

Kubernetes Interview Questions

Download PDFPrint

Use the filter to quickly find topics like pods, deployments, services, ingress, storage, scheduling, and debugging.

Filter (e.g., pods, deployments, services, ingress, etcd, schedi

Showing 85 of 85

1. What is Kubernetes and what problem does it solve?

Kubernetes is an open-source platform for automating deployment, scaling, and management of containerized applications. It solves the problem of managing the deployment and scaling of containers in a production environment by providing a platform-agnostic way to manage containerized applications.

2. What are the key components of a Kubernetes cluster?

The key components of a Kubernetes cluster are the Master node, Worker nodes, API server, etcd, Controller manager, Scheduler, and Kubelet.

3. What is a Pod in Kubernetes?

A pod in Kubernetes is the smallest and simplest unit in the Kubernetes object model. It represents a single instance of a running process in your cluster.

4. What is a Replication Controller in Kubernetes?

A Replication Controller in Kubernetes is responsible for ensuring that a specified number of pod replicas are running at any given time. It provides self-healing capabilities to the cluster by automatically replacing failed or terminated pods.

5. What is the role of the API server in a Kubernetes cluster?

The API server is the central management component in a Kubernetes cluster. It exposes the Kubernetes API, which allows clients to interact with the cluster. The API server processes REST requests and updates the etcd datastore with the new state of the cluster.

6. What is a Service in Kubernetes?

A Service in Kubernetes is a component that defines a logical set of Pods and a policy by which to access them. Services provide a stable IP address and DNS name, load balancing, and network traffic management to the Pods they represent.

7. What is a Deployment in Kubernetes?

A Deployment in Kubernetes is a higher-level component that manages Replication Controllers and Pods. It provides declarative updates for Pods and Replication Controllers, and it can be used to create, update, and roll back a complete deployment.

8. What is a Namespace in Kubernetes?

A Namespace in Kubernetes is a virtual cluster created within a cluster. It provides a way to divide cluster resources between multiple users, teams, or projects. Each namespace has its own unique name and is isolated from other namespaces.

9. What is a ConfigMap in Kubernetes?

A ConfigMap in Kubernetes is a mechanism to store configuration data as key-value pairs that can be used by Pods and other parts of the cluster. ConfigMaps allow you to decouple configuration data from your application code, making it easier to manage and maintain.

10. What is a Volume in Kubernetes?

A Volume in Kubernetes is a mechanism to store data that is accessible to containers in a Pod. Volumes provide a way to persist data across container restarts and to share data between multiple containers in a Pod. Kubernetes supports several types of volumes, including hostPath, emptyDir, and more.

11. What is a StatefulSet in Kubernetes?

A StatefulSet in Kubernetes is a higher-level component that manages Pods and provides guarantees about the ordering and unique naming of Pods. StatefulSets are used to manage stateful applications, such as databases, where data needs to persist even if the Pods are restarted.

12. What is a DaemonSet in Kubernetes?

A DaemonSet in Kubernetes is a higher-level component that ensures that a specific Pod runs on all or some of the nodes in a cluster. DaemonSets are used to run system-level or node-level services, such as log collectors, network plugins, and more.

13. What is a Job in Kubernetes?

A Job in Kubernetes is a higher-level component that represents a batch operation that runs to completion. Jobs are used to run one-off or infrequent tasks, such as data processing, backup and restore operations, and more.

14. What is a Horizontal Pod Autoscaler in Kubernetes?



A Horizontal Pod Autoscaler in Kubernetes is a higher-level component that automatically increases or decreases the number of replicas of a Deployment, ReplicaSet, or StatefulSet based on the observed CPU utilization or other custom metrics.

15. What is an Ingress in Kubernetes?

An Ingress in Kubernetes is a higher-level component that provides a way to route external traffic to Services in a cluster. Ingresses can be used to expose multiple Services under a single IP address, route traffic based on URL paths, and provide load balancing, SSL/TLS termination, and more.

