod Autoscaler (HPA)?

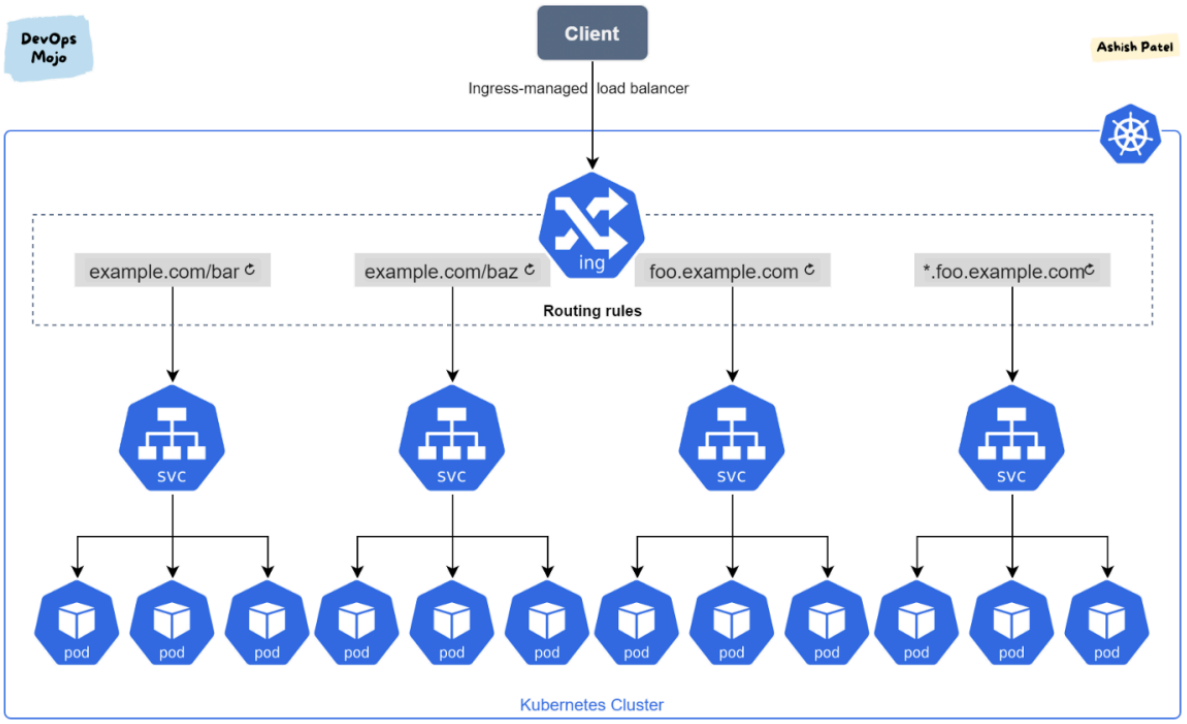
The Horizontal Pod Autoscaler automatically scales the number of Pods in a deployment based on observed CPU utilization, memory usage, or other custom metrics.

14. What are DaemonSets in Kubernetes?

A DaemonSet ensures that a copy of a Pod runs on all (or some) worker nodes. It is useful for running background processes like log collection or monitoring on every node.

15. What is the difference between a Deployment and a StatefulSet?

- **Deployment**: Best for stateless applications, provides scaling and rolling updates.
- **StatefulSet**: Best for stateful applications requiring persistent storage and ordered deployment.



16. What is Ingress in Kubernetes?

Ingress is an API object that manages external access to services in a cluster, typically HTTP/HTTPS traffic. It allows routing rules and provides more control over external access compared to a Service.

17. What are the types of Kubernetes Services?

- **ClusterIP**: Exposes the Service only within the cluster.
- **NodePort**: Exposes the Service on a static port on each node's IP.
- **LoadBalancer**: Exposes the Service externally using a cloud provider's load balancer.
- **ExternalName**: Maps a Service to an external DNS name.

18. How do you perform a Rolling Update in Kubernetes?

A rolling update can be performed using Kubernetes `kubectl apply` or through updating the deployment manifest. It replaces Pods incrementally with new versions to avoid downtime.

