KUBERNETES CONTAINER ORCHESTRATION

100 INTERVIEW QUESTIONS AND ANSWERS

@vaibhavsaini

Question 1: What is Kubernetes?
-- Answer:
Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications.

Question 2: What are the key components of Kubernetes architecture?
-- Answer:
Kubernetes has key components like Master Node (API Server, Controller Manager, Scheduler, etcd) and Worker Nodes (Kubelet, Kube Proxy, Container Runtime).

Question 3: Explain what is a Pod in Kubernetes.
-- Answer:
A Pod is the smallest deployable unit in Kubernetes, representing a single instance of a running process. It can contain one or more containers.

Question 4: What is the purpose of a Deployment in Kubernetes?
-- Answer:
A Deployment is used to ensure the desired number of instances of a Pod is running, providing features like scaling, rolling updates, and version management.

Question 5: How does Kubernetes manage networking between Pods?
-- Answer:
Kubernetes uses a networking model where each Pod gets its own IP address, and Pods can communicate with each other using their IPs or DNS names.

Question 6: What is a Service in Kubernetes?
-- Answer:
A Service defines a logical set of Pods and a policy by which to access them. It provides load balancing and service discovery for Pods.

Question 7: Explain Horizontal Pod Autoscaling in Kubernetes.
-- Answer:
Horizontal Pod Autoscaling automatically adjusts the number of Pod replicas in a Deployment or ReplicaSet based on observed CPU utilization or custom metrics.

Question 8: What is a StatefulSet in Kubernetes?
-- Answer:
A StatefulSet is used to manage stateful applications by providing stable network identifiers and persistent storage for each Pod.

Question 9: Describe Kubernetes ConfigMap and Secrets.
-- Answer:
ConfigMaps store configuration data as key-value pairs, while Secrets store sensitive information like passwords, tokens, and keys.

Question 10: How do you expose a Kubernetes Service to the external world?
-- Answer:
You can expose a Service using NodePort, LoadBalancer, or Ingress depending on the requirements.

Question 11: What is a Namespace in Kubernetes?

-- Answer:

A Namespace provides a way to logically divide cluster resources. It allows multiple teams or projects to share a Kubernetes cluster without colliding with each other.

Question 12: Explain Kubernetes Rolling Updates.

-- Answer:

Rolling Updates in Kubernetes involve updating Pods one by one to ensure there is no downtime. It replaces old Pods with new ones gradually.

Question 13: What is a DaemonSet in Kubernetes?

-- Answer:

A DaemonSet ensures that all or some nodes run a copy of a Pod. It is useful for tasks that need to run on all or specific nodes, like log collection agents.

Question 14: What is a PVC (PersistentVolumeClaim) in Kubernetes?

-- Answer:

A PVC is a request for storage by a user in Kubernetes. It allows Pods to request storage resources without needing to know the details of the underlying storage.

Question 15: Explain Kubernetes Helm.

-- Answer:

Helm is a package manager for Kubernetes that allows you to define, install, and upgrade even the most complex Kubernetes applications.

Question 16: What is a Kubernetes Operator?

-- Answer:

A Kubernetes Operator is an application-specific controller that extends the Kubernetes API to create, configure, and manage instances of complex stateful applications.

Question 17: How does Kubernetes handle high availability for control plane components?

-- Answer:

Kubernetes achieves control plane high availability using techniques like leader election, replication, and etcd's distributed data store.

Question 18: What is the purpose of a ConfigMap and Secret in Kubernetes?

-- Answer:

ConfigMaps store non-sensitive configuration data, while Secrets store sensitive data like passwords. Both allow you to decouple configuration from the container image.

Question 19: Explain the difference between a ReplicaSet and a Deployment in Kubernetes.

-- Answer:

A ReplicaSet ensures a specified number of replicas of a Pod are running, but a Deployment provides additional features like rolling updates and scaling.

Question 20: What is Kubernetes Ingress?

-- Answer:

Ingress is an API object that manages external access to services within a cluster. It provides routing and load balancing of incoming traffic to different services.

Question 21: What is a Pod in Kubernetes?
-- Answer:
A Pod is the smallest deployable unit in Kubernetes. It can contain one or more containers that share the same network namespace and storage.

Question 22: Explain the concept of a StatefulSet in Kubernetes.
-- Answer:
A StatefulSet is used for managing stateful applications. It provides guarantees about the ordering and uniqueness of Pods, which is important for databases.

