



100 must-know Kubernetes interview questions and answers in 2025

You've come to the perfect place if you want to work as a successful Kubernetes developer for a top Silicon Valley organization or build a team of competent Kubernetes engineers. For your Kubernetes interview, we've carefully created a list of Kubernetes interview questions to give you an idea of the types of Kubernetes interview questions you can ask or be asked.

Apply as Kubernetes developer

I'm hiring developers

I'm looking for jobs

Share this



Kubernetes, also known as K8s, is a container orchestration platform for managing containerized workloads and services. It allocates application workloads across the Kubernetes cluster and

an increased [demand for Kubernetes developers](#). This demand has also created opportunities for [Kubernetes developers](#) to work with leading tech firms in the US.

Here, we present a carefully curated list of trending Kubernetes interview questions and answers that will assist hiring managers in evaluating the right candidate and help developers excel in their following interviews.

Table of contents

[Basic Kubernetes interview questions and answers \(30\)](#)

[Intermediate Kubernetes interview questions and answers \(39\)](#)

[Advanced Kubernetes interview questions and answers \(31\)](#)

BASIC KUBERNETES INTERVIEW QUESTIONS AND ANSWERS

1. Are you familiar with K8s?

[Hide Answer](#)

It is another name for Kubernetes, an open-source platform for managing containerized applications. Kubernetes automates the scaling, deployment, and management of containerized applications, allowing them to run consistently across different computing environments.

2. What is Kubernetes and what does it do?

[Hide Answer](#)

Kubernetes is an open-source container management tool that automates the deployment, scaling, and descaling of containers. It was developed by Google based on their experience of running containerized workloads for over 15 years. Kubernetes has

3. How are Kubernetes and Docker linked?

[Hide Answer](#)

Docker builds containers, which then communicate with each other via Kubernetes. Kubernetes supports multiple container runtimes, including Docker, CRI-O, and others. In simple terms, Kubernetes is analogous to an operating system, and Docker containers are comparable to applications installed on that operating system.

Docker is a containerization platform that allows developers to package and distribute their applications as self-contained units, known as containers. Kubernetes, on the other hand, is a container orchestration platform that automates the deployment, scaling, and management of containerized applications.

The two technologies, Kubernetes and Docker, work together to enable the deployment and management of applications in a distributed environment.

4. Can you explain what container orchestration is?

[Hide Answer](#)

Container orchestration is a process that involves managing and coordinating the deployment, scaling, and operation of multiple containers that run on a single server or across multiple servers. It automates container provisioning, networking, resource allocation, load balancing, availability, and lifecycle management tasks. Container orchestration tools, such as Kubernetes, Docker Swarm, and Apache Mesos, help ensure that containers work together seamlessly and efficiently to deliver the desired applications or services.

5. Why do we need container orchestration?

[Hide Answer](#)

Container orchestration is needed to manage and automate containerized applications deployment, scaling, and management. It helps to reduce operational overhead, increase efficiency and scalability, and ensure infrastructure availability, thus helping to improve application performance.

6. List the features of Kubernetes.

[Hide Answer](#)

Some of the features of Kubernetes are:

Automated scheduling: Kubernetes automatically schedules containers to run on the available resources in the cluster.

Self-healing capabilities: Kubernetes automatically replaces failed containers and reschedules them on healthy nodes in the cluster.

Automated rollouts and rollback: Kubernetes is responsible for automating the deployment process, enabling users to roll out new versions of their applications easily and roll them back in case of any issues.

Horizontal scaling and load balancing: Kubernetes automatically scales and distributes traffic across multiple instances of a containerized application.

Configuration management: Kubernetes enables users to define and manage the application configuration settings separately from the application code.

Service discovery & networking: Kubernetes provides a built-in service discovery mechanism to locate and communicate with other services in the cluster.

Security & compliance: Kubernetes provides several security features, including access control to ensure the security of the cluster and compliance with organizational policies.

7. How does Kubernetes help in containerized deployment?

[Hide Answer](#)

configuration that instructs Kubernetes on creating and updating instances of your application. Kubernetes manage these instances, which can automatically recover from failures and scale up or down based on demand.

8. What are clusters in Kubernetes?

[Hide Answer](#)

Clusters in Kubernetes refer to a group of interconnected physical or virtual machines called nodes that work together to efficiently and automatically run containerized applications in a distributed and fault-tolerant manner. Kubernetes clusters allow engineers to orchestrate and monitor containers across multiple physical, virtual, and cloud servers. Kubernetes clusters are designed to be highly available, resilient, and scalable. By using a cluster, engineers can take advantage of the automatic scaling and self-healing capabilities of Kubernetes for optimized application performance.

9. Explain Google Container Engine (Google Kubernetes Engine).

[Hide Answer](#)

It is a Google-managed implementation of the Kubernetes open-source management platform for clusters and Docker containers. It provides a managed environment for deploying, scaling, and managing your containerized applications using Google infrastructure. It is designed to simplify containerized applications' deployment, management, and scaling in a production environment.

10. What is Heapster?

[Hide Answer](#)

A Heapster is a cluster-wide aggregator of data that runs on each node. It is a Kubernetes project that provides a robust monitoring solution for Kubernetes clusters.

more efficient and scalable way to collect and expose resource utilization data from Kubernetes nodes and pods.

11. What do you know about Minikube?

[Hide Answer](#)

Minikube is a lightweight implementation of Kubernetes, which creates a VM on your local machine. It is a tool that sets the Kubernetes environment on your laptop or PC, and it addresses a different type of use case than most other distributions, such as Rancher, EKS, and OpenShift. It creates a lightweight, self-contained environment with all the necessary components for running Kubernetes, such as the API server, etcd, and kubelet. This allows developers to experiment with Kubernetes without the need for a full-scale production environment.

