Kubernetes Interview Question & Answers

https://digitalksystems.com

We offer a wide range of comprehensive middleware courses on Udemy, including WebLogic Administration, Oracle OHS Administration, JBoss Administration, IBM WebSphere Administration, SOA Suite Administration, SSL Keys and Keystore Administration, OID Administration, Docker Administration, Kubernetes Administration DevOps and more. These courses are meticulously designed to meet real-world professional requirements, ensuring that the content aligns with what is needed to work in a live production environment. Each course is curated with practical concepts, hands-on labs, and detailed explanations to equip learners with the skills they need to manage complex middleware environments confidently. By taking these courses, learners can avoid paying high fees for training programs that often lack practical application or outcomes. Our courses are aimed at making professionals job-ready in a short time, ensuring that they gain the expertise necessary to excel in middleware technologies without unnecessary expenses or wasted time.

DigiTalk

https://digitalksystems.com/

Reach us at digitalk.fmw@gmail.com
DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw
Playlists: https://www.youtube.com/@digitalk.middleware/playlists

DigiTalk Udemy Courses and Coupon Code

Kubernetes Administration

https://www.udemy.com/course/mastering-kubernetes-administration-from-intro-to-advanced/?referralCode=7E0250BEC21F46192834

Coupon Code: C48311E6276C7678BE45

WebSphere Administration

https://www.udemy.com/course/mastering-ibm-websphere-9x-administration/?referralCode=F372B3EA78D7BEA92416

Coupon Code: D7BB41A951A764F212D2

Mastering SSL (TLS), Keys and Certificates Management

https://www.udemy.com/course/mastering-ssl-tls-keys-and-certificates-management/?referralCode=DB9F1429CB01921969F6

Coupon Code: EDB22E66F664B4920777

SOA Suite Administration

https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?referralCode=2BCF3895067AA7BC22E0

Coupon Code: F0DA9B3FCDED9C6770EF

JBoss 8 Administration

https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?referralCode=D095CE8B1EE9D065C822

Coupon Code: C5AF753A6DF55F9CE0F9

OHS Administration

https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?referralCode=165F6A3A8B662CEA1489

Coupon Code: 49830BDBEF3C99624D43

WebLogic Server Administration

https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?referralCode=340FBB94DBFEE1A8CC09

Coupon Code: C3B616B82D57F495E4DA

Mastering Oracle Internet Direct (OID) Administration

https://www.udemy.com/course/mastering-oracle-internet-direct-oid-administration/?referralCode=906E9AC0059A29A3CD0D

Coupon Code: 515B41C98AA8F3075ACE

For any query or concerns please write us on digitalk.fmw@gmail.com

1. What is Kubernetes?
Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It was originally developed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF).

2. What are the key components of Kubernetes architecture?

- **Master Node Components:** API Server, Controller Manager, Scheduler, etcd.
- **Worker Node Components:** Kubelet, Kube Proxy, Container Runtime (Docker, CRI-O, or containerd).

3. What is a Pod in Kubernetes?
A Pod is the smallest deployable unit in Kubernetes that encapsulates one or more containers, shared storage, and network resources.

4. What is the role of a Kubernetes API Server?
The API Server acts as the front-end for Kubernetes, exposing REST APIs that users and other components interact with to manage cluster resources.

5. What is etcd in Kubernetes?
Etcd is a distributed key-value store used by Kubernetes to store all cluster data, including configurations and states.

6. What is the difference between a ReplicaSet and a Deployment?
A **ReplicaSet** ensures a specified number of pod replicas are running at all times, while a **Deployment** provides declarative updates, allowing rollback and rolling updates of applications.

7. What is the difference between StatefulSet and Deployment?
StatefulSet maintains the identity of each pod across restarts, ensuring stable network identifiers and persistent storage, while Deployment focuses on stateless applications.

8. What is the purpose of a Service in Kubernetes?
A Service in Kubernetes provides a stable endpoint to expose a set of pods, ensuring consistent network access despite dynamic pod IPs.