Question 23: What is Horizontal Pod Autoscaling (HPA) in Kubernetes?
-- Answer:
HPA automatically scales the number of Pods in a deployment based on CPU utilization or custom metrics. It ensures that the application can handle varying loads.

Question 24: How does Kubernetes handle service discovery and load balancing?
-- Answer:
Kubernetes uses DNS to provide service discovery for Pods. Services are assigned a DNS name, and requests to that name are balanced across the healthy Pods.

Question 25: Explain the concept of a Deployment strategy in Kubernetes.
-- Answer:
A Deployment strategy defines how updates are applied to Pods. Strategies include RollingUpdate, Recreate, and Blue-Green, each with specific behavior during updates.

Question 26: What is a Pod Security Policy in Kubernetes?
-- Answer:
A Pod Security Policy defines a set of conditions that Pods must satisfy to be accepted in the system. It is used to restrict what a Pod can do.

Question 27: How does Kubernetes manage container networking?
-- Answer:
Kubernetes creates a unique IP address for each Pod, and containers within a Pod share the same network namespace. Services provide load balancing and connectivity.

Question 28: What is a HorizontalPodAutoscaler in Kubernetes?
-- Answer:
A HorizontalPodAutoscaler (HPA) automatically adjusts the number of replicas in a deployment based on observed CPU utilization or other custom metrics.

Question 29: Explain Kubernetes Cluster Autoscaling.
-- Answer:
Cluster Autoscaling adjusts the number of nodes in a cluster based on resource demands. It ensures that there are enough resources available for all Pods.

Question 30: What are Kubernetes Labels and Selectors?
-- Answer:
Labels are key-value pairs attached to Kubernetes objects. Selectors are used to identify a set of objects based on their labels, facilitating grouping and organization.

Question 31: What is a Kubernetes Namespace?

-- Answer:

A Namespace provides a way to logically divide cluster resources into distinct virtual clusters. It helps manage and isolate resources for different teams or projects.

Question 32: Explain the purpose of Kubernetes ConfigMap and Secret.

-- Answer:

ConfigMap stores configuration data as key-value pairs, while Secret stores sensitive information like passwords. They are used to separate configuration from application code.

Question 33: What is the role of a Service in Kubernetes?

-- Answer:

A Service in Kubernetes exposes a set of Pods as a network service. It provides load balancing and ensures that Pods are reachable by other parts of the application.

Question 34: How does Kubernetes handle application updates?

-- Answer:

Kubernetes supports rolling updates for Deployments, where new Pods are gradually deployed while old Pods are terminated. This ensures zero-downtime updates.

Question 35: Explain the use of Kubernetes Ingress.

-- Answer:

Ingress manages external access to services within the cluster. It acts as an API gateway and allows routing based on hostnames, paths, or other rules.

Question 36: What is the role of a DaemonSet in Kubernetes?

-- Answer:

A DaemonSet ensures that all (or a subset of) Nodes run a copy of a Pod. It's commonly used for running monitoring agents or log collectors on every node.

Question 37: What is the difference between a Job and a CronJob in Kubernetes?

-- Answer:

A Job runs a set of Pods to completion, while a CronJob creates Jobs on a schedule. CronJobs are used for periodically running tasks.

Question 38: Explain Kubernetes Network Policies.

-- Answer:

Network Policies define rules for communication between Pods. They allow you to control ingress and egress traffic based on labels and namespaces.

Question 39: What is a Kubernetes Operator?

-- Answer:

An Operator is a method of packaging, deploying, and managing a Kubernetes application. It extends the Kubernetes API to automate complex tasks.

Question 40: How can you scale a StatefulSet in Kubernetes?

-- Answer:

You can scale a StatefulSet using the `kubectl scale` command or by updating the `replicas` field in the StatefulSet definition. Kubernetes will handle the rest.

Question 41: What is Kubernetes Pod?
-- Answer:
A Pod is the smallest deployable unit in Kubernetes, representing a single instance of a running process in a cluster. It can contain one or more containers.

Question 42: Explain Kubernetes Deployment.
-- Answer:
A Deployment in Kubernetes is a higher-level abstraction that manages the creation and scaling of ReplicaSets. It ensures the desired number of replicas are running at all times.

Question 43: What is Horizontal Pod Autoscaling in Kubernetes?
-- Answer:
Horizontal Pod Autoscaling automatically scales the number of Pods in a Deployment or ReplicaSet based on observed CPU utilization or other custom metrics.

Question 44: What are Kubernetes Labels and Selectors?
-- Answer:
Labels are key-value pairs attached to resources. Selectors are used to identify resources based on labels, allowing grouping and efficient resource management.