12. What do you know about Kubectl?

[Hide Answer](#)

A Kubectl is a command-line tool or platform through which you can pass commands to a cluster. Kubectl is the Kubernetes-specific command line tool that lets you communicate and control Kubernetes clusters. With Kubectl, you can deploy applications, inspect and manage cluster resources, view logs, and debug your applications running on Kubernetes. Kubectl can also be used to manage remote and cloud clusters such as GKE.

13. Can you elaborate on the above question?

[Hide Answer](#)

Here's a brief overview of what Kubectl can do:

- Create, read, update, and delete Kubernetes resources (pods, services, deployments, etc.)
 - Interact with the Kubernetes API server to manage cluster resources
 - Monitor the status of resources and diagnose issues
 - Manage Kubernetes configurations and secrets
 - View logs and metrics for applications running on the cluster
-

14. What is a node in Kubernetes?

[Hide Answer](#)

A node is the primary worker machine in the Kubernetes cluster, also known as the minion. It may be a physical or a virtual machine depending on the cluster. It has several components, including a kubelet, container runtime, and Kubernetes components to communicate with the control plane. Nodes can run one or more containers and can be added or removed from the cluster dynamically.

15. List the main components of Kubernetes architecture.

[Hide Answer](#)

Two main components of Kubernetes architecture are the Master node and the Worker node.

Master node: The master node is the control plane making global decisions inside the cluster. The master node comprises the control plane components responsible for managing and coordinating the cluster. These components are the API server, scheduler, cloud controller manager, and controller manager.

16. Can you tell me about kube-proxy?

[Hide Answer](#)

Kube-proxy can run on every node and perform TCP/UDP packet forwarding across the backend network service. Kube-proxy is an important component of Kubernetes networking that helps ensure reliable and efficient communication between pods and services within the cluster. By routing traffic to the correct destination, kube-proxy helps ensure reliable and efficient communication within the cluster.

17. What is the master node in Kubernetes?

[Hide Answer](#)

It controls and manages the worker nodes. It makes up the control plane of a cluster and is responsible for scheduling tasks and monitoring the state of the cluster. The master node has several components, such as Kube-APIServer, Kube-Controller-manager, Etcd, and Kube-Scheduler, to help manage worker nodes.

18. Tell me more about the kube-scheduler.

[Hide Answer](#)

Kube-scheduler distributes and manages the workload on the worker nodes. It uses various policies to select the most suitable node for a pod based on factors such as resource requirements, node capacity, and pod affinity/anti-affinity. The kube-scheduler component is also responsible for binding the selected node to the pod and updating the Kubernetes API server with the updated information.

[Hide Answer](#)

The node in Kubernetes that keeps track of resource utilization is the kubelet. It runs on each worker node and is responsible for managing the state of the node, including starting and stopping pods, as well as monitoring their resource usage. The Metrics Server collects the relevant resource usage stats from kubelet and generates aggregated metrics via metrics API.

20. What is Kubernetes controller manager?

[Hide Answer](#)

The Kubernetes controller manager embeds controllers and is responsible for creating a namespace. Here controllers are control loops that watch the state of your cluster, then make or request changes where needed. Every controller tries to move the current cluster state to the desired state.

21. List the different types of controllers in Kubernetes.

[Hide Answer](#)

Here are some of the types of controllers:

- Node controller
 - Replication controller
 - Service account and token controller
 - Endpoints controller
 - Namespace controller
-

[Hide Answer](#)

Pronounced as "ett-see-dee," it is written in Go programming language and used to coordinate distributed work and key-value pairs. It is an open-source distributed key-value store that holds and manages the critical information distributed systems need to keep running. It is built on the Raft consensus algorithm, which ensures datastore consistency across all the nodes.

23. List the different types of services in Kubernetes.

[Hide Answer](#)

Here are some of the services in Kubernetes:

Cluster IP: This is the default service type in Kubernetes, and it exposes the service on a cluster-internal IP. This means that only the services inside the cluster can access it.

Node Port: This type of service exposes the service on a static port on each node in the cluster. This makes the service accessible from outside the cluster.

Load balancer: This type of service provisions an external load balancer in the cloud infrastructure and directs traffic to the Kubernetes service. This allows you to expose your service to the internet.

External name: This type of service maps the service to an external DNS name. This allows you to reference external services by name from within your cluster.

24. Tell me about load balancers in Kubernetes.

[Hide Answer](#)

In Kubernetes, a load balancer is a component that distributes incoming network traffic across multiple instances of an application running in a cluster. The load balancer sends connections to one server in the pool based on an algorithm to determine the next

25. What is Ingress network?

[Hide Answer](#)

An Ingress network is a set of protocols that acts as an entry point for external traffic into the Kubernetes cluster and manages access to services within the cluster. An Ingress network is traffic whose source lies in the public internet or an external network and is sent to the destined node in the private network. It is used to manage user access for the services within the Kubernetes cluster.

26. What is the role of the Ingress network?

[Hide Answer](#)

The Ingress network manages external access to services in the cluster. Specifically, it enables the load balancing of traffic, termination of SSL/TLS, and virtual hosting for HTTP and HTTPS routes exposed from outside the cluster to services within it. The Ingress network helps streamline communication between the applications and external clients by providing a single entry point for incoming traffic.

27. Explain the cloud controller manager.

[Hide Answer](#)

The cloud controller manager lets you link the cluster to the cloud provider's API. Cloud-controller manager allows cloud vendors to evolve independently from the core Kubernetes code by abstracting the provider-specific code. It abstracts provider-specific code and functionality, which allows cloud vendors to develop and maintain their code independently from the core Kubernetes code.

of the benefits of both Kubernetes and the cloud provider while minimizing the potential for compatibility issues.

