

◆ Beginner Level (1-30 Questions)

1. What is Kubernetes?

Answer: Kubernetes (K8s) is an open-source container orchestration platform that automates deployment, scaling, and management of containerized applications.

2. What are the main components of Kubernetes architecture?

Answer:

- ✓ **Master Node** – Controls the cluster
- ✓ **Worker Nodes** – Run application workloads
- ✓ **API Server** – Entry point for cluster communication
- ✓ **etcd** – Stores cluster configuration data
- ✓ **Scheduler** – Assigns workloads to nodes
- ✓ **Controller Manager** – Ensures cluster state

3. What is a Pod in Kubernetes?

Answer: A **Pod** is the smallest deployable unit in Kubernetes that can contain **one or more** containers sharing the same network and storage.

4. What is a Deployment in Kubernetes?

Answer: A **Deployment** ensures a specified number of pod replicas are running and automatically handles updates and rollbacks.

📌 **Example Command:**

```
kubectl create deployment my-app --image=nginx
```

5. What are Kubernetes Services?

Answer: Services expose Pods and provide stable network access. Types:

- ✓ **ClusterIP** – Internal communication
- ✓ **NodePort** – Exposes service on a port across all nodes
- ✓ **LoadBalancer** – Provisions an external load balancer
- ✓ **Ingress** – Manages external access with routing rules

6. How do you list all running pods in Kubernetes?

Answer:

```
kubectl get pods
```

7. How do you scale a Deployment in Kubernetes?

Answer:

```
kubectl scale deployment my-app --replicas=5
```

8. What is a ConfigMap in Kubernetes?

Answer: A **ConfigMap** is used to store non-sensitive configuration data (e.g., environment variables).

◆ Intermediate Level (31-70 Questions)

31. What is a Persistent Volume (PV) and Persistent Volume Claim (PVC)?

Answer:

- ✓ **PV:** Storage provisioned in the cluster
- ✓ **PVC:** A request for storage from a PV

32. How does Kubernetes handle rolling updates?

Answer: Kubernetes uses **rolling updates** to gradually replace pods without downtime.

📌 **Example Command:**

```
kubectl set image deployment/my-app my-container=nginx:latest
```

33. What is a StatefulSet?

Answer: A StatefulSet manages stateful applications like databases, ensuring each pod has a stable identity and persistent storage.

34. How do you monitor a Kubernetes cluster?

Answer:

- ✓ **Prometheus & Grafana** – Metrics collection & visualization
- ✓ **Fluentd, Elasticsearch, Kibana (EFK Stack)** – Logging

35. What is the difference between Horizontal Pod Autoscaler (HPA) and Vertical Pod Autoscaler (VPA)?

Answer:

- ✓ **HPA** – Adjusts the number of pods based on CPU/memory usage
- ✓ **VPA** – Adjusts the resource requests/limits of a running pod

📌 **Enable HPA:**

```
kubectl autoscale deployment my-app --cpu-percent=50 --min=2 --max=10
```

36. How do you perform a rollback in Kubernetes?

Answer:

```
kubectl rollout undo deployment my-app
```

37. What are Network Policies in Kubernetes?

Answer: **Network Policies** control how Pods communicate within the cluster and with external networks.

📌 Example Network Policy YAML:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
spec:
  podSelector: {}
  policyTypes:
    - Ingress
```

◆ Expert Level (71-100 Questions)

71. What is the difference between Kubernetes Ingress and LoadBalancer?

Answer:

- ✅ **Ingress** – Manages external access via HTTP/S routing
- ✅ **LoadBalancer** – Provides direct external access (e.g., AWS ELB)

72. How do you troubleshoot a failing pod in Kubernetes?

Answer:

```
kubectl logs <pod-name>
kubectl describe pod <pod-name>
kubectl get events
```

73. What is a Custom Resource Definition (CRD)?

Answer: CRDs allow you to extend Kubernetes with custom objects and APIs.

📌 Example CRD Definition:

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: myresources.example.com
spec:
  group: example.com
  names:
    kind: MyResource
    plural: myresources
  scope: Namespaced
  versions:
    - name: v1
```

```
served: true
storage: true
```

74. How does Kubernetes handle multi-cluster management?

Answer: Tools like **KubeFed (Federation)**, **Cluster API**, and **Rancher** help manage multiple clusters efficiently.