16. What is a Cluster in Kubernetes?

A Cluster in Kubernetes is a set of physical or virtual machines that run a set of coordinated nodes. Each node runs containerized applications managed by Kubernetes, and the nodes communicate with each other to form the cluster.

17. What is a Node in Kubernetes?

A Node in Kubernetes is a single machine in a cluster. Each node runs an instance of the Kubernetes control plane, as well as one or more Pods that host containerized applications.

18. What is a Label in Kubernetes?

A Label in Kubernetes is a key-value pair that can be attached to objects in the cluster, such as Pods, Services, and more. Labels are used to categorize and organize resources, and they can be used to query, select, and manage groups of resources.

19. What is a Resource in Kubernetes?

A Resource in Kubernetes is a fundamental unit of work in the cluster. Resources represent objects in the cluster, such as Pods, Services, and more. Resources are defined by custom resource definitions (CRDs) or by using built-in resource types.

20. What is a Secret in Kubernetes?

A Secret in Kubernetes is a resource that holds sensitive information, such as passwords, API keys, and more. Secrets are stored in an encrypted form in the cluster, and they can be used by Pods and other components in a secure manner.

21. What is a Namespace in Kubernetes?

A Namespace in Kubernetes is a virtual cluster within a physical cluster. Namespaces provide a way to isolate resources and limit their scope, and they are used to group and organize resources in the cluster.

22. What is a Service in Kubernetes?

A Service in Kubernetes is a resource that provides stable network connectivity to a group of Pods. Services abstract the network communication between Pods and provide load balancing, service discovery, and more.

23. What is a Volume in Kubernetes?

A Volume in Kubernetes is a way to store data that is accessible to one or more Pods. Volumes can be used to persist data across restarts, to share data between Pods, and to provide access to external storage systems, such as NFS, iSCSI, and more.

24. What is a Persistent Volume in Kubernetes?

A Persistent Volume (PV) in Kubernetes is a resource that represents a piece of storage in the cluster. PVs can be dynamically provisioned or statically provisioned, and they can be used to persist data across restarts and to share data between Pods.

25. What is a Persistent Volume Claim in Kubernetes?

A Persistent Volume Claim (PVC) in Kubernetes is a request for storage by a user. PVCs are used to claim a PV, and they can be dynamically provisioned by the cluster or statically bound to a specific PV.

26. What is a Deployment in Kubernetes?

A Deployment in Kubernetes is a resource that manages the creation and update of Pods. Deployments provide a way to roll out new versions of an application, roll back to previous versions, and more.

27. What is a DaemonSet in Kubernetes?

A DaemonSet in Kubernetes is a resource that ensures that a specific Pod runs on all or some of the nodes in a cluster. DaemonSets are used to run system daemons, such as logging agents and monitoring agents, and they are also used to run application components that need to be deployed on every node in the cluster.

28. What is a Job in Kubernetes?

A Job in Kubernetes is a resource that manages the execution of one or more Pods to completion. Jobs are used to run batch workloads, such as data processing, and they are also used to run tasks that need to be completed, such as database migrations.

29. What is a CronJob in Kubernetes?



A CronJob in Kubernetes is a resource that runs a Job on a specified schedule. CronJobs are used to run scheduled workloads, such as backups and maintenance tasks, and they are defined using a cron-style schedule expression.

30. What is a StatefulSet in Kubernetes?

A StatefulSet in Kubernetes is a resource that manages the deployment and scaling of stateful applications. StatefulSets provide a way to manage network identities, persist data across restarts, and more, and they are used to run stateful applications, such as databases and caching systems.

31. What is a ConfigMap in Kubernetes?

A ConfigMap in Kubernetes is a resource that stores configuration data that can be consumed by Pods and other resources in the cluster. ConfigMaps provide a way to manage configuration data separately from the application code, and they can be used to store environment variables, command-line flags, and more.

32. What is a Secret in Kubernetes?

A Secret in Kubernetes is a resource that stores sensitive information, such as passwords, API keys, and more. Secrets provide a secure way to store sensitive data, and they can be used to store database credentials, API keys, and more.