28. What are the different types of cloud controller managers?

[Hide Answer](#)

Here are some of the types of cloud controller managers:

Node controller: Responsible for managing the lifecycle of nodes in the cluster, such as creating, updating, and deleting nodes

Route controller: Manages ingress and egress traffic for services running in the Kubernetes cluster

Volume controller: Responsible for managing the lifecycle of volumes in the cluster, such as creating, attaching, and detaching volumes.

Service controller: Manages the lifecycle of Kubernetes services, such as creating, updating, and deleting services.

29. Do you know what container resource monitoring is?

[Hide Answer](#)

Container resource monitoring is the process of constantly collecting metrics. Also, it tracks the health of containerized applications to improve their health and performance and ensure they are operating smoothly. Container resource monitoring has become popular because it provides robust capabilities to track potential failures.

30. What is the init container?

[Hide Answer](#)

images. Examples of tasks that an init container might perform include downloading configuration files, setting up a network connection, or initializing a database schema.

Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

Apply Now

INTERMEDIATE KUBERNETES INTERVIEW QUESTIONS AND ANSWERS

1. List some container resource monitoring tools.

[Hide Answer](#)

Here are some of the resource monitoring tools:

- Grafana
- Kibana
- CAdvisor
- Prometheus
- SolarWinds
- ElasticSearch
- Sysdig

2. What is Grafana?

[Hide Answer](#)

written in TypeScript and Go, respectively.

3. What is CAdvisor?

[Hide Answer](#)

CAdvisor stands for container advisor, which discovers active containers and monitors them. The tool runs as a daemon process in the background, collecting, aggregating, and processing useful DevOps information. With CAdvisor you can track historical resource usage with histograms and analyze the memory footprint of the code running on servers. It primarily helps understand resource consumption.

4. What is Prometheus?

[Hide Answer](#)

Prometheus is an open-source monitoring and alerting system of the Cloud Native Computing Foundation. It was originally developed by SoundCloud and is designed to help users monitor their systems and applications by collecting and processing real-time metrics data. Prometheus has a powerful query language and a flexible alerting system that allows users to define custom rules and receive alerts when certain conditions are met.

Additionally, Prometheus offers various options for data visualization, including built-in graph and dashboarding capabilities. Prometheus supports static and dynamic service discovery mechanisms to discover and monitor targets automatically.

5. Differentiate between a replica set and a replication controller.

[Hide Answer](#)

pods with a single command.

The replica set is the same as the replication controller except that they have more options for the selectors. They use set-based selectors to manage the pods. Here the rolling-update command won't work.

6. Which selectors does the replica set use?

[Hide Answer](#)

A replica set in Kubernetes uses label selectors to identify which pods it should manage. The selectors specify a set of key-value pairs that the replica set uses to match against the labels applied to the pods. Set-based selectors allow filtering keys according to a set of values. There are three kinds of operators: in, not in, and exists. The replica set will look for pods whose labels match the selectors.

7. Which selectors do replication controllers use?

[Hide Answer](#)

Replication controllers use label selectors to identify the set of pods that they manage. Specifically, they use equality-based selectors, which allow filtering by label key and values. These selectors look for pods with labels that match a specific key-value pair. To use an equality-based selector, you can use the "-l" or "--selector" option.

8. What do equality-based selectors do?

[Hide Answer](#)

They allow filtering by label keys and values. Thus they will only look for pods with the exact same phrase as the label. When a pod or other resource is created, it can be

9. How do you monitor applications in Kubernetes?

[Hide Answer](#)

Application monitoring with Kubernetes is not reliant on a single monitoring provider. You can utilize resource metrics or entire metrics pipelines to collect monitoring information on new clusters.

The resource metrics pipeline only delivers a subset of metrics related to cluster components such as the Horizontal Pod Autoscaler controller and the kubectl top function. The lightweight, short-term, in-memory metrics server collects these measurements, which are then provided via the metrics.k8s.io API.

A complete metrics pipeline gives you access to more detailed metrics. Using technologies such as the Horizontal Pod Autoscaler, Kubernetes may respond to these indicators by dynamically scaling or adjusting the cluster based on its present condition.

10. What do you know about Headless service?

[Hide Answer](#)

The Headless Service creates a service grouping that does not allocate an IP address or forward traffic. So here, the spec.clusterIP is explicitly set to "None," and the spec.type is set to "ClusterIP."

This service is used for stateful apps, such as databases, where it is essential to maintain a consistent network identity for each instance. With Headless service, you don't need the proxy to reach the pods.

11. List some security measures that you can take while using Kubernetes.

[Hide Answer](#)

Here are some of the security measures we can take:

- Define source quota
 - Provide limited access to nodes of Kubernetes
-

12. What do you know about federated clusters?

[Hide Answer](#)

Federated clusters help manage multiple Kubernetes clusters, which means you can manage multiple Kubernetes clusters in one place. It provides a centralized view of all the clusters, making it easier to manage and scale applications across different clusters.

Using a federated cluster, you can deploy and manage applications consistently across all your clusters, enabling better workload distribution, resource management, and redundancy.

13. Explain orchestration when it comes to software and DevOps.

[Hide Answer](#)

The automated configuration, coordination, and management of computer systems, services, and applications is called orchestration in DevOps. It leverages DevOps tools that allow for version control, rapid updates and releases, and other best practices for software engineering.

Orchestration in software involves coordinating and managing multiple computer systems, applications, and/or services, stringing together various tasks to execute a larger workflow or process. These processes can consist of numerous tasks that are automated and can involve multiple systems.