19. What is a Helm Chart?

Helm is a package manager for Kubernetes that simplifies the deployment of applications. Helm charts are a collection of files that describe a set of Kubernetes resources.

20. What is the difference between Helm and Kustomize?

- **Helm**: Template-based package manager that allows creating, managing, and upgrading Kubernetes applications.
- **Kustomize**: Native Kubernetes tool for customizing raw, template-free YAML files for different environments.

Advanced Kubernetes Questions

21. What is a Kubernetes Operator?

An Operator is a method of packaging, deploying, and managing a Kubernetes application. Operators extend Kubernetes' functionality by managing the lifecycle of complex applications and automating common tasks such as configuration, scaling, and backup.

22. Explain the Kubernetes Control Plane.

The Kubernetes Control Plane is the collection of processes that control Kubernetes nodes. It consists of the API server, controller manager, scheduler, etcd, and cloud controller manager. It ensures that the desired state of the system matches the current state.

23. How do you monitor a Kubernetes cluster?

You can monitor a Kubernetes cluster using tools like **Prometheus**, **Grafana**, **Kubernetes Dashboard**, and **ELK stack**. These tools help in tracking metrics, resource usage, logs, and cluster health.

24. What are Persistent Volumes (PV) and Persistent Volume Claims (PVC)?

- **PV**: A storage resource in a Kubernetes cluster that is provisioned by an admin or dynamically.
- **PVC**: A request for storage by a user that links to a PV to provide persistent storage for Pods.

25. What is the difference between CNI, CSI, and CRI in Kubernetes?

- **CNI (Container Network Interface)**: Standard interface for network configuration.
- **CSI (Container Storage Interface)**: Standard interface for storage configuration.
- **CRI (Container Runtime Interface)**: Standard interface for interacting with container runtimes like Docker, CRI-O, or containerd.

26. Explain Network Policies in Kubernetes.

Network Policies are used to control traffic between Pods, namespaces, or external entities. They allow you to define rules for what type of traffic is allowed to flow in and out of the Pods.

27. How do you secure a Kubernetes cluster?

- Use **RBAC** (Role-Based Access Control) to limit access.
- Enable **Pod Security Policies** to enforce security standards.
- Use **Secrets** for managing sensitive information.
- Enable **Network Policies** to control traffic.
- Keep Kubernetes and its components up-to-date.

28. What is a Service Mesh in Kubernetes?

A Service Mesh, like Istio or Linkerd, provides traffic management, observability, and security between microservices within a Kubernetes cluster. It abstracts the communication between services and offers features like load balancing, service discovery, and failure recovery.

29. What is etcd in Kubernetes?

etcd is a distributed, consistent key-value store used by Kubernetes to store all cluster-related data, including the cluster state, configuration, and metadata. It ensures consistency across the Kubernetes control plane.

30. What are taints and tolerations in Kubernetes?

- **Taints**: Prevent Pods from being scheduled on nodes.
- **Tolerations**: Allow Pods to be scheduled on nodes with matching taints.

Scenario-Based Kubernetes Questions

31. How would you troubleshoot a Kubernetes Pod stuck in `CrashLoopBackOff` state?

- Check logs using `kubectl logs <pod-name>`.
- Investigate the events using `kubectl describe pod <pod-name>`.
- Ensure there are no misconfigurations in the Pod definition (like wrong image names, missing config maps, etc.).

32. What would you do if a Kubernetes node becomes NotReady?

- Use `kubectl get nodes` to check the node status.

- Inspect the node using `kubectl describe node <node-name>`.
- Check logs on the node (`journalctl`, `dmesg`).
- Investigate kubelet and networking issues.

33. How would you handle a high CPU usage issue in your Kubernetes cluster?

- Use `kubectl top` to check Pod and node-level resource consumption.
- Set resource limits and requests in the Pod definitions.
- Implement Horizontal Pod Autoscaler (HPA) for automatic scaling.

Follow me on **LinkedIn** for more 😊