

Most Asked Interview Questions of Kubernetes

Following is the list of most asked interview questions of Kubernetes with their best possible answers:

1) What is Kubernetes? / What do you understand by Kubernetes?

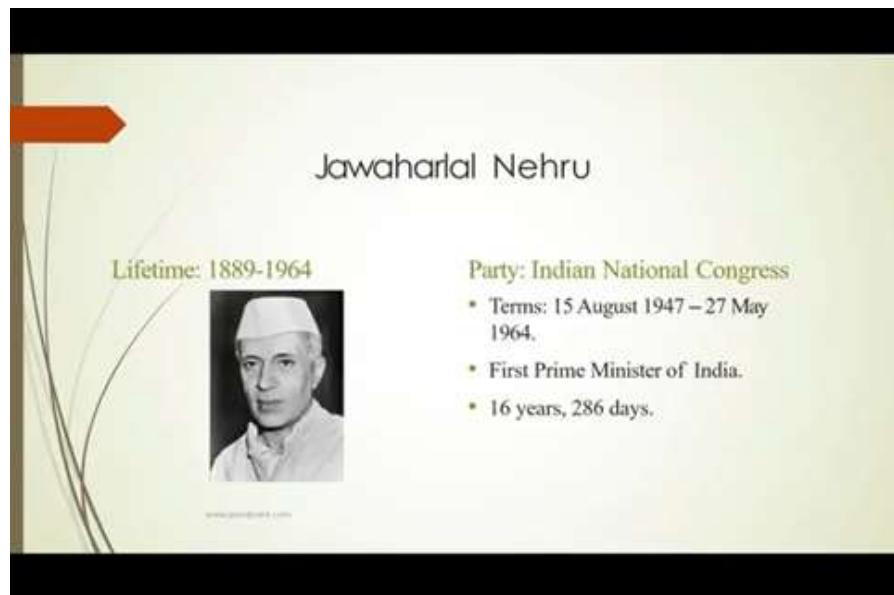
Kubernetes is an open-source container-orchestration tool or system used to automate tasks such as the management, monitoring, scaling, and deploying containerized applications.

2) What is the use of Kubernetes?

Kubernetes is mainly used to easily manage several containers (since it can handle the grouping of containers), which provides logical units that can be discovered and managed.

3) Who was the inventor of Kubernetes?

Kubernetes was initially designed and developed by Google and is now maintained by the Cloud Native Computing Foundation.



4) What are K8s?

K8s is nothing but just another term for Kubernetes.

↑ SCROLL TO TOP

5) What was the main motive behind the development of Kubernetes?

The main motive behind the development of Kubernetes is to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts."

6) What do you understand by the term orchestration when it comes to software and DevOps?

The term orchestration specifies integrating multiple services that allow them to automate processes or synchronize information in a specific time sequence.

For example, suppose we have six or seven microservices for an application to run, then if you place them in separate containers, this would inevitably create obstacles for communication. Using orchestration, we can do it quickly to enable all services in individual containers to work seamlessly to accomplish a single goal.

7) What is the relation between Docker and Kubernetes?

Docker is an open-source platform used to handle software development. It is mainly used to package the settings and dependencies that the software/application needs to run into a container, which allows for portability and several other advantages. On the other hand, Kubernetes is used to allow the manual linking and orchestration of several containers, running on multiple hosts that have been created using Docker.

8) What are the key differences between the Docker Swarm and Kubernetes?

Docker Swarm is an open-source container orchestration platform used to cluster and schedule Docker containers. It is a native of Docker. Following is the list of key differences between the Docker Swarm and Kubernetes:

Docker Swarm	Kubernetes
Docker Swarm is easy and convenient to set up, but it doesn't have a robust cluster.	Kubernetes is more complicated than Docker Swarm to set up, but it assures a robust cluster.
↑ SCROLL TO TOP	

Docker Swarm can't do auto-scaling as the Kubernetes can do, but Docker's scaling is five times faster than Kubernetes.	Kubernetes can do auto-scaling, but scaling is slower than Docker Swarm.
Docker Swarm doesn't provide a GUI.	Kubernetes provides a GUI in the form of a dashboard.
Docker Swarm provides an automatic load balancing feature of traffic between containers in a cluster.	It requires manual intervention in Kubernetes for load balancing such traffic.
Docker requires third-party tools such as the ELK stack for logging and monitoring.	Kubernetes provides such integrated tools for logging and monitoring purposes.
In Docker Swarm, we can easily share storage volumes with any container.	In Kubernetes, we can only share storage volumes with containers in the same pod.
We can deploy rolling updates in Docker Swarm but can't deploy automatic rollbacks.	In Kubernetes, we can deploy rolling updates as well as automatic rollbacks.

