



Courses

[Free Courses](#) [Interview Questions](#) [Tutorials](#) [Community](#)

Home / Interview Question / Top Kubernetes Interview Questions and Answers

Top Kubernetes Interview Questions and Answers

By Naveen 7.1 K Views 21 min read Updated on April 22, 2022

We have given a list of top Kubernetes interview questions to help you take your career to the next level. We have included the benefits of Kubernetes, comparison with Docker Swarm, Kubernetes pod, node, Heapster, container cluster, kubelet, GKE, kube-proxy, and more in this blog. **Learn Kubernetes from Intellipaat's DevOps Course and fast-track your career!**

Become a Certified Professional

Categories	
Automation	3
Big Data	12
Business Intelligence	21
Cloud Computing	19
Cyber Security	2
Data Science	11
Database	6
Digital Marketing	2
Mobile Development	2
No-SQL	5
Programming	16
Project Management	5
Salesforce	3
Testing	4
Website Development	6

Top Answers to Kubernetes Interview Questions

Kubernetes is the best and among the most sought-after orchestration tools in the market today. This tool has become a favorite of leading companies, including The New York Times, SAP, Huawei, eBay, and more. However, there are only a few of those professionals who can meet the demands and fill the rising need for Kubernetes-certified professionals with their skills. With the help of this Kubernetes Interview Questions blog, we will cover some of the frequently asked questions that you may face during job interviews. So, prepare the following Kubernetes interview questions before you go for your

you may face during job interviews. So, prepare the following Kubernetes interview questions before you go for your interview:

- [Q1. What is Kubernetes?](#)
- [Q2. What are the benefits of Kubernetes?](#)
- [Q3. What is a Kubernetes cluster?](#)
- [Q4. What is Kubernetes used for?](#)
- [Q5. How does Kubernetes work?](#)
- [Q6. What is the difference between Kubernetes and Docker Swarm?](#)
- [Q7. What is orchestration in software?](#)
- [Q8. What is a Kubernetes namespace?](#)
- [Q9. What are federated clusters?](#)

Download Salary Trends

Learn how professionals like you got upto 100% hike!

Email Address

+48 PL



Phone Number

Submit

[1. Basic](#)

[2. Intermediate](#)

[3. Advanced](#)

Watch this video on Kubernetes Interview Questions and Answers:



Basic Interview Questions

1. What is Kubernetes?

First, let us compare Kubernetes with Docker Swarm:

Comparison	Kubernetes	Docker Swarm
Controller	Master	Manager
Slave	Nodes	Workers
Deployment unit	Pod	Task

Load balancing	Service	Ingress

Kubernetes is a container orchestration tool that is used for automating the tasks of managing, monitoring, scaling, and deploying containerized applications. It creates groups of containers that can be logically discovered and managed for easy operations on containers.

2. What are the benefits of Kubernetes?

With the container orchestration tool Kubernetes, it becomes extremely easy to handle containers. We can respond to customer demands by deploying the applications faster and in a more predictable manner.

Here, we will list some of the benefits of Kubernetes:

- Automatic scheduling
- Automated rollback
- Horizontal scaling
- Auto-healing capabilities

Prepare yourself for the industry with our [Top DevOps Interview Questions and Answers!](#)

3. What is a Kubernetes cluster?

A Kubernetes cluster is a group of nodes that run containerized applications across various environments and machines—cloud-based, physical, virtual, and on-premises. It enables the easy development of applications as well as their management and movement.

4. What is Kubernetes used for?

Kubernetes is used for the automation of the manual operations that are involved in the deployment, management, and scaling of containerized applications. It keeps track of the ones that are deployed into the cloud, restarts orphaned ones, shuts down the unused, and automatically provides resources such as storage, memory, and CPU when required.