Question 45: Explain Kubernetes Taints and Tolerations.
-- Answer:
Taints are applied to Nodes to repel Pods unless the Pods have matching Tolerations. They are used to ensure certain Pods run on specific Nodes.

Question 46: What is Kubernetes Helm?
-- Answer:
Helm is a package manager for Kubernetes applications. It helps define, install, and upgrade even the most complex Kubernetes applications.

Question 47: Explain Kubernetes Persistent Volumes (PV) and Persistent Volume Claims (PVC).
-- Answer:
Persistent Volumes are storage resources in a cluster, while Persistent Volume Claims are requests for storage by users. PVCs bind to PVs to provide storage.

Question 48: What is Kubernetes StatefulSet?
-- Answer:
A StatefulSet is similar to a Deployment but maintains a sticky identity for each Pod. It's used for applications that require stable and unique network identities.

Question 49: How does Kubernetes handle Node failures?
-- Answer:
Kubernetes detects Node failures and reschedules affected Pods onto healthy Nodes. The desired state defined in YAML files ensures automatic recovery.

Question 50: Explain Kubernetes Rolling Back a Deployment.
-- Answer:
To roll back a Deployment to a previous version, use the `kubectl rollout undo` command. Kubernetes will ensure the desired state matches the specified revision.

Question 41: What is Kubernetes Pod?

-- Answer:

A Pod is the smallest deployable unit in Kubernetes, representing a single instance of a running process in a cluster. It can contain one or more containers.

Question 42: Explain Kubernetes Deployment.

-- Answer:

A Deployment in Kubernetes is a higher-level abstraction that manages the creation and scaling of ReplicaSets. It ensures the desired number of replicas are running at all times.

Question 43: What is Horizontal Pod Autoscaling in Kubernetes?

-- Answer:

Horizontal Pod Autoscaling automatically scales the number of Pods in a Deployment or ReplicaSet based on observed CPU utilization or other custom metrics.

Question 44: What are Kubernetes Labels and Selectors?

-- Answer:

Labels are key-value pairs attached to resources. Selectors are used to identify resources based on labels, allowing grouping and efficient resource management.

Question 45: Explain Kubernetes Taints and Tolerations.

-- Answer:

Taints are applied to Nodes to repel Pods unless the Pods have matching Tolerations. They are used to ensure certain Pods run on specific Nodes.

Question 46: What is Kubernetes Helm?

-- Answer:

Helm is a package manager for Kubernetes applications. It helps define, install, and upgrade even the most complex Kubernetes applications.

Question 47: Explain Kubernetes Persistent Volumes (PV) and Persistent Volume Claims (PVC).

-- Answer:

Persistent Volumes are storage resources in a cluster, while Persistent Volume Claims are requests for storage by users. PVCs bind to PVs to provide storage.

Question 48: What is Kubernetes StatefulSet?

-- Answer:

A StatefulSet is similar to a Deployment but maintains a sticky identity for each Pod. It's used for applications that require stable and unique network identities.

Question 49: How does Kubernetes handle Node failures?

-- Answer:

Kubernetes detects Node failures and reschedules affected Pods onto healthy Nodes. The desired state defined in YAML files ensures automatic recovery.

Question 50: Explain Kubernetes Rolling Back a Deployment.

-- Answer:

To roll back a Deployment to a previous version, use the `kubectl rollout undo` command. Kubernetes will ensure the desired state matches the specified revision.

Question 51: What is Kubernetes ConfigMap?

-- Answer:

A ConfigMap is used to store non-sensitive configuration data in key-value pairs. It's often used to configure applications running in Pods.

Question 52: Explain Kubernetes Secrets.

-- Answer:

Kubernetes Secrets are used to store sensitive information like passwords, tokens, or keys. They're stored securely and can be used by Pods.

Question 53: What is Kubernetes Service?

-- Answer:

A Kubernetes Service exposes a set of Pods as a network service. It provides a stable IP address and DNS name for access.

Question 54: Describe Kubernetes Ingress.

-- Answer:

Kubernetes Ingress manages external access to services within a cluster. It handles HTTP and HTTPS routing, allowing traffic to reach the correct service.

Question 55: What is Kubernetes NetworkPolicy?

-- Answer:

Kubernetes NetworkPolicy is used to control network traffic to and from Pods. It defines how Pods can communicate with each other and other network endpoints.

Question 56: Explain Kubernetes Namespace.

-- Answer:

Kubernetes Namespace provides a way to divide cluster resources into isolated virtual clusters. It's often used to separate projects or teams.

