**Interview Questions (Kubernetes)**

**Basic Questions**

1. **What is Kubernetes?**

Answer: Kubernetes is an open-source platform designed to automate deploying, scaling, and operating containerized applications.

1. **What are the main components of Kubernetes architecture?**

Answer: The main components are the API server, etcd, controller manager, scheduler, kubelet, kube-proxy, and the container runtime.

1. **What is a Pod in Kubernetes?**

Answer: A Pod is the smallest and simplest Kubernetes object, representing a single instance of a running process in your cluster. Pods can contain one or more containers.

1. **What is a Node in Kubernetes?**

Answer: A Node is a worker machine in Kubernetes, where Pods are deployed. It can be either a virtual or physical machine.

1. **What is the purpose of the kubelet?**

Answer: The kubelet is the primary "node agent" that runs on each node. It ensures that containers are running in a Pod.

1. **What does kubectl do?**

Answer: kubectl is a command-line tool for interacting with the Kubernetes API server to manage Kubernetes clusters and applications.

1. **Explain the role of the Kubernetes API server.**

Answer: The API server is the front end for the Kubernetes control plane, handling RESTful requests to manage cluster resources.

1. **What is etcd in Kubernetes?**

Answer: etcd is a distributed key-value store used for storing all cluster data, including configuration and state information.

1. **What is a ReplicaSet?**

Answer: A ReplicaSet ensures that a specified number of pod replicas are running at any given time.

1. **What is a Deployment in Kubernetes?**

Answer: A Deployment provides declarative updates for Pods and ReplicaSets, allowing you to define the desired state of your applications.

**Intermediate Questions**

1. **What is a Namespace in Kubernetes?**

Answer: Namespaces provide a way to divide cluster resources between multiple users or teams, offering a scope for names and resource management.

1. **How does the Kubernetes Scheduler work?**

Answer: The Scheduler assigns pods to nodes based on resource availability and other constraints by observing unassigned pods and selecting appropriate nodes for them.

1. **What is a Service in Kubernetes and what are its types?**

Answer: A Service is an abstraction that defines a logical set of Pods and a policy to access them. Types include ClusterIP, NodePort, LoadBalancer, and ExternalName.

1. **What is the difference between a Deployment and a StatefulSet?**

Answer: Deployments are used for stateless applications, providing easy scaling and updates, while StatefulSets are used for stateful applications, ensuring ordered deployment and unique network identities.

1. **What is an Ingress in Kubernetes?**

Answer: An Ingress is an API object that manages external access to services within a cluster, typically HTTP and HTTPS, providing routing rules and load balancing.

1. **How do you perform a rolling update in Kubernetes?**

Answer: A rolling update is performed using a Deployment by updating the image or other properties. Kubernetes gradually replaces the old Pods with new ones, ensuring zero downtime.

1. **What is a ConfigMap in Kubernetes?**

Answer: A ConfigMap is used to store configuration data in key-value pairs. It decouples configuration artifacts from image content to keep containerized applications portable.

1. **What is a DaemonSet and when would you use it?**

Answer: A DaemonSet ensures that all (or some) nodes run a copy of a Pod. It is used for background tasks like log collection and monitoring agents.

1. **Explain the concept of Volumes in Kubernetes.**

Answer: Volumes provide a way for containers in a Pod to share data and persist data beyond the lifecycle of a Pod. Different volume types include emptyDir, hostPath, and PersistentVolume.

1. **What is the role of the kube-proxy?**

Answer: kube-proxy maintains network rules on nodes, enabling network communication to your Pods from network sessions inside or outside of your cluster.

**Advanced Questions**

1. **What is the purpose of the Controller Manager in Kubernetes?**

Answer: The Controller Manager runs controllers that regulate the state of the cluster, such as the Node Controller, Replication Controller, and Endpoints Controller.

1. **How does Kubernetes ensure high availability?**

Answer: High availability is ensured through multiple instances of control plane components, automatic rescheduling of pods, and self-healing features.

1. **What is the difference between kubectl apply and kubectl create?**

Answer: kubectl create is used to create a resource from a file, while kubectl apply creates or updates a resource from a file, allowing for declarative configuration.

1. **What is a PersistentVolume (PV) and PersistentVolumeClaim (PVC)?**

Answer: A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator. A PersistentVolumeClaim (PVC) is a request for storage by a user.

1. **Explain the concept of Labels and Selectors in Kubernetes.**

Answer: Labels are key-value pairs attached to objects. Selectors are used to filter and group objects based on their labels.

1. **What is a Sidecar container pattern?**

Answer: A Sidecar container is a utility container in a Pod that enhances or extends the functionality of the main container, such as logging or proxying.

1. **How does Horizontal Pod Autoscaler (HPA) work?**

Answer: HPA automatically scales the number of pods in a deployment or replica set based on observed CPU utilization or other select metrics.

1. **What is NetworkPolicy in Kubernetes?**

Answer: NetworkPolicy is a specification of how groups of pods are allowed to communicate with each other and with other network endpoints.

1. **What are Init Containers and when would you use them?**

Answer: Init containers are specialized containers that run before app containers in a Pod. They can perform initialization tasks such as setting up the environment or loading configuration.

1. **What is the difference between a Job and a CronJob in Kubernetes?**

Answer: A Job creates one or more Pods and ensures that a specified number of them successfully complete. A CronJob creates Jobs on a time-based schedule.

**Practical and Scenario-Based Questions**

1. **How would you debug a failing Pod in Kubernetes?**

Answer: Check the pod status with kubectl get pods, describe the pod with kubectl describe pod <pod\_name>, and check the logs with kubectl logs <pod\_name>.

1. **What would you do if a node becomes unresponsive in a Kubernetes cluster?**

Answer: Kubernetes automatically tries to reschedule the pods to other available nodes. You can check node status with kubectl get nodes and investigate with kubectl describe node <node\_name>.

1. **How do you manage secrets in Kubernetes?**

Answer: Use the Kubernetes Secrets object to manage sensitive information. Create secrets using kubectl create secret, and reference them in pod specifications.

1. **How do you scale a Deployment in Kubernetes?**

Answer: Use kubectl scale command with the desired number of replicas, or update the deployment specification with the new replica count.

1. **How would you roll back a Deployment?**

Answer: Use kubectl rollout undo deployment/<deployment\_name> to roll back to the previous version.

1. **Explain how you would set up a multi-container Pod.**

Answer: Define a Pod specification with multiple containers in the spec.containers section. Ensure the containers share volumes and network settings as needed.

1. **What is the use of kubectl exec command?**

Answer: kubectl exec is used to execute commands in a container within a Pod, useful for debugging or performing tasks inside running containers.

1. **What are Taints and Tolerations in Kubernetes?**

Answer: Taints are applied to nodes to repel certain pods, and tolerations are applied to pods to allow them to schedule on nodes with matching taints.

1. **Explain the concept of Resource Quotas in Kubernetes.**

Answer: Resource Quotas are used to limit the resource consumption (CPU, memory) within a namespace, ensuring fair usage among different teams or applications.

1. **How would you expose a Deployment to external traffic?**

Answer: Create a Service of type NodePort or LoadBalancer to expose the Deployment. Use kubectl expose deployment <deployment\_name> to create a service.