33. What is an Ingress in Kubernetes?

An Ingress in Kubernetes is a resource that provides external access to the services in a cluster. Ingresses provide a way to expose HTTP and HTTPS routes, and they can be used to load balance incoming traffic, terminate SSL, and more.

34. What is a Custom Resource Definition (CRD) in Kubernetes?

A Custom Resource Definition (CRD) in Kubernetes is a way to define custom resources that can be used in a cluster. CRDs provide a way to extend the Kubernetes API and add custom resource types to a cluster, and they can be used to define custom resources that can be managed by Kubernetes controllers.

35. What is a Controller in Kubernetes?

A Controller in Kubernetes is a component that manages the state of the cluster. Controllers are used to ensure that the desired state of the cluster is maintained, and they can be used to manage Deployments, StatefulSets, Jobs, and more.

36. What is a Service Account in Kubernetes?

A Service Account in Kubernetes is an account that is associated with a particular set of Pods and provides a way to control access to the API server. Service Accounts can be used to manage authentication and authorization for Pods, and they can be associated with RBAC roles to control access to resources in the cluster.

37. What is a DaemonSet in Kubernetes?

A DaemonSet in Kubernetes is a resource that ensures that a single instance of a Pod is running on each node in the cluster. DaemonSets are used to run system-level components, such as logging agents, and they can be used to ensure that critical components are always running and available.

38. What is a Namespace in Kubernetes?

A Namespace in Kubernetes is a virtual cluster within a physical cluster, and it provides a way to isolate resources and control access to resources. Namespaces can be used to manage resources for different environments, such as development, staging, and production, and they can also be used to control resource quotas and limit resource usage.

39. What is a StatefulSet in Kubernetes?

A StatefulSet in Kubernetes is a resource that manages stateful applications and provides a way to maintain the unique network identity of Pods. StatefulSets are used to manage stateful applications, such as databases, and they provide a way to ensure that Pods maintain the same network identity across restarts, upgrades, and rescheduling.

40. What is an Operator in Kubernetes?

An Operator in Kubernetes is a custom controller that extends the Kubernetes API and provides a way to automate the management of complex applications. Operators are used to manage complex applications, such as databases, and they provide a way to automate the management of application lifecycle, configuration, scaling, and more.

41. What is Ingress in Kubernetes?

Ingress in Kubernetes is a resource that provides a way to manage external access to Services in a cluster. Ingress can be used to route incoming traffic to multiple Services based on rules, such as URL paths or host names, and it can be used to manage SSL termination, load balancing, and other networking tasks.

42. What is a ConfigMap in Kubernetes?

A ConfigMap in Kubernetes is a resource that provides a way to manage configuration data for Pods and Services. ConfigMaps can be used to store configuration files, environment variables, and other configuration data, and they can be mounted as volumes or used as

 environment variables in Pods.

43. What is a Secret in Kubernetes?

A Secret in Kubernetes is a resource that provides a way to store and manage sensitive information, such as passwords and certificates, in a cluster. Secrets can be mounted as volumes or used as environment variables in Pods, and they are encrypted at rest and in transit.

44. What is a Job in Kubernetes?

A Job in Kubernetes is a resource that provides a way to run a single instance of a task to completion in a cluster. Jobs can be used to run batch processing tasks, such as data import and export, and they provide a way to ensure that a task runs to completion, regardless of node failures or restarts.

45. What is a Persistent Volume (PV) and Persistent Volume Claim (PVC) in Kubernetes?

A Persistent Volume (PV) in Kubernetes is a resource that provides a way to manage and persist data in a cluster, and a Persistent Volume Claim (PVC) is a resource that requests a specific amount of storage from a PV. PVs and PVCs are used to manage persistent storage for Pods, and they provide a way to dynamically provision storage, manage storage lifecycle, and more.

46. What is a Service in Kubernetes?

A Service in Kubernetes is a resource that provides a stable network identity and network service to a set of Pods. Services can be used to load balance incoming network traffic, and they can be exposed to the external network or to other Pods in the cluster.