14. How do you perform maintenance on the K8 node?

available, the Kubernetes administrator must execute the maintenance work of applying the security patch to the running container to prevent it from becoming vulnerable, which is an inevitable component of the administration. To safely drain the K8s node, use the following two commands:

- `kubectl cordon`
- `kubectl drain --ignore-daemon set`

15. Tell the difference between Docker Swarm and Kubernetes.

[Hide Answer](#)

Here are some key differences between Docker Swarm and Kubernetes:

Architecture: Docker Swarm is a clustering and scheduling tool for Docker containers, while Kubernetes is a container orchestration platform that can manage containers from different container runtimes such as Docker, containerd, and CRI-O.

Installation: Docker Swarm has a simple installation process, and it is relatively easy to set up and manage a Swarm cluster. Kubernetes, on the other hand, has a more complex installation process and requires a lot of configuration and setup before the cluster is up and running.

Scalability: Docker Swarm and Kubernetes can scale horizontally by adding or removing nodes, but Kubernetes has more advanced scaling features than Swarm.

Load balancing: Kubernetes has an advanced load-balancing feature that can distribute traffic across containers based on various factors such as resource utilization, network latency, and more. Swarm, on the other hand, has a simpler load-balancing mechanism that relies on round-robin scheduling.

16. List some features of Kubernetes.

[Hide Answer](#)

Here are some of the features of Kubernetes:

- Scales resources vertically and horizontally
 - Automated deployment and rollout of containerized applications
 - Self-healing capabilities to restart or replace failed containers automatically
 - Load balancing and service discovery for distributing traffic across containers
-

17. How can you use Kubernetes for workload distribution optimization?

[Hide Answer](#)

As Kubernetes helps efficiently optimize resources, it helps with resource distribution. The orchestration tool also ensures that specific resources are used with the application to enhance overall operational efficiency.

Load balancing: Kubernetes provides a built-in load balancer to distribute incoming network traffic to the pods running on the cluster. This helps to evenly distribute the workload across the nodes in the cluster, which can help to optimize resource utilization.

Auto scaling: Kubernetes can automatically scale the number of replicas of a pod based on the resource utilization of the existing pods. This helps to ensure that there are always enough resources available to handle the workload and can help to optimize resource utilization.

Resource management: Kubernetes allows you to specify the resource requirements and limits for each pod, which helps to ensure that the right amount of resources is allocated to each pod.

18. How can an organization improve its technical operations and keep costs low through Kubernetes?

[Hide Answer](#)

An organization can use the DevOps framework to achieve low cost and improve technical operations. Here's how an organization can use Kubernetes to improve

containerized applications, which reduces manual intervention and saves time.

Organizations can use Kubernetes to automate the deployment process and improve technical operations.

Efficient resource utilization: Kubernetes can optimize the use of computing resources by allowing containers to share resources and run on the same nodes. This feature helps to reduce the cost of infrastructure by utilizing resources efficiently.

Horizontal scaling: Kubernetes allows an organization to scale its applications horizontally by adding more containers to the cluster. This feature allows organizations to handle increased traffic without over provisioning resources, which can lead to cost savings.

Monitoring and logging: Kubernetes provides monitoring and logging capabilities, which allow an organization to identify and resolve issues quickly. By using these features, organizations can improve technical operations and reduce the cost of troubleshooting.

19. What is contained in node status?

[Hide Answer](#)

A node status contains address, condition, capacity, and info. It identifies the status of the nodes within the system.

Address: It depends on your cloud provider or bare metal configuration whether you have to use it or not.

Condition: It discusses the status or conditions of all running nodes.

Capacity: This block tells about the total resources processed by the node.

Info: It discusses general information about the node, such as the Kubernetes version, kernel version, container runtime details, and the current operating system which the node uses.

20. Tell the process name that runs on the Kubernetes master node.

process is the access point for all the REST commands that control the cluster. It is the only Kubernetes control panel component with a user-accessible API and the sole master component you will interact with. It acts as a gateway to the cluster and supports life cycle orchestration.

21. Explain the pod in Kubernetes.

[Hide Answer](#)

Pods are fundamental structures in Kubernetes that represent a single instance of a running process in a cluster. A pod encapsulates one or more containers, which share the same network namespace and can communicate with each other through the localhost interface.

Pods provide a logical unit of deployment in Kubernetes, allowing for easy scaling and management of containerized applications. They can be used to deploy stateless services and stateful applications that require shared storage or coordination between containers.

Each pod in Kubernetes has a unique IP address, which allows it to communicate with other pods and services in the cluster.

22. What does a kube-scheduler do?

[Hide Answer](#)

Kube-scheduler allows nodes to schedule freshly created pods. It determines which nodes are accurate placements for every pod in the scheduling queue according to available resources and constraints. This scheduler selects an optimal node for each new Pod or unscheduled Pod for them to run on. But still, every container in pods has different requirements for resources and needs.

[Hide Answer](#)

The clusters of containers in Kubernetes are a set of nodes. They create some routes so the containers running on the nodes can communicate. They allow containers to run across multiple machines and environments, such as physical, virtual, and cloud-based. Kubernetes clusters coordinate processes such as scheduling and scaling applications, maintaining a cluster's state, and implementing updates.

24. Explain DaemonSets.

[Hide Answer](#)

The DaemonSets are pod sets that run on a host only one time on a host. And so, they are used attributes of host layers such as -network or network monitoring. DaemonSet enables you to run the Kubernetes pod on the cluster nodes that fulfill certain criteria or requirements. It ensures that some or all nodes run a copy of a pod. DaemonSets are used to deploy background services, run cluster storage, log collection, and deploy a node monitoring daemon on every node

25. List some initial namespaces from which Kubernetes starts.

[Hide Answer](#)

Some initial namespaces are as follows:

Default: This namespace is included by Kubernetes so that users can start using the new cluster without needing first to create a namespace.

