Section 1: Pod and Controller Behavior

Q1 If an initContainer fails and the main container has restartPolicy: Never, what happens to the Pod?

> Answer: The Pod gets stuck in Init:Error or Init:CrashLoopBackOff and never starts the main container.

Q2 You delete pod-1 from a 3-replica StatefulSet. Do pods get renamed to fill the gap?

> Answer: No. Kubernetes recreates the exact same pod-1 with the same name, same PVC binding, and same DNS entry.

Q3 Does a DaemonSet automatically bypass NoSchedule taints on master or control-plane nodes?

> Answer: No. This is one of the most common wrong answers I hear. DaemonSets do not get automatic NoSchedule tolerance.

Q4 You push a new image while a rolling update is still in progress. Does Kubernetes wait for it to finish?

> Answer: No. Kubernetes immediately starts a new rollout for the latest image and scales down the in-progress ReplicaSet.

Q5 When a node goes NotReady, how long before Pods get evicted and can you control this per Pod?

> Answer: By default, Pods are evicted after 5 minutes (300 seconds). The kube-controller-manager controls this.

Q6 Can two containers in the same Pod bind to the same port?

> Answer: No. All containers in a Pod share one network namespace, which means they share the same port space.

Q7 With ReadWriteOnce access mode, can multiple Pods on the same node use the PVC at the same time?

> **Answer: ReadWriteOnce means the volume can be mounted read-write by a single node, not a single Pod.**

Q8 What happens to HPA if the metrics server goes down during a traffic spike?

> **Answer: HPA stops all scaling decisions, both up and down, and holds at the current replica count until metrics come back.**

Q9 Can you kubectl port-forward to a CrashLoopBackOff Pod?

> **Answer: Technically yes, but it will break every time the container crashes and restarts.**

Q10 What happens to mounted tokens and API access if you delete a ServiceAccount while Pods are still running?

> **Answer: Running Pods keep their existing tokens and continue working until those tokens expire, roughly 1 hour for projected service account tokens.**

Section 2: Production Troubleshooting

Q11 Can strict anti-affinity rules create a scheduling deadlock?

> **Answer: Yes. A classic one: you have 3 nodes, a required anti-affinity rule saying no two pods on the same node, and you try to schedule 4 replicas. The 4th sits Pending forever.**

Q12 A Job has parallelism: 3 and one Pod fails with restartPolicy: Never. Does the Job create a replacement?

> **Answer: Yes. The Job controller creates a new Pod to maintain the desired parallelism level.**

Q13 Can you modify resource requests and limits on a running Pod?

> Answer: In Kubernetes 1.27+, In-Place Pod Vertical Scaling allows modifying CPU without a restart in most cases. Memory is more restricted.

Q14 If a NetworkPolicy only specifies ingress rules, is egress traffic blocked?

> Answer: No. Egress is only blocked if you explicitly include 'Egress' in policyTypes.

Q15 If a shared Persistent Volume gets corrupted, can it cascade failures across multiple namespaces?

> Answer: For ReadWriteMany volumes backed by NFS or EFS, multiple PVCs across namespaces can reference the same underlying storage.

Section 3: Debugging Real Scenarios

Q16 Your Pod is in CrashLoopBackOff but logs show nothing. Where do you start?

> Answer: Start with kubectl describe pod and look at the Events section. It often tells you about OOMKilled, probe failures, or image issues before the container writes a single log line.

Q17 A StatefulSet Pod will not recreate properly after deletion. How do you fix this without losing data?

> Answer: First check PVC status. If it is stuck in Terminating, a finalizer is blocking it. Only remove the finalizer manually after confirming there is no active I/O.

Q18 Cluster Autoscaler is not scaling up even though Pods are Pending. What do you check?

> Answer: Most common reason: Pods have no resource requests. CA completely ignores Pods with zero requests.

Q19 A NetworkPolicy is breaking cross-namespace service communication. How do you debug and fix it?

> **Answer: Start with a default-deny baseline, then add explicit allow rules. Use namespaceSelector to target the source namespace and podSelector for specific Pods.**

Q20 A microservice needs to connect to an external database through a VPN inside the cluster. How do you architect this?

> **Answer: Deploy VPN gateway Pods as a Deployment with anti-affinity across availability zones, fronted by a ClusterIP Service. Apps connect through the Service, not directly to VPN pods.**

Section 4: Security and Architecture

Q21 How do you isolate tenants on a shared EKS cluster with proper security, quotas, and observability?

> **Answer: Layer your isolation. Use namespace-level ResourceQuotas and LimitRanges for compute governance, NetworkPolicies with default-deny per namespace for traffic, and RBAC with least-privilege roles scoped to each namespace.**

Q22 The kubelet keeps restarting on a specific node. How do you isolate the issue?

> **Answer: Cordon the node first so nothing new gets scheduled there. Then check system resources with top, df, and iostat.**

Q23 A critical production Pod keeps getting evicted due to node memory pressure. How do you prevent this?

> **Answer: Set equal requests and limits to get Guaranteed QoS class. Guaranteed Pods are the last ones evicted, only when they exceed their own limits.**

Q24 An application needs TCP and UDP on the same port number. How do you configure this in Kubernetes?

> Answer: Kubernetes Services cannot expose the same port for both TCP and UDP in a single Service object. You need two separate Service objects.

Q25 A rolling update caused downtime even though you had it configured. What advanced strategies fix this?

> Answer: Most common cause: the readiness probe passed before the app was actually ready, or the app did not handle SIGTERM gracefully so in-flight requests were dropped.

Section 5: Performance and Advanced Topics

Q26 Your Istio Envoy sidecar is using more CPU and memory than the actual application. How do you optimize?

> Answer: Start by right-sizing using actual observed data from Prometheus. Look at envoy_server_total_connections and container_memory_usage_bytes before changing anything.

Q27 You are building a Kubernetes operator. How do you design the CRD and reconciliation loop?

> Answer: Keep spec and status separate. Spec is what the user declares. Status is what the operator writes. Never mix them.

Q28 Multiple nodes are showing high disk I/O because of container logs. How do you address this?

> Answer: Configure log rotation in kubelet using --container-log-max-size and --container-log-max-files. Add ephemeral-storage limits to pods so a single chatty container cannot fill the node disk.

Q29 Your etcd cluster performance is degrading. What are the root causes, and how do you fix it?

> Answer: etcd is extremely sensitive to disk latency. The number one cause is slow fsync on the WAL. etcd needs under 10ms disk latency consistently.

Q30 How do you enforce that all images in the cluster must come from a trusted internal registry?

> Answer: Use OPA Gatekeeper or Kyverno with a policy that validates the image field against an allowlist of registry prefixes. This runs at admission time so unauthorized images are rejected before scheduling.