75. What is the purpose of Kubernetes Admission Controllers?

Answer: They **validate and modify requests** before they reach the API server (e.g., PodSecurityPolicy).

76. What are Mutating and Validating Webhooks?

Answer:

- ✓ **Mutating Webhooks:** Modify API requests before they are persisted
- ✓ **Validating Webhooks:** Validate requests before execution

77. What is Kubernetes Chaos Engineering?

Answer: **Chaos Engineering** involves deliberately breaking systems to test resilience. **LitmusChaos & Chaos Mesh** are popular tools.

78. How do you secure Kubernetes clusters?

Answer:

- ✓ Enable **RBAC** (Role-Based Access Control)
- ✓ Use **Network Policies**
- ✓ Scan images for vulnerabilities
- ✓ Enable **Pod Security Policies**
- ✓ Use **Secrets** for sensitive data

79. What is Kubernetes Service Mesh?

Answer: A **Service Mesh** (e.g., Istio, Linkerd) manages service-to-service communication, including traffic control, security, and observability.

📌 **Install Istio:**

```
istioctl install --set profile=demo -y
```

80. What is Kubernetes Cluster Autoscaler?

Answer: **Cluster Autoscaler** automatically adjusts the number of nodes based on resource demands.

📌 **Enable Cluster Autoscaler on AWS:**

```
eksctl create cluster --name my-cluster --enable-cluster-autoscaler
```


◆ Expert Level (81-100 Questions)

81. What is the difference between a DaemonSet and a Deployment?

Answer:

- ✓ **DaemonSet** ensures a pod runs on **every node** (e.g., for monitoring/logging agents).
- ✓ **Deployment** manages replicas of pods and scales based on demand.

📌 **Example DaemonSet YAML:**

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: log-agent
spec:
  selector:
    matchLabels:
      app: log-agent
  template:
    metadata:
      labels:
        app: log-agent
    spec:
      containers:
        - name: log-agent
          image: fluentd
```

82. What is the difference between Helm and Kustomize?

Answer:

- ✓ **Helm** – Uses charts (templating) for package management.
- ✓ **Kustomize** – Uses overlays to manage Kubernetes manifests without templates.

📌 **Install a Helm chart:**

```
helm install my-release bitnami/nginx
```

📌 **Apply a Kustomization:**

```
kubectl apply -k ./kustomize-dir/
```

83. How does Kubernetes handle container logs?

Answer: Kubernetes stores logs in **/var/log/pods/** on the node. You can access them using:

```
kubectl logs <pod-name>
kubectl logs -f <pod-name> # Follow logs in real-time
```

84. What is a Sidecar pattern in Kubernetes?

Answer: A **Sidecar** is a secondary container running alongside the main container in a **Pod**, often used for logging, monitoring, or service mesh.

📌 **Example Use Case:** A sidecar container can collect logs and forward them to an external system.

85. What are Init Containers in Kubernetes?

Answer: **Init Containers** run **before** the main container in a Pod, used for setup tasks like fetching configs or waiting for dependencies.

📌 **Example Init Container YAML:**

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  initContainers:
    - name: setup
      image: busybox
      command: ['sh', '-c', 'echo Initializing... && sleep 5']
  containers:
    - name: app
      image: nginx
```

86. How do you check the resource usage of a Kubernetes Pod?

Answer:

```
kubectl top pod <pod-name> --containers
```

✅ You can also use **Metrics Server** or tools like **Prometheus + Grafana**.