Kube-system: It is the namespace for objects and is created by the Kubernetes system.

Kube-public: This namespace is readable to authenticated and non-authenticated clients alike. It is reserved mainly for cluster usage.

[Hide Answer](#)

It is a default service in Kubernetes that apps inside your cluster can access. It provides a load-balanced IP address. One or more pods matching the label selector can forward traffic to the IP address.

With ClusterIP addresses, it is easier to manage and connect services in a Kubernetes cluster easier. Because each service has a unique IP address, traffic can be easily routed to the correct service without knowing the underlying pods' specific IP address.

27. What do you know about NodePort?

[Hide Answer](#)

NodePort is a Kubernetes networking concept that allows external traffic to access a service in a cluster directly. It is done by opening a specific port on all nodes and sending traffic from this port to the service. When a service is created with a NodePort type, Kubernetes assigns a static port number to the service. The assigned port is then opened on every node in the cluster. This allows any traffic that is sent to this port on any node to be forwarded to the corresponding service.

28. What do you know about Ingress network? And how does it work?

[Hide Answer](#)

It is an object that lets users access services from your Kubernetes, even outside the Kubernetes cluster. You can also configure the access by defining which connections reach which services.

As it is an API object, it provides some routing rules. You can easily manage user access for the services in the Kubernetes cluster through these rules. It is done through HTTPS/HTTP. Hence users can easily set rules for traffic routing without needing load balancers.

[Hide Answer](#)

Within a pod, containers share networking space and can reach one another through localhost. This means they can communicate using standard inter-process communication mechanisms like TCP/IP or Unix domain sockets as if running on the same machine. In addition to communicating through localhost, pod containers can communicate with each other using Kubernetes services or environment variables.

30. Differentiate between ConfigMaps and Secret?

[Hide Answer](#)

Where Secret stores critical data in an encrypted format, ConfigMaps stores the configuration of an application in a normal plain text format. However, both of them can be used as volume and be mounted on a pod through a pod definition file.

31. Explain Kubernetes RBAC.

[Hide Answer](#)

It stands for Role-Based Access Control. It allows the management of permissions for users and groups inside a Kubernetes cluster. With RBAC, one can set access permissions and the required conditions to fulfill to gain access. Roles define a set of permissions granted to users or groups, and role bindings associate roles with specific users or groups.

Using RBAC, you can limit access to sensitive resources and ensure that only authorized users and applications can perform certain actions within the cluster. This helps to improve the security and reliability of your Kubernetes environment.

32. How can you get a static IP for a Kubernetes load balancer?

must also check the Kubernetes version and allocate static IPs under Networking > External IP addresses. You can follow these steps for assigning a static IP:

- Reserve a static IP address from your cloud provider's pool of available IPs.
- Create a Kubernetes service of type "LoadBalancer."
- Assign the reserved static IP address to the service using the loadBalancerIP field in the service definition.
- Check that the service uses the static IP address by inspecting the service using the `kubectl describe service`.

33. What tools will you use for container orchestration?

[Hide Answer](#)

Some tools we can use for container orchestration are:

Docker Swarm: It is an orchestration management tool running on Docker applications that helps in the creation and deployment of Docker nodes.

Apache Mesos: It is a cluster manager that helps in resource isolation and sharing.

Kubernetes: Kubernetes (commonly abbreviated K8s) is an open-source container orchestration system that automates software deployment, scaling, and management.

34. Can you list some of the objects of Kubernetes?

[Hide Answer](#)

Here are some of the objects used by Kubernetes:

- Pods
- Controllers and replication set
- Cron jobs

- Stateful sets
 - Deployments
-

35. Define StatefulSets.

[Hide Answer](#)

StatefulSets are a type of workload API that manage stateful applications. They can also be used to manage the scaling and deployment of pod sets. StatefulSets are often used to manage the deployment and scaling of pods that require stable network identities and persistent storage, making them well-suited for stateful workloads.

36. Can you list some reasons why DaemonSets are used?

[Hide Answer](#)

DaemonSets are used for the following reasons:

Running system-level services: DaemonSets are often used to run system-level services that need to be deployed on every node in the cluster, such as logging agents, monitoring agents, or network services like load balancers.

Data collection: DaemonSets can collect data or metrics from each node in the cluster, such as system metrics, network traffic, or application logs.

Consistent deployment: By using DaemonSets to deploy services or applications, you can ensure that they are deployed consistently across all nodes in the cluster, which can help reduce configuration errors and ensure consistent operation.

High availability: Running services or applications as DaemonSets can provide high availability, as they can be automatically deployed on any new nodes added to the cluster and replaced if a node fails or becomes unavailable.

Resource isolation: Running services or applications as DaemonSets can provide resource isolation, as they can be deployed on specific cluster nodes with the necessary

37. What is the purpose of Operators?

[Hide Answer](#)

In Kubernetes, managing applications isn't as simple as maintaining stateless applications where each replica reaches the required status and updates are done the same way. Because of the stateful nature of stateful applications, upgrading each replica may necessitate distinct treatment. As a result, managing stateful applications frequently necessitates using a human operator. This is where Kubernetes Operator is useful.

Kubernetes Operators are software extensions that automate the management of complex applications on Kubernetes clusters. Operators are designed to be highly customizable and adaptable to the needs of specific applications. They can perform tasks such as monitoring the health of an application, performing backups and restores, upgrading or rolling back versions, scaling resources up or down, and more.