47. What is a ReplicationController in Kubernetes?

A ReplicationController in Kubernetes is a resource that provides a way to manage the lifecycle of multiple identical Pods. ReplicationControllers ensure that the specified number of Pods are running at all times, and they can be used to scale up or down the number of running Pods based on demand.

48. What is a StatefulSet in Kubernetes?

A StatefulSet in Kubernetes is a resource that provides a way to manage the deployment and scaling of stateful applications. StatefulSets provide a stable network identity and persistent storage to each Pod, and they can be used to deploy databases, message queues, and other stateful applications.

49. What is a DaemonSet in Kubernetes?

A DaemonSet in Kubernetes is a resource that provides a way to manage the deployment and scaling of system-level daemons. DaemonSets ensure that a specified Pod runs on every node in the cluster, or on a specific set of nodes, and they can be used to deploy logging, monitoring, and other system-level daemons.

50. What is a Deployment in Kubernetes?

A Deployment in Kubernetes is a resource that provides a declarative way to manage the lifecycle of Pods. Deployments provide a way to specify the desired state of Pods, and they ensure that the specified number of Pods are running and available at all times. Deployments also provide a way to roll out updates to Pods, rollbacks, and more.

51. What is the command to list all the pods in a namespace?

The command to list all the pods in a namespace is ``kubectl get pods -n <namespace-name>``.

52. What is the command to describe a specific pod in detail?

The command to describe a specific pod in detail is ``kubectl describe pod <pod-name>``.

53. What is the command to apply a yaml file to create resources in Kubernetes cluster?

The command to apply a yaml file to create resources in Kubernetes cluster is ``kubectl apply -f <file-name.yaml>``.

54. What is the command to scale the replicas of a deployment?

The command to scale the replicas of a deployment is ``kubectl scale deployment <deployment-name> --replicas=<new-replicas-count>``.

55. What is the command to delete a resource by name?

The command to delete a resource by name is ``kubectl delete <resource-type> <resource-name>``.

56. What is the command to check the logs of a pod?

The command to check the logs of a pod is ``kubectl logs <pod-name>``.

57. What is the command to check the events of a namespace?

The command to check the events of a namespace is ``kubectl get events -n <namespace-name>``.

58. What is the command to check the status of a deployment?

✂ The command to check the status of a deployment is ``kubectl rollout status deployment <deployment-name>``.

59. What is the command to check the details of a resource type?

The command to check the details of a resource type is ``kubectl explain <resource-type>``.

60. What is the command to run a command in a running container of a pod?

The command to run a command in a running container of a pod is ``kubectl exec -it <pod-name> -- <command>``.

61. What is the command to check the details of a specific node in the cluster?

The command to check the details of a specific node in the cluster is ``kubectl describe node <node-name>``.

62. What is the command to update a resource in the cluster?

The command to update a resource in the cluster is ``kubectl apply -f <file-name.yaml>``.

63. What is the command to check the resources that are running in a specific namespace?

The command to check the resources that are running in a specific namespace is ``kubectl get all -n <namespace-name>``.

64. What is the command to check the history of a deployment?

The command to check the history of a deployment is ``kubectl rollout history deployment <deployment-name>``.

65. What is the command to port-forward a service to the local machine?

The command to port-forward a service to the local machine is ``kubectl port-forward svc/<service-name> <local-port>:<remote-port>``.

66. What is the command to check the resource utilization of nodes in the cluster?

The command to check the resource utilization of nodes in the cluster is ``kubectl top node``.

67. What is the command to check the resource utilization of pods in the cluster?

The command to check the resource utilization of pods in the cluster is ``kubectl top pod``.

68. What is the command to label a node in the cluster?

The command to label a node in the cluster is ``kubectl label node <node-name> <label-key>=<label-value>``.

69. What is the command to check the labels of a node in the cluster?

The command to check the labels of a node in the cluster is ``kubectl get nodes --show-labels``.

70. What is the command to replace a resource in the cluster?

The command to replace a resource in the cluster is ``kubectl replace -f <file-name.yaml>``.