9) What do you understand by a node in Kubernetes?

In Kubernetes, a node is the smallest unit of hardware. It is used to define a single machine in a cluster that can act as a virtual machine from a cloud provider or physical machine in the data center. Every machine of the Kubernetes cluster can act as a substitute for other machines.

10) What is the use of a Kubernetes Kube-scheduler?

A Kube-scheduler is the default scheduler for Kubernetes. It is used to assign nodes to newly created pods.

↑ SCROLL TO TOP

11) What do you understand by daemon sets in Kubernetes?

Daemon sets are sets of pods that run on a host and are used for host layers attributes like monitoring network or simple network.

12) What is Heapster in Kubernetes?

A Heapster is a metrics collection and performance monitoring system for data collected by the Kublet.

13) What are the main reasons behind using Kubernetes?

Kubernetes is mainly used because of the following reasons:

- We can use Kubernetes easily on on-premises bare metal, OpenStack, public clouds Google, Azure, AWS, etc.
- Using Kubernetes, we can avoid vendor lock issues as it can use any vendor-specific APIs or services except where Kubernetes provides an abstraction, e.g., load balancer and storage.
- It enables applications that need to be released and updated without any downtime.

14) What does the node status contain in Kubernetes?

In Kubernetes, the main components of a node status are Address, Condition, Capacity, and Info.

↑ SCROLL TO TOP

Kubernetes Network Policy?

In Kubernetes, the Network Policy specifies how the same namespace's pods would communicate with each other and the network endpoint.

16) What process is used to run on the Kubernetes Master Node?

The Kube-api server process runs on the Kubernetes Master Node. It is used to scale the deployment of more instances.

17) Which types of systems use Kubernetes?

Kubernetes is the Linux kernel that is used for distributed systems. It provides a consistent interface for applications that use the shared pool of resources.

18) What do you understand by Kubernetes controller manager?

The Kubernetes controller manager is a daemon used for garbage collection, core control loops, and namespace creation. It also enables the running of more than one process on the master node.

19) What is the use of a namespace in Kubernetes?

↑ SCROLL TO TOP

In Kubernetes, Namespaces are used for dividing cluster resources between users. It is also helpful for more than one user to spread projects or teams and provide a scope of resources.

20) What is a pod in Kubernetes?

In Kubernetes, containers are not run directly so, pods are high-level structures used to wrap one or more containers to make them able to run. Containers in the same pod share a local network and the same resources. That's why they can easily communicate with other containers in the same pod. In this way, they were on the same machine while maintaining a degree of isolation.

21) What is the role of the Kube-scheduler in Kubernetes?

In Kubernetes, the Kube-scheduler are used to assign nodes to newly created pods.

22) What do you understand by a cluster of containers in Kubernetes?

A cluster of containers is nothing but a set of machine elements or nodes. Clusters specify specific routes so that the containers running on the nodes can communicate with each other. In Kubernetes, the container engine also provides hosting for the API server.

23) What are the different types of controller managers?

Following is the list of different types of controller managers:

- Endpoints controller
- Service accounts controller
- Namespace controller
- Node controller
- Replication controller
- Token controller

24) What is the Google Container Engine? What is its use?

↑ SCROLL TO TOP

The Google Container Engine is an open-source management platform tailor-made for Docker containers and clusters. It is used to provide support for the clusters that run in Google public cloud services.

25) What is the biggest disadvantage of Kubernetes?

The biggest disadvantages of Kubernetes are listed below:

- Security is the biggest issue with Kubernetes. It is not very secure.
- Kubernetes dashboard is not very user-friendly.
- Kubernetes is very complex to use. That's why it can reduce productivity.
- Kubernetes is costlier than its alternatives.

26) What is kubelet in Kubernetes?

In Kubernetes, the kubelet is a service agent used to control and maintain a group of pods by checking pod specifications. The kubelet runs on each node and makes them able to communicate between a master node and a slave node.

27) What do you understand by the node port service?

The node port service is a way to attain external traffic to your service. It is used to open a particular port on all nodes and forward the network traffic to this port.

↑ SCROLL TO TOP

28) What is Cluster IP in Kubernetes