9. What are the types of Services in Kubernetes?

- **ClusterIP:** Internal service accessible within the cluster.
- **NodePort:** Exposes service on a static port on each node.
- **LoadBalancer:** Uses an external load balancer to expose the service.
- **ExternalName:** Maps a service to an external domain name.

10. What is a ConfigMap in Kubernetes?
A ConfigMap allows storing non-confidential configuration data separately from application code.

11. What is a Secret in Kubernetes?
A Secret stores sensitive information like passwords, API keys, and certificates securely in Kubernetes.

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

- **PV:** A cluster-wide storage resource.
- **PVC:** A request for storage by a pod that claims a PV based on requirements.

13. What is the role of Ingress in Kubernetes?
Ingress manages external access to services within a cluster using HTTP and HTTPS routing.

14. How does Kubernetes handle networking?
Kubernetes networking allows all pods to communicate with each other using unique IPs, eliminating the need for NAT. CNI (Container Network Interface) plugins like Calico, Flannel, and WeaveNet manage networking.

15. What is the purpose of DaemonSets in Kubernetes?
DaemonSets ensure that a copy of a pod runs on all (or some) nodes, commonly used for logging, monitoring, and networking services.

16. What are Kubernetes Namespaces?
Namespaces allow isolation of resources within a cluster, enabling multi-tenancy and organization of environments like dev, test, and prod.

17. What is Helm in Kubernetes?
Helm is a package manager for Kubernetes that simplifies application deployment using pre-configured charts.

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

- **HPA:** Adjusts the number of pod replicas based on CPU/memory usage.
- **VPA:** Adjusts resource requests and limits of individual pods based on real-time usage.

19. What is Kubernetes RBAC (Role-Based Access Control)?
RBAC controls access to Kubernetes resources based on user roles and permissions.

20. What is a Kubernetes Operator?
Operators extend Kubernetes functionality by automating the management of complex applications.

21. How does Kubernetes handle rolling updates and rollbacks?
Kubernetes allows rolling updates to deploy changes gradually and rollbacks to revert to a previous stable state if issues arise.

22. What is a Kubernetes Job and CronJob?

- **Job:** Runs a task to completion.
- **CronJob:** Runs a scheduled job at specified intervals.

23. What is the difference between Docker Swarm and Kubernetes?
Kubernetes is more powerful, with built-in service discovery, auto-scaling, and self-healing, whereas Docker Swarm is simpler but lacks advanced orchestration features.

24. What is a Sidecar Container in Kubernetes?
A Sidecar container runs alongside the main application container in the same pod, providing auxiliary functions like logging, monitoring, or proxying.

25. What is the difference between a Multi-Container Pod and a Single-Container Pod?
A Multi-Container Pod runs multiple containers that share storage and networking, allowing inter-container communication, whereas a Single-Container Pod runs just one container per pod.

26. What is a Custom Resource Definition (CRD) in Kubernetes?
CRDs allow users to define and manage custom objects in Kubernetes, extending its API.

27. What is a Mutating and Validating Admission Webhook?

- **Mutating Webhook:** Modifies incoming API requests before they are processed.
- **Validating Webhook:** Validates requests before allowing them.

28. What is the difference between Kube-proxy and CNI?

- **Kube-proxy:** Maintains network rules for communication between pods.
- **CNI:** Manages pod network interfaces and IP allocation.

29. How do you debug a Kubernetes Pod?

- Check pod logs using `kubectl logs`.
- Inspect events using `kubectl describe pod`.
- Execute commands inside a running pod using `kubectl exec -it`.
- View live resource usage using `kubectl top pod`.

30. How do you perform a disaster recovery in Kubernetes?

- Backup etcd regularly.

- Use Helm charts and YAML manifests for easy redeployment.
- Enable multi-zone cluster deployment for high availability.
- Use persistent volumes with cloud storage for data recovery.

31. What is the role of the Kubernetes Scheduler?
The Scheduler is responsible for placing pods on the appropriate nodes based on resource availability, constraints, and policies.

32. What is the difference between "kubectl apply" and "kubectl create"?

- **kubectl apply**: Used to create or update a resource based on the provided configuration file. It is idempotent.
- **kubectl create**: Used to create a resource, but does not update it if it already exists.