38. Can you tell me some Kubectl commands?

[Hide Answer](#)

Some Kubectl commands are:

- kubectl annotate
- kubectl config
- kubectl autoscale
- kubectl config current-context
- kubectl apply
- kubectl config set.
- kubectl cluster-info
- kubectl attach

[Hide Answer](#)

Kube-apiserver is used to configure and validate API objects. Also, it is used to enable users to interact with the Kubernetes cluster where the Kubernetes server API is used. Thus, it is the front-end part of the control plane, and it is this frontend that tests whether the incoming request is legitimate or not.

A Kube-apiserver validates and provides data for API objects and includes pods, services, and replication controllers. The API server sends command REST operations and provides the frontmost to the cluster's shared state through which all other components interact.

Syntax:

```
kube-apiserver [flags]
```

Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

[Apply Now](#)

ADVANCED KUBERNETES INTERVIEW QUESTIONS AND ANSWERS

1. What you will do to upgrade a Kubernetes cluster?

[Hide Answer](#)

To update the Kubernetes cluster, one needs to update the components of the cluster. and the specific instructions to upgrade the same depends on the specific setup and configuration used with your cluster.

Here's the upgrade process:

- Now, upgrade the control plane components.
 - Also, upgrade the worker nodes, one by one, through a rolling upgrade process.
 - Finally, verify whether the cluster and the components are functioning properly.
-

2. Explain the two types of Kubernetes pods.

[Hide Answer](#)

The two Kubernetes pods are single-container pods and multi-container pods. Here's a brief explanation of each.

Single-container pods: These pods contain only one container and are the most common type of pod used in Kubernetes. They can be created using commands such as `kubectl run` or `kubectl create`.

Multi-container pods: These pods contain multiple containers that are tightly coupled and need to run together on the same host. Multi-container pods are created using the `kubectl create` command with a YAML file that defines the pod's configuration..

3. Tell me about labels in Kubernetes.

[Hide Answer](#)

Labels are added to objects during their creation and can be altered by the users at run time. They are precisely collections of keys that hold some values.

They are a type of metadata that work through the concept of key-value pair attached to objects. In many cases, they are also used to define "identifying aspects" of an object mainly to help the user at a later stage. Also, as they do not imply semantics to Kubernetes by default, they do not directly change any functionality like other metadata.

[Hide Answer](#)

Some of the objectives of a replication controller are:

- Control and administer pod lifecycle
 - Monitor and verify the permissible number of replicas
 - Help user to check pod status
 - Enable the alteration of pod
-

5. What is persistent volume?

[Hide Answer](#)

Persistent volume (PV) is a storage piece that an administrator provides to a Kubernetes cluster. Suppose a developer needs persistent storage for an application in a cluster; they do so by creating a persistent volume claim (PVC) and then mounting the volume to a path in the pod. Once this process is done, the pod can claim any volume matching its requirements.

6. Do you know anything about the Secrets in Kubernetes?

[Hide Answer](#)

Yes, the Secrets in Kubernetes are objects that hold sensitive information such as login credentials for the user, passwords, OAuth tokens, and SSH keys. Here "Secrets" give you more control over this information and its accidental exposure. Using Secrets, Kubernetes ensures that the sensitive data is encrypted, only authorized entities can access it, and it is not exposed to unauthorized users or applications.

7. Tell me about Sematext Docker Agent.

on a Docker host. It collects and streams logs, events, and metrics from Docker containers and nodes to Sematext Cloud, a cloud-based monitoring and logging platform. With Sematext Docker Agent, users can monitor their Docker environments in real-time, troubleshoot issues quickly, and gain insights into container performance. Sematext Docker Agent supports Kubernetes, Docker Swarm, and other container orchestration systems.

8. Explain OpenShift.

[Hide Answer](#)

It is a public cloud Kubernetes platform that aids developers in building applications. Its offerings include automated installation, life cycle management, and upgrades. OpenShift provides developers with tools and services to build, deploy, and manage applications in a containerized environment. These tools include an integrated development environment (IDE), container images, source code management, automated builds, and deployments.

9. Mention some differences between Kubernetes Volumes and Docker Volumes .

[Hide Answer](#)

Not limited to any container	Limited to the pod in a container
Support all containers deployed in the pod	Does not support all the containers deployed
The Kubernetes cluster manages Kubernetes volumes.	The Docker engine manages Docker volumes
Kubernetes volumes are tied to the pod's lifecycle, which means that they are created when the pod is created and destroyed when the pod is terminated.	Docker volumes have a lifecycle that is independent of the container that uses them, which means that they can persist even after the container is removed.

10. How can you provide API-Security on Kubernetes?

[Hide Answer](#)

Securing APIs on Kubernetes is crucial to prevent unauthorized access and data breaches. One approach is to implement Role-Based Access Control (RBAC) policies with the kube-dashboard, which allows defining permissions for individual users or groups. Another option is to use Kubeless, a Kubernetes-native serverless framework, to protect APIs by automatically injecting security measures such as rate limiting and authentication. Additionally, it's important to use the correct authorization mode, such as the Node Authorizer or Attribute-Based Access Control (ABAC), to ensure that only authorized requests are accepted. By combining these methods, Kubernetes can provide strong API security for your applications.

11. How do you debug a POD that isn't being scheduled?

[Hide Answer](#)

The scheduler in K8 is in charge of spawning pods into nodes. A variety of circumstances can cause unstartable POD. The most typical reason for POD not starting is a lack of resources; use commands like `kubectl describe POD > -n Namespace >` to find out why. Also, watch `kubectl get events` to observe the cluster's events.

[Hide Answer](#)

The different types of Kubernetes volumes are as follows:

EmptyDir: This volume is first created when a node is assigned with a pod. Initially, it is empty. A volume of type emptyDir is available for the lifetime of the pod.