5. How does Kubernetes work?

The best way to carry out the management of the life cycle of containerized applications over a large scale is through a container orchestration system like Kubernetes. It automates the deployment and scaling of several containers simultaneously. Containers that are running the same application are arranged together and act as replicas. They serve to load balance incoming requests. Kubernetes, then, supervises these groups of containers and ensures that they are functioning correctly.

Read in short about Docker in the [Docker Cheat Sheet](#).

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

Docker Swarm is a default container orchestration tool that comes with Docker. Docker Swarm can only orchestrate simple Docker containers. Kubernetes, on the other hand, helps manage much more complex software application containers. Kubernetes offers support for larger demand production environment.

7. What is orchestration in software?

Application orchestration in the software process means that we can integrate two or more applications. We will be able to automate arrangement, coordination, and management of computer software. The goal of any orchestration process is to streamline and optimize frequent repeatable processes.

Go through the [Best DevOps Course in New York](#) to get a clear understanding of DevOps!

GET 100% HIRE!

Master Most in Demand Skills Now !

Email Address

+48 PL ▼

Phone Number

Submit

8. What is a Kubernetes namespace?

The Kubernetes namespace is used in the environment wherein we have multiple users spread in the geographically vast areas and working on multiple projects. What the namespace does is dividing the cluster resources between multiple users.

9. What are federated clusters?

Multiple clusters that are managed as a single cluster is referred to as federated clusters.

Intermediate Interview Questions

10. What is a pod in Kubernetes?

We can think of a Kubernetes pod as a group of containers that are run on the same host. So, if we regularly deploy single containers, then our container and the pod will be one and the same.

11. What is a node in Kubernetes?

A node in Kubernetes is a worker machine which is also known as a minion. This node could be a physical machine or a virtual machine. For each node, there is a service to run pods, and it is managed by master components. The node services could include kubelet, kube-proxy, and so on.

Learn more on DevOps in this [DevOps Training in Sydney](#) to get ahead in your career!

12. What is a Heapster?

The Heapster lets us do the container cluster monitoring. It lets us do cluster-wide monitoring and event data aggregation. It has native support for Kubernetes.

13. What is a container cluster?

A container cluster lets us place and manage containers in a dynamic setup. It can be considered as a set of nodes or Compute Engine instances. The API server of Kubernetes does not run on cluster nodes, instead the Container Engine hosts the API server.

If you have any doubts or queries related to DevOps, get them clarified from DevOps experts on our [DevOps Community](#)!

14. What is a kubelet?

We can think of a kubelet as the lowest level component in a Kubernetes. The kubelet is responsible for making the individual machines run. The sole purpose of a kubelet is that in a given set of containers, it has to ensure that they are all running.

15. How to write a Kubernetes scheduler?

The kube-scheduler is the default scheduler for Kubernetes. It is designed such that if you prefer, you can write your own

one and use that instead.

Following is the syntax:

```
kube-scheduler [flags]
```

The scheduling life cycle:

1. A pod is created and the preferred state is mentioned, and without filling the node name, it is saved to etcd
2. The scheduler notices the new pod with no node bound
3. It finds a suitable node for that pod
4. It then informs the API server to bind the pod to the node, and next, the new desired state is saved to etcd
5. Kubelets watch the pods that are bound and start the containers on the particular node

Interested in becoming a DevOps expert? Click here to learn more in this [DevOps Course in Toronto!](#)

Career Transition

16. What are the ways to provide API Security on Kubernetes?

Following are some of the ways that provide API Security:

- Using the correct auth mode with the API server authentication mode= Node, RBAC
- Ensuring that the traffic is protected by TLS
- Using API authentication
- Ensuring that kubeless protects its API via authorization-mode=Webhook
- Monitoring RBAC failures
- Removing default Service Account permissions
- Ensuring that the kube-dashboard applies a restrictive RBAC policy
- Implementing a pod security policy for container restrictions and the protection of the node
- Using the latest version of kube

17. If an organization is looking for ways to improve its deployment methods and desires a more scalable and responsive platform, what should be done?