Question 57: What is Kubernetes DaemonSet?

-- Answer:

A DaemonSet ensures that a copy of a Pod runs on each Node in a cluster. It's used for tasks like monitoring or logging that should run on every Node.

Question 58: How does Kubernetes handle rolling updates?

-- Answer:

Kubernetes performs rolling updates by gradually replacing old Pods with new ones. This ensures minimal disruption while deploying new versions.

Question 59: What is Kubernetes Helm Chart?

-- Answer:

A Helm Chart is a collection of files describing a Kubernetes application. It simplifies application deployment, management, and versioning.

Question 60: Explain Kubernetes Custom Resource Definitions (CRDs).

-- Answer:

CRDs extend Kubernetes API to include custom resources. They allow users to define their own API objects, adding new functionality to the cluster.

Question 61: Describe Kubernetes ServiceAccount.

-- Answer:

A ServiceAccount is used to provide Pods with necessary credentials to access the Kubernetes API or other resources. It follows the principle of least privilege.

Question 62: What are Kubernetes Operators?

-- Answer:

Operators are software extensions that simplify the management of applications on Kubernetes. They automate complex tasks, ensuring applications are running as intended.

Question 63: How does Kubernetes handle scaling of resources?

-- Answer:

Kubernetes provides Horizontal Pod Autoscaling and Vertical Pod Autoscaling to automatically adjust the number of Pods or resources assigned to a Pod based on demand.

Question 64: What is Kubernetes Kubeconfig?

-- Answer:

Kubeconfig is a configuration file used to specify clusters, users, and contexts in a Kubernetes environment. It allows users to interact with multiple clusters.

Question 65: Explain Kubernetes Cluster Autoscaler.

-- Answer:

Cluster Autoscaler adjusts the size of the cluster by adding or removing Nodes based on demand. It ensures optimal resource utilization and application availability.

Question 66: What is Kubernetes Service Discovery?

-- Answer:

Kubernetes Service Discovery allows applications to discover other services within a cluster. It uses DNS to provide a simple and consistent way to connect services.

Question 67: Describe Kubernetes Resource Quotas.

-- Answer:

Resource Quotas limit the amount of CPU, memory, and other resources a Namespace can consume. They prevent resource exhaustion and ensure fair allocation.

Question 68: What is Kubernetes Network Plugin (CNI)?

-- Answer:

A Container Network Interface (CNI) plugin is responsible for configuring network connectivity for Pods. It enables communication between Pods across Nodes.

Question 69: Explain Kubernetes Role-Based Access Control (RBAC).

-- Answer:

RBAC is a method to control access to Kubernetes resources based on roles and permissions. It restricts users and groups to perform authorized actions.

Question 70: How does Kubernetes handle storage volumes?

-- Answer:

Kubernetes supports storage volumes to store data independently of the lifecycle of Pods. It allows sharing data between Pods and ensures data persistence.

Question 71: What is Kubernetes StatefulSet?

-- Answer:

StatefulSets manage the deployment and scaling of stateful applications. They maintain stable network identities and persistent storage for Pods.

Question 72: Explain Kubernetes Rolling Updates and Blue-Green Deployments.

-- Answer:

Rolling Updates replace Pods in a controlled manner to deploy new versions. Blue-Green Deployments use separate environments for new and old versions, allowing quick rollback.

Question 73: What is Kubernetes Helm?

-- Answer:

Helm is a package manager for Kubernetes applications. It streamlines the installation, management, and deployment of complex applications.

Question 74: Describe Kubernetes Horizontal Pod Autoscaling.

-- Answer:

Horizontal Pod Autoscaling adjusts the number of Pods in a deployment or replica set based on CPU utilization or custom metrics. It ensures optimal resource utilization.

Question 75: What is Kubernetes Volume?

-- Answer:

A Volume in Kubernetes is used to store data shared between Containers in a Pod. It allows data persistence and sharing even when Pods are rescheduled.

Question 76: Explain Kubernetes Taints and Tolerations.

-- Answer:

Taints are used to repel Pods from specific Nodes. Tolerations in Pods allow them to tolerate the taints, ensuring they are scheduled on tainted Nodes.

Question 77: What is Kubernetes Horizontal Pod Autoscaling?

-- Answer:

Horizontal Pod Autoscaling automatically adjusts the number of replicas of a Deployment or ReplicaSet based on observed CPU utilization or custom metrics.

Question 78: Describe Kubernetes Vertical Pod Autoscaling.