Flocker: It is an open-source and clustered container data volume manager.

HostPath: This volume mounts a file or directory from the host node's filesystem into the pod. It can provide access to host files or share files between containers on the same host.

NFS: Network File System (NFS) allows computers to either access or share files over the network. It is a dedicated file storage when multiple users must retrieve data for centralized disk capacity.

13. What is PVC?

[Hide Answer](#)

PVC stands for Persistent Volume Claim, a request made by a user for storage in a Kubernetes cluster. When a user creates a PVC, Kubernetes matches it with an appropriate Persistent Volume (PV) based on the storage requirements and properties specified in the PVC.

Persistent Volumes are network-attached storage devices that multiple pods can access in a cluster. A PVC is a request for a specific amount of storage, which is bound to a single pod or multiple pods, depending on the user's requirements.

14. What do you know about Kubernetes network policy?

[Hide Answer](#)

define rules that specify which traffic can flow between pods and which traffic should be blocked.

15. What are the limitations of using the default namespace?

[Hide Answer](#)

Using the default namespace alone makes it difficult to keep track of all the applications you can manage in your cluster over time. Custom namespaces help in grouping apps into logical groupings, such as one namespace for monitoring programs and another for security applications, and so on. This makes it easier to manage and monitor different applications within your cluster and control access to specific resources.

16. How can you safely drain the K8s node?

[Hide Answer](#)

We can use the following commands

- Use the command 'kubectl cordon' to mark the node as "un-schedulable". This moves the node to maintenance mode and prevents new pods from being scheduled on it.
 - The 'kubectl drain --ignore-daemon-set' command removes the existing pods from the node. The '--ignore-daemon-set' option will skip over any pods that are part of a DaemonSet, which ensures that essential system services are not disrupted.
-

17. How can you perform maintenance in a single pod?

[Hide Answer](#)

Here are the steps for performing maintenance in a single pod:

- Get the name of the pod you want to perform maintenance using the command

- Verify that the label has been applied to the pod
 - Perform maintenance on the pod as needed
 - Remove the maintenance label from the pod when you're done
 - Verify that the label has been removed
-

18. How can you control the usage of resources in a pod?

[Hide Answer](#)

We can control the resource usage of the pod through the use of limit and request.

Request: It controls the minimum guaranteed amount of resources (CPU and memory) that are required for a container.

Limit: It puts a limit on the resources any single container can make use of. This means that the container can never consume more than the memory amount or CPU amount indicated.

You can think of it as request is the minimum amount of resources to be used by a pod while limit is the maximum amount of resources that pod can use.

19. List various K8's services that can run on nodes.

[Hide Answer](#)

Kubernetes (K8s) runs various services on nodes, including:

On Worker Nodes:

kubelet: Responsible for node management and ensuring that containers are running as expected.

kube-proxy: Provides network proxy services to enable communication between nodes and services.

On Master Nodes:

kube-scheduler: Assigns newly-created pods to a node based on resource availability and other constraints.

kube-controller-manager: Monitors the cluster state and performs tasks to maintain desired state, such as scaling up or down.

Other services that may run on worker nodes include kube-dns, which provides DNS resolution for Kubernetes services, and kubelet, which manages pod lifecycles on the node.

20. What is PDB (Pod Disruption Budget)?

[Hide Answer](#)

The PDB (Pod Disruption Budget) allows you to limit the disruption to application use when its pods need to be rescheduled for some other reason on the Kubernetes nodes.

Kubernetes' Pod Disruption Budget (PDB) is a handy layer of defense for dealing with this type of issue. If you already have a Kubernetes cluster in production, the command 'kubectl get pdb --all-namespaces' will show you all the PDBs in all the namespaces of your Kubernetes cluster.

A Kubernetes administrator can make a one-of-a-kind deployment. Pod Disruption Budget ensures that the application's minimum number of running pods is preserved, as specified by the property minAvailable in the spec file. It is useful while executing a drain, as the drain will stop until the PDB is respected, ensuring the application's High Availability (HA). In the following spec file, minAvailable is set to 2, implying that the minimum number of available pods is two (even after the election).

21. Why should we use custom namespaces ?

[Hide Answer](#)

If you use default namespaces alone, it gets difficult in the long run since you will not be able to manage them within a cluster as a whole. By default, all resources in a

With custom namespaces, this problem is resolved. By creating custom namespaces, you can logically group your resources based on your needs, such as separating production and development environments or separating applications by team or department. This makes it easier to manage and maintain your resources within the cluster, and also provides better security and resource isolation.

22. What are the various logging patterns you can use to get central logs from pod?

[Hide Answer](#)

There are some common logging patterns as follows:

Node level logging agent: This involves installing a logging agent directly on the node where the pod is running. The logging agent collects logs from all the containers on that node and sends them to a central logging server.

Streaming sidecar container: In this pattern, a separate container, known as a sidecar, is added to the pod. The sidecar container is responsible for streaming the logs from the other containers in the pod to a central logging server.

Sidecar container with the logging agent: Similar to the streaming sidecar container pattern, a separate container is added to the pod. However, in this pattern, the sidecar container also includes a logging agent that collects logs from the other containers in the pod and streams them to a central logging server.

Fluentd logging pattern: This pattern involves deploying Fluentd, a unified logging layer, to capture, filter, and forward logs generated by the main container to a central logging system.

23. In the service given below, how can you convert the spec into an external one?

```
ports:
  - protocol: UDP
    port: 9090
    targetPort: 9080
```

[Hide Answer](#)

By using LoadBalancer as follows

```
spec:
  selector:
    app: some-app
  type: LoadBalancer
  ports:
    - protocol: UDP
      port: 9090
      targetPort: 9080
```

The type LoadBalancer field creates an external load balancer to distribute traffic to the service across the worker nodes.