The company should move to a cloud environment and implement a microservice architecture for implementing Docker containers. Once the base framework is set up, Kubernetes can be used for the autonomous development of applications and the quick delivery of the same by the team.

Interested in getting an industry-recognized certification in DevOps? Enroll in Intellipaat's [DevOps Course in Bangalore](#) now!

18. If an organization has a large distributed system with several data centers, virtual machines, and a huge number of employees working on various tasks, how can the tasks be managed with consistency with the help of Kubernetes?

The company can do well with something that offers scale-out capability, agility, and the DevOps practice to the cloud-based applications. Kubernetes, in this situation, can enable the customization of the scheduling architecture and support multiple container formats. This results in greater efficiency as well as provides support for various container networking solutions and container storage.

19. What is the difference between a replica set and a replication controller?

The difference is mainly in the selectors used for pod replication. A replica set uses set-based selectors, and replication controllers use equity-based selectors.

20. How does Kubernetes scale?

The **kubectl scale** command enables the ability to instantly change the number of replicas needed for running an application. While using this command, the new number of replicas need to be specified by setting the **-replicas** flag.

Prepare yourself for the DevOps certification with this comprehensive [DevOps Training in Hyderabad!](#)

21. What is a Kubernetes context?

A context is a group of access parameters that has a cluster, a user, and a namespace. The current context is the cluster that is currently the default for kubectl, and all kubectl commands run against that cluster.

22. Give examples of some recommended security measures for Kubernetes.

- Defining resource quotas
 - Auditing support
 - Providing restricted access to etcd
 - Regular security updates
 - Network segmentation
 - Strict resource policies
 - Regular scans for security vulnerabilities
-
- Using images from repositories that are authorized

If you wish to get certified in DevOps, check out this [DevOps Certification](#) blog!

23. What is a Headless Service?

The headless service is like normal services but without the Cluster IP. It enables direct access to pods without the need for a proxy.

Courses you may like

Advanced Interview Questions

24. What is Minikube?

The Minikube makes it easy for the local running of Kubernetes. Within a virtual machine, the Minikube runs a single-node Kubernetes cluster.

25. What is Kubectl?

Kubectl is a Kubernetes command-line tool that is used for deploying and managing applications on Kubernetes. Kubectl is

especially useful for inspecting the cluster resources, and for creating, updating, and deleting the components.

26. What is GKE?

GKE is Google Kubernetes Engine which is used for managing and orchestrating systems for Docker containers. GKE also lets us orchestrate container clusters within the Google Public Cloud.

27. What is kube-proxy?

The kube-proxy runs on each of the nodes. It can do simple tasks such as TCP, UDP, forwarding, and so on. It shows the services in the Kubernetes API on each node.

28. What are the components of a Kubernetes Master?

The components of the Kubernetes Master include the API server, the controller manager, the Scheduler, and the etcd components. The Kubernetes Master components are responsible for running and managing the Kubernetes cluster.

29. What is the use of kube-controller-manager?

It is the Kubernetes Controller Manager. The kube-controller-manager is a daemon that embeds the core control loops which regulate the system state, and it is a non-terminating loop.

30. What is load balancing on Kubernetes?

The process of load balancing will let us expose services. There are two types of load balancing when it comes to Kubernetes:

- Internal load balancing: This is used for balancing the loads automatically and allocating the pods with the required configuration.
- External load balancing: This directs the traffic from the external loads to the backend pods.

Also, check why [Kubernetes and DevOps](#) are a perfect pair for handling DevOps!

31. What does a kube-scheduler do?

The kube-scheduler has the job of assigning the nodes to the newly created pods.

32. Where is the Kubernetes cluster data stored?

The primary data store of Kubernetes is etcd, which is responsible for storing all Kubernetes cluster data.

33. How to set a static IP for Kubernetes load balancer?

Kubernetes Master assigns a new IP address.