33. What is the function of kubelet in Kubernetes?
The kubelet is an agent that runs on each worker node. It ensures the containers described in a pod are running and healthy.

34. How does Kubernetes perform load balancing?
Kubernetes uses Services to provide load balancing. It routes traffic to the healthy pods based on the defined type of Service (ClusterIP, NodePort, LoadBalancer).

35. What is the purpose of a Kubernetes Endpoint?
An Endpoint object represents a set of IP addresses for the pods associated with a service. It is used for accessing the pods that belong to the service.

36. What is the difference between "kubectl exec" and "kubectl run"?

- **kubectl exec**: Used to run a command in a running container.
- **kubectl run**: Used to create a pod with a specified container to execute a command.

37. What is Kubernetes Vertical Pod Autoscaler (VPA)?
The VPA automatically adjusts the CPU and memory requests and limits for pods based on usage patterns.

38. What is the role of the Controller Manager in Kubernetes?
The Controller Manager ensures that the cluster is in the desired state by monitoring the state of resources and adjusting when necessary, e.g., ensuring the correct number of replicas.

39. How can you manage configurations in Kubernetes?
Kubernetes uses ConfigMaps and Secrets to manage configurations. ConfigMaps handle non-sensitive data, while Secrets store sensitive data.

40. What is the difference between kubectl describe and kubectl get?

- **kubectl describe**: Provides detailed information about a resource, including events and status.

- **kubectl get**: Lists basic information about the resource, such as name and status.

41. What is Kubernetes Cluster Autoscaler?
The Cluster Autoscaler automatically adjusts the number of nodes in the cluster based on the resource demand from the pods.

42. How does Kubernetes handle pod health checks?
Kubernetes uses Liveness and Readiness probes to check the health of pods. Liveness probes detect if a pod is alive, while Readiness probes check if a pod is ready to accept traffic.

43. What is the role of a Kubernetes Namespace?
Namespaces provide a way to divide cluster resources between multiple users or teams. They help in organizing resources and managing access control.

44. What is the difference between a pod and a container in Kubernetes?
A pod is the smallest deployable unit in Kubernetes, and it can host one or more containers, while a container is an isolated environment running a specific application.

45. What is a Kubernetes Network Policy?
Network Policies define rules for controlling communication between pods, allowing you to specify which pods can communicate with each other and which cannot.

46. What is a ReplicaSet in Kubernetes?
A ReplicaSet ensures that a specified number of pod replicas are running at all times, providing fault tolerance by creating additional replicas if needed.

47. How does Kubernetes handle persistent storage?
Kubernetes uses Persistent Volumes (PVs) and Persistent Volume Claims (PVCs) to manage storage. Pods can request storage using PVCs, and PVs provide the storage resource.

48. What is a Kubernetes StatefulSet?
A StatefulSet is used for managing stateful applications, ensuring that each pod in the set has a unique identity, stable network identifiers, and persistent storage.

49. How does Kubernetes handle secret management?
Kubernetes uses the Secret resource to store sensitive data securely, which can be accessed by pods without exposing it in plaintext.

50. What is the purpose of the Kubernetes API Server?
The API Server is the central management entity that exposes the Kubernetes API and handles all requests to interact with the cluster.

51. What is Kubernetes Ingress?
Ingress manages external HTTP and HTTPS access to services within a cluster, providing routing rules and SSL termination.

52. What is a Kubernetes CronJob?
A CronJob is used to schedule and run tasks at specific intervals, similar to a cron job in Unix/Linux systems.

53. What are the different types of volumes in Kubernetes?
Kubernetes supports various volume types, including EmptyDir, HostPath, PersistentVolumeClaim, NFS, and cloud-provider volumes like AWS EBS.

54. How can you scale a Kubernetes deployment?
You can scale a deployment using `kubectl scale` or by modifying the `replicas` field in the deployment YAML file and applying the changes with `kubectl apply`.

55. What is a DaemonSet in Kubernetes?
A DaemonSet ensures that a pod is running on every node in the cluster, commonly used for tasks like logging or monitoring.