24. Complete the following configuration spec file and turn it Ingress:

```
metadata:
  name: someapp-ingress
Spec:
```

[Hide Answer](#)

```
metadata:
  name: someapp-ingress
spec:
  rules:
  - host: my.host
    http:
      paths:
      - backend:
          serviceName: someapp-internal-service
          servicePort: 9090
.
```

25. Can you schedule the pods to the node if the node is tainted?

[Hide Answer](#)

If a node is tainted, pods will not be scheduled on it by default, but you can use tolerations in the pod spec to allow specific pods to be scheduled on the tainted node. Tolerations are used to specify that a pod can tolerate (or "ignore") a certain taint, allowing it to be scheduled on a tainted node. This can be useful in scenarios where you want to reserve certain nodes for specific types of workloads or to mark nodes as unsuitable for certain workloads.

26. How can you achieve zero downtime in Kubernetes?

[Hide Answer](#)

We can achieve zero downtime in Kubernetes through RollingUpdate strategy. It is a process that allows updating the Kubernetes system with little effect on performance and zero downtime. This strategy involves gradually replacing old instances of an application with new ones, thus ensuring that the application is always available to end-users.

When you use RollingUpdate strategy, Kubernetes creates a new replica set with the updated version of your application and gradually replaces the old replica set with the

27. How can you check that a pod always runs?

[Hide Answer](#)

We can check that a pod always runs through a liveness probe. In case the check fails, it restarts the container. This probe is helpful in multiple instances when the container runs but the application crashes.

Example:

```
spec:
  containers:
  - name: liveness
    image: k8s.gcr.io/liveness
    args:
    - /server
    livenessProbe:
      httpGet:
        path: /heal
```

In this example, the `/heal` endpoint of the `k8s.gcr.io/liveness` container is checked by the liveness probe. If the probe fails to receive a response from this endpoint, Kubernetes will automatically restart the container. This helps ensure that the pod always runs, even if the container encounters issues.

28. Can you tell if the rolling update with stateful set replicas = 1 looks right?

[Hide Answer](#)

No, it doesn't look right since there is only 1 replica, and any changes to the stateful set may result in an outage. It means that the rolling update of a StatefulSet needs to rip apart one or more old pods before replicating them. Therefore, it's generally recommended to have at least two replicas in a StatefulSet to ensure high availability during rolling updates.

[Hide Answer](#)

In such cases, SIGKILL instantly ends the container and spawns a new one using out of memory (OOM) error. This will result in the kernel sending a SIGKILL signal to the process running inside the container. By default, Kubernetes will send a SIGTERM signal to the container's main process and wait for a certain period of time (determined by the `terminationGracePeriodSeconds` parameter) for the process to exit gracefully.

30. How can you run a pod on a specific node?

[Hide Answer](#)

We can run a pod on a specific node using node affinity. Here, a node is assigned to an arbitrary label, and they are configured to be assigned to that node as per the label created.

For example, this code snippet creates an arbitrary label, node location, and assigns the value Germany to the node named person-01:

```
kubectl label nodes person-01 nodelocation=Germany
```

31. What happens when the master or the worker node fails?

[Hide Answer](#)

Inside Kubernetes, when the master node fails, the cluster remains operational. There is no effect on the pod creation or any service member changes.

If a worker node fails, the master stops receiving updates, and any pods running on the failed worker node are terminated. The Kubernetes scheduler will then create new replicas of the terminated pods on other healthy worker nodes in the cluster.

[Apply Now](#)

WRAPPING UP

The above-mentioned Kubernetes technical interview questions will help the candidates improve their interview preparation. It will also help recruiters weigh the candidate's skills appropriately.

With good coverage across basic, medium, and advanced-level Kubernetes interview questions and answers, we have tried to cover the most popular questions. If you are a developer, try the Turing test and grab the chance to work with the top U.S. companies from the comfort of your home. And, if you are a recruiter and want to skip the lengthy interview process, Turing can help you source, vet, match, and manage the world's best [Kubernetes developers](#) remotely.

Get remote Kubernetes developer jobs with top U.S. companies!

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

[Apply now](#)

Apply for the latest remote jobs

neer

[JOB](#)

 11-50  Software

Senior Softwar

[FULL-TIME REMOTI](#)

 Posted a year ago 



Job description templates →

Learn how to write a clear and comprehensive job description to attract highly skilled Kubernetes developers to your organization.



Kubernetes developer resume tips →

Turing.com lists out the do's and don'ts behind a great resume to help you find a top remote Kubernetes developer job.

Check out more interview questions

 React.js |

Based on your skills

 React

 Node.js

 Python

 AWS

 JavaScript

 HTML

 Java

 Flutter

 C#

 SQL

 Angular

iOS iOS

 PHP

[+ See more skills](#)

Based on your role

 Front-end

 Full Stack

 Backend

 Machine Learning

 Remote Developer Software Engineer[+ See more roles](#)

Hire remote developers

Tell us the skills you need and we'll find the best developer for you in days, not weeks.

Hire Developers

AI & AGI solutions

LLM training

Generative AI

AI/Data

Custom engineering

All solutions →

On-demand talent

Technical professionals and teams



[How to get hired](#)

[Developer reviews](#)

[Talent resources](#)

[Tech interview questions](#)

Resources

[Blog](#)

[Case studies](#)

[Use cases](#)

[More resources](#) →

Company

[About](#)

[Press](#)

[Turing careers](#)

Connect

[Contact us](#)

[Help center](#)

[Sitemap](#)

[Terms of service](#)

[Privacy policy](#)



1900 Embarcadero Road Palo Alto, CA, 94303

© 2025 Turing