71. What is the command to check the events of a specific resource in the cluster?

The command to check the events of a specific resource in the cluster is ``kubectl describe <resource-type> <resource-name>``.

72. What is the command to check the logs of a specific pod in the cluster?

The command to check the logs of a specific pod in the cluster is ``kubectl logs <pod-name>``.

73. What is the command to check the logs of a specific container in a pod?

The command to check the logs of a specific container in a pod is ``kubectl logs <pod-name> -c <container-name>``.

74. What is the command to check the environment variables of a specific pod in the cluster?

The command to check the environment variables of a specific pod in the cluster is ``kubectl exec <pod-name> env``.

75. What is the command to run a command in a specific container in a pod?

The command to run a command in a specific container in a pod is ``kubectl exec <pod-name> -c <container-name> -- <command>``.

76. What is the command to check the history of a resource in the cluster?

The command to check the history of a resource in the cluster is ``kubectl rollout history <resource-type> <resource-name>``.

77. What is the command to undo the last rollout of a resource in the cluster?

The command to undo the last rollout of a resource in the cluster is ``kubectl rollout undo <resource-type> <resource-name>``.

78. What is the command to check the available storage classes in the cluster?

The command to check the available storage classes in the cluster is ``kubectl get storageclass``.

79. What is the command to check the available cluster-level resources in the cluster?

✕ The command to check the available cluster-level resources in the cluster is ``kubectl api-resources --namespaced=false``.

80. What is the command to check the available namespaced resources in the cluster?

The command to check the available namespaced resources in the cluster is ``kubectl api-resources --namespaced=true``.

81. How do you troubleshoot a failed pod in the cluster?

Troubleshooting a failed pod in the cluster can be done by checking the logs of the pod with the ``kubectl logs`` command, checking the events of the pod with the ``kubectl describe pod`` command, and checking the resource utilization of the pod and its container(s) with the ``kubectl top`` command.

82. What is the command to update the image of a deployment in the cluster?

The command to update the image of a deployment in the cluster is ``kubectl set image deployment <deployment-name> <container-name>=<new-image-name>``.

83. What is the command to drain a node in the cluster?

The command to drain a node in the cluster is ``kubectl drain <node-name> [--ignore-daemonsets] [--grace-period=<seconds>]``.

84. What is the command to check the state of the resources in a particular namespace in the cluster?

The command to check the state of the resources in a particular namespace in the cluster is ``kubectl get all --namespace=<namespace-name>``.

85. What is the command to label a node in the cluster?

The command to label a node in the cluster is ``kubectl label node <node-name> <label-key>=<label-value>``.

Tip: When answering K8s questions, mention desired vs. current state, controllers & reconciliation loops, and how you observe/trace with events & logs.

SEARCH

Search topics...

Go

INTERVIEW TOPICS

Java	Q&A
Java 8	Q&A
Spring	Q&A
Hibernate	Q&A
React JS	Q&A
SQL	Q&A
MongoDB	Q&A
JavaScript	Q&A
Kafka	Q&A
Design Patterns	Q&A
Microservices	Q&A
REST API	Q&A
Behavioral	Q&A

QUICK ACTIONS



[Start System Design](#)[Practice Coding](#)[Earn a Certificate](#)

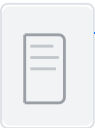
RECOMMENDED READING



[Learn Java 17 Programming](#)



[Learn JavaScript Quickly](#)



[SQL QuickStart Guide](#)



[System Design Interview](#)

[See all](#) ➔

SPONSORED

Ads help keep content free.

NEED HELP?

Have a topic request or correction? We'd love to hear from you.

Contact Us

THANK YOU!

Your comments and complements are always the inspiration that keeps us motivated.

ABOUT

[PROJECTS](#)

[ABOUT US](#)

[BLOG](#)

[AWARDS](#)

ADDRESS



Sterling, Virginia, 20166, US



info@techventurellc.com



641 451 3463



641 451 3463

FOLLOW US



© 2022 Copyright: techventurellc.com