87. What happens when a node fails in Kubernetes?

Answer:

- 1 Kubelet marks the node as **NotReady**
- 2 Pods are evicted and rescheduled on healthy nodes
- 3 Cluster Autoscaler may replace the failed node

📌 **Check node status:**

```
kubectl get nodes
```

88. What is the purpose of a Pod Disruption Budget (PDB)?

Answer: A **PDB** ensures that a minimum number of pods remain available during voluntary disruptions (e.g., rolling updates).

📌 **Example PDB YAML:**

```
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: my-app-pdb
spec:
  minAvailable: 2
  selector:
    matchLabels:
      app: my-app
```

89. What is the difference between Kubernetes and Docker Swarm?

Answer:

- ✅ **Kubernetes** – More complex, powerful, with advanced scaling and scheduling.
 - ✅ **Docker Swarm** – Simpler, tightly integrated with Docker but less feature-rich.
-

90. How do you delete all pods in a namespace?

Answer:

```
kubectl delete pods --all -n <namespace>
```

91. What are API Aggregation and Aggregated API Servers in Kubernetes?

Answer:

- ✅ API Aggregation allows extending the Kubernetes API using **custom API servers**.
- ✅ Aggregated API Servers register custom resources dynamically.

📌 Example: **metrics.k8s.io** is an aggregated API for Metrics Server.

92. What are Kubernetes Taints and Tolerations?

Answer:

- ✅ **Taints** prevent certain pods from being scheduled on a node.
- ✅ **Tolerations** allow pods to bypass taints.

📌 **Apply a taint to a node:**

```
kubectl taint nodes node1 key=value:NoSchedule
```

Apply a toleration in a Pod spec:

```
tolerations:  
- key: "key"  
  operator: "Equal"  
  value: "value"  
  effect: "NoSchedule"
```

93. How do you expose a Kubernetes Service externally?

Answer: Use **NodePort**, **LoadBalancer**, or **Ingress**.

Expose using LoadBalancer:

```
kubectl expose deployment my-app --type=LoadBalancer --port=80
```

94. How does Kubernetes handle pod communication within a cluster?

Answer: Pods communicate using **pod IPs** within a flat network. CNI plugins like **Calico**, **Flannel**, or **Cilium** manage network policies.

95. What is the role of etcd in Kubernetes?

Answer: **etcd** is a distributed key-value store that holds all cluster state and configurations.

Check etcd health:

```
kubectl exec etcd-master -n kube-system -- etcdctl cluster-health
```

96. What is the purpose of Kubernetes Garbage Collection?



Answer: Kubernetes automatically removes unused resources like **terminated pods**, **unreferenced images**, and **old ReplicaSets**.

Manually clean up old pods:

```
kubectl delete pod --field-selector=status.phase=Succeeded
```

97. What are the different ways to provide storage in Kubernetes?

Answer:

-  Persistent Volumes (PV)
-  Persistent Volume Claims (PVC)

- ✓ Storage Classes
- ✓ CSI (Container Storage Interface)

📌 Example PVC YAML:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

98. What is an Operator in Kubernetes?

Answer: An **Operator** automates complex application lifecycle management using **Custom Resource Definitions (CRDs)**.

📌 Example: **Prometheus Operator** manages Prometheus instances in Kubernetes.

99. How do you back up and restore a Kubernetes cluster?

Answer:

- ✓ Use **etcd snapshots** for backups:

```
ETCDCTL_API=3 etcdctl snapshot save backup.db
```

- ✓ Use tools like **Velero** for full cluster backup.
-

100. What is KEDA (Kubernetes Event-Driven Autoscaling)?

Answer: **KEDA** scales workloads based on events (e.g., Kafka messages, Prometheus metrics).

📌 Install KEDA:

```
helm repo add kedacore https://kedacore.github.io/charts
helm install keda kedacore/keda --namespace keda
```