We can set a static IP for Kubernetes load balancer by changing the DNS records whenever Kubernetes Master assigns a new IP address.

34. Explain the Kubernetes architecture.

Pods

Pods are the smallest units that Kubernetes administers. It constitutes a set of containers. It shares a single IP address and all the resources, such as storage and memory, among every container within it. A pod can have a single container when the service or application is a single process.

Deployments

Kubernetes deployments determine the scale at which one wants to run an application, such as how the pods need to be replicated on the Kubernetes nodes, the desired number of pod replicas to be run, and the desired update strategy for the deployment.

Services

If a pod dies, Kubernetes replaces it to prevent any downtime. A service is the only interface that the application consumers deal with. When pods are changed, their internal names and IPs might change as well. A service exposes a single IP address or machine name linked to pods whose numbers and names are unreliable. It ensures that nothing appears changed to the outside network.

Nodes

A Kubernetes node collects, runs, and manages pods that function together.

The Kubernetes Control Plane

The Kubernetes control plane is the main entry point for users and administrators to handle the management of various nodes. HTTP calls or command-line scripts are used to assign operations to it. How Kubernetes interacts with applications is controlled by the control plane.

Cluster

The above components put together in a single unit is referred to as a cluster.

Kubernetes Components:

The control plane and the individual nodes consist of three main components each.

Control plane

- **API Server**

The Kubernetes API server validates and configures data for API objects, including pods, replication controllers, services, etc. It serves REST operations and provides the frontend to the cluster's shared state through which all other components communicate.

- **Scheduler**

The scheduler assigns work to the nodes, keeps track of the capacity of resources, and ensures that a worker node's operation is within the right threshold.

- **Controller Manager**

The controller manager ensures that a cluster's shared state is operating in the desired manner. It monitors various controllers, which respond to events.

Worker Node Components:

- **Kubelet**

A kubelet keeps track of the state of a pod and ensures that every container is operating well.

- **Kube proxy**

The kube proxy is a network proxy that maintains network rules on nodes. It sends requests for work to the appropriate containers.

- etcd

This etcd component manages and holds the critical data that distributed systems require to operate. It is an open-source distributed key-value store that is used to share the state of a cluster. It helps with the setup of the overlay network for containers.

35. What do I need on-premises to run the Kubernetes architecture?

Many on-premises environments are remodeled to enable Kubernetes integration. Integrating storage, servers, and networking into a smoothly running environment requires top skills. For Kubernetes, deciding the right storage and networking equipment is crucial as it facilitates interaction with resources for storage, load balancers, etc. A critical part of Kubernetes’ value proposition is the ability to automate storage and the networking components.

[Next](#)

Course Schedule

Name	Date	
AWS Certification	2022-05-28 2022-05-29 (Sat-Sun) Weekend batch	View Details
AWS Certification	2022-06-04 2022-06-05 (Sat-Sun) Weekend batch	View Details
AWS Certification	2022-06-11 2022-06-12 (Sat-Sun) Weekend batch	View Details

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Find Best DevOps Training in Other Regions

<u>Bangalore</u>	<u>Melbourne</u>	<u>Chicago</u>	<u>Hyderabad</u>	<u>San Francisco</u>	<u>London</u>	<u>New York</u>	<u>Toronto</u>	<u>India</u>	<u>Chennai</u>	<u>Delhi</u>
<u>Gurgaon</u>	<u>Kolkata</u>	<u>Mumbai</u>	<u>Pune</u>	<u>Singapore</u>	<u>Dubai</u>	<u>Houston</u>	<u>Jersey City</u>	<u>Los Angeles</u>	<u>Noida</u>	<u>Ahmedabad</u>
<u>Coimbatore</u>	<u>Sydney</u>	<u>Dallas</u>	<u>Atlanta</u>	<u>Michigan</u>	<u>Manchester</u>	<u>Montreal</u>				