56. How does Kubernetes perform rolling updates?
Rolling updates allow updating pods incrementally, ensuring that some replicas are always available during the update process. Kubernetes ensures that the update does not take down the service.

57. What is a Kubernetes ServiceAccount?
A ServiceAccount provides an identity for processes that run in a pod to interact with the Kubernetes API.

58. What is Kubernetes Helm?
Helm is a package manager for Kubernetes, allowing you to define, install, and upgrade complex applications using reusable charts.

59. What is a Kubernetes Node?
A Node is a machine, either physical or virtual, that runs the necessary services to manage pods, including the kubelet and container runtime.

60. How can you monitor Kubernetes clusters?
Kubernetes clusters can be monitored using tools like Prometheus, Grafana, or by using cloud-based services like AWS CloudWatch, Google Cloud Monitoring, or Azure Monitor.

61. What is a Kubernetes Persistent Volume (PV)?
A Persistent Volume (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically using StorageClass.

62. What is a Kubernetes Persistent Volume Claim (PVC)?
A PVC is a request for storage by a user or application, specifying size, access modes, and storage class, which is then matched to a suitable PV.

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

- **Job**: Runs a task until completion.
- **CronJob**: Runs tasks on a scheduled basis at specified intervals.

64. What is the role of Kube Proxy?
Kube Proxy is responsible for maintaining network rules and providing network connectivity to Pods, either by forwarding traffic or by using iptables or ipvs.

65. What is Kubernetes Horizontal Pod Autoscaler (HPA)?
HPA automatically adjusts the number of replicas in a deployment based on metrics like CPU utilization or custom metrics.

66. How does Kubernetes handle storage provisioning?
Kubernetes uses StorageClasses to define the type of storage to be used and provisions persistent volumes dynamically based on the storage class.

67. What is the role of the Kubernetes Scheduler?
The Scheduler places pods on suitable nodes based on resource requirements, constraints, and policies.

68. What are the key Kubernetes primitives?
The key Kubernetes primitives include Pods, Services, Deployments, ReplicaSets, ConfigMaps, Secrets, Volumes, Namespaces, and Jobs.

69. What are Kubernetes Taints and Tolerations?
Taints are applied to nodes to prevent pods from being scheduled on them unless the pod has a matching toleration, allowing control over pod placement.

70. What is the role of the API Server in Kubernetes?
The API Server is the central control plane component that exposes the Kubernetes API, enabling communication between different cluster components.

71. What is Kubernetes Control Plane?
The Control Plane is responsible for managing the Kubernetes cluster. It consists of components like the API Server, Scheduler, Controller Manager, and etcd that ensure the desired state of the cluster is maintained.

72. What is etcd in Kubernetes?
etcd is a distributed key-value store used to store all the configuration data and the cluster state in Kubernetes. It holds information such as node status, pod configurations, and service definitions.

73. What is the difference between Deployment and StatefulSet?

- **Deployment**: Manages stateless applications, where pods are interchangeable.
- **StatefulSet**: Manages stateful applications, ensuring stable network identities and persistent storage.

74. What is the use of the "kubectl get pods" command?
The `kubectl get pods` command lists all the pods running in the cluster, providing basic details like the pod name, status, and namespace.

75. What is a Kubernetes Pod?
A Pod is the smallest deployable unit in Kubernetes, which can contain one or more containers that share network and storage resources.

76. What is the purpose of a Kubernetes Service?
A Service is an abstraction that defines a set of pods and a policy to access them. It provides load balancing and ensures that clients can reach the correct pod.

77. What is Kubernetes ConfigMap used for?
A ConfigMap is used to store non-sensitive configuration data in the form of key-value pairs that can be accessed by applications running in pods.

78. What is the purpose of a Kubernetes Secret?
A Secret is used to store sensitive information, such as passwords, OAuth tokens, or SSH keys, that are needed by applications running in pods.

79. How can you upgrade a Kubernetes cluster?
To upgrade a Kubernetes cluster, you can use the `kubeadm upgrade` command or use cloud-based managed services to perform the upgrade process.

80. What is a Kubernetes ClusterIP Service?
A ClusterIP Service exposes the service on an internal IP address within the cluster. It can only be accessed by other pods within the cluster.