-- Answer:

Vertical Pod Autoscaling adjusts the resource requests and limits of Containers in a Pod to optimize resource utilization. It helps Pods use their allocated resources more effectively.

Question 79: What is Kubernetes Probes?

-- Answer:

Probes are used to determine the health of a Container in a Pod. Liveness probes check if a Container is running, while readiness probes check if it's ready to receive traffic.

Question 80: Explain Kubernetes Readiness and Liveness Probes.

-- Answer:

Readiness Probes ensure that a Pod is ready to serve traffic before receiving requests. Liveness Probes detect and restart Containers that have failed or hung.

Question 81: What is Kubernetes Cluster Architecture?
-- Answer:
Kubernetes Cluster consists of the Control Plane and Nodes. Control Plane manages the cluster, while Nodes run Containers. ETCD stores configuration data.

Question 82: Describe Kubernetes Control Plane Components.
-- Answer:
Control Plane includes:
- API Server: Exposes Kubernetes API
- etcd: Consistent and highly available key-value store
- Scheduler: Assigns Pods to Nodes
- Controller Manager: Ensures desired state of cluster

Question 83: Explain Kubernetes Node Components.
-- Answer:
Node components include:
- Kubelet: Ensures Containers are running in a Pod
- Kube Proxy: Maintains network rules for Pods
- Container Runtime: Runs Containers (e.g., Docker)

Question 84: What is Kubernetes ETCD?
-- Answer:
ETCD is a distributed key-value store used by Kubernetes to store all cluster data, configuration, and state information.

Question 85: Describe Kubernetes API Server.
-- Answer:
Kubernetes API Server exposes the Kubernetes API, which provides programmatic access to the cluster and is used for creating and managing resources.

Question 86: What is Kubernetes Namespace?
-- Answer:
A Namespace is a virtual cluster within a physical cluster, used to separate resources and control access to them. It provides a scope for names.

Question 87: Explain Kubernetes Pod Security Policies.
-- Answer:
Pod Security Policies are used to control the security settings of Pods. They define conditions Pods must meet to be accepted in a Namespace.

Question 88: What is Kubernetes HorizontalPodAutoscaler?
-- Answer:
HorizontalPodAutoscaler automatically scales the number of Pods in a deployment or replica set based on observed CPU utilization or custom metrics.

Question 89: Describe Kubernetes VerticalPodAutoscaler.
-- Answer:
VerticalPodAutoscaler adjusts the resource requests and limits of Containers in a Pod to optimize resource utilization, improving application performance.

Question 90: What is Kubernetes Control Plane Node?
-- Answer:
A Control Plane Node (also known as Master Node) hosts the Control Plane components like API Server, Scheduler, Controller Manager, and etcd.

Question 91: What is Kubernetes Node?
-- Answer:
A Node in Kubernetes is a worker machine that runs Containers. It's managed by the Control Plane and can run one or multiple Pods.

Question 92: Explain Kubernetes Pod.
-- Answer:
A Pod is the smallest deployable unit in Kubernetes, representing a single instance of a process. It can contain one or more tightly coupled Containers.

Question 93: What is Kubernetes Service?
-- Answer:
A Service in Kubernetes provides a way to expose a set of Pods as a network service, allowing them to communicate with other Pods within or outside the cluster.

Question 94: Describe Kubernetes Ingress.
-- Answer:
Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. It manages external access and load balancing.

Question 95: What is Kubernetes ConfigMap?
-- Answer:
A ConfigMap in Kubernetes is used to store configuration data separate from application code. It allows you to manage configuration changes without changing code.

Question 96: Explain Kubernetes Secret.
-- Answer:
A Secret in Kubernetes is used to store sensitive information like passwords and tokens. It ensures secure storage and transmission of sensitive data.

Question 97: What is Kubernetes DaemonSet?
-- Answer:
DaemonSet ensures that a copy of a Pod is running on all or selected nodes. It's useful for deploying monitoring agents or other system-level tasks.

Question 98: Describe Kubernetes Network Policies.
-- Answer:
Network Policies in Kubernetes control the communication between Pods and define how they can communicate based on labels and namespaces.

Question 99: What is Kubernetes Persistent Volume?
-- Answer:
A Persistent Volume in Kubernetes is used to store data beyond the lifecycle of Pods. It provides data persistence even when Pods are rescheduled.

Question 100: Explain Kubernetes Persistent Volume Claim.
-- Answer:
A Persistent Volume Claim in Kubernetes is a request for storage resources by a user. It's used to dynamically provision Persistent Volumes for Pods.