81. What is the difference between NodePort and LoadBalancer Service types?

- **NodePort**: Exposes the service on a static port on all nodes in the cluster.
- **LoadBalancer**: Provisions a load balancer to route traffic from external sources to the service.

82. How do you expose a Kubernetes pod to the internet?
You can expose a pod to the internet by using a `LoadBalancer` type Service or creating an `Ingress` resource to manage external access.

83. What is Kubernetes Resource Requests and Limits?

- **Resource Requests**: Specify the minimum amount of CPU or memory that a container will get.
- **Resource Limits**: Specify the maximum amount of CPU or memory a container can use.

84. What is the Kubernetes Horizontal Pod Autoscaler (HPA)?
The Horizontal Pod Autoscaler automatically adjusts the number of pods in a deployment or replica set based on CPU utilization or custom metrics.

85. What is the role of a Kubernetes ReplicaSet?
A ReplicaSet ensures that a specified number of identical pods are running at any given time, providing fault tolerance and scaling.

86. How does Kubernetes handle inter-pod communication?
Kubernetes allows inter-pod communication through its networking model, where each pod gets its own unique IP address and can communicate with other pods using DNS names or IP addresses.

87. What is the Kubernetes Pod Disruption Budget (PDB)?
A Pod Disruption Budget ensures that a certain number or percentage of pods in a deployment are always available during voluntary disruptions (e.g., updates or maintenance).

88. What are Kubernetes Namespaces used for?
Namespaces in Kubernetes provide a way to partition cluster resources between multiple users or teams, helping to organize resources and enforce access control policies.

89. What is the purpose of the Kubernetes Scheduler?
The Scheduler is responsible for assigning pods to nodes in the cluster based on resource availability, constraints, and other factors.

90. What is the role of Kubernetes Ingress Controller?
The Ingress Controller is responsible for implementing the rules defined in the Ingress resource, which manages external HTTP/HTTPS access to services within a Kubernetes cluster.

91. What is the Kubernetes Dashboard?
The Kubernetes Dashboard is a web-based user interface for managing Kubernetes clusters. It allows users to deploy applications, monitor resources, and troubleshoot issues.

92. What is the role of the KubeProxy in Kubernetes?
KubeProxy manages network rules for pod communication within the cluster. It routes traffic between pods and services, implementing virtual IPs for services.

93. What is the difference between a StatefulSet and a DaemonSet?

- **StatefulSet**: Manages stateful applications, ensuring that each pod has a stable identity and persistent storage.
- **DaemonSet**: Ensures that a copy of a pod runs on every node in the cluster, often used for system-level tasks like monitoring.

94. What is the function of the Kubernetes Controller Manager?
The Controller Manager is responsible for ensuring that the current state of the cluster matches the desired state, such as maintaining the correct number of replicas for deployments.

95. How do you restrict access to a Kubernetes cluster?
Access to a Kubernetes cluster can be restricted using Role-Based Access Control (RBAC) policies, ServiceAccounts, and API Server authentication mechanisms.

96. What is the purpose of the Kubernetes Admission Controller?
Admission Controllers are plugins that intercept API requests to validate, mutate, or authorize the resources being created or modified within the cluster.

97. What is the difference between Kubernetes Deployment and ReplicaSet?

- **Deployment**: Manages the rollout, scaling, and rollback of applications.
- **ReplicaSet**: Ensures a specified number of pod replicas are running, and it is often managed by a Deployment.

98. What is a Kubernetes Job?
A Job is used to manage batch processing tasks in Kubernetes. It ensures that a specified number of pods complete their work successfully.

99. How can you manage sensitive data in Kubernetes?
Sensitive data can be managed using Kubernetes Secrets, which store sensitive information in an encrypted form that can be used by pods securely.

100. What is Kubernetes Taint and Toleration?
Taints are applied to nodes to repel pods from being scheduled on them unless the pod has a matching toleration. This allows more fine-grained control over pod placement.

DISCLAIMER AND CONSENT

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with Middleware and DevOps. While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by any other third party organization.

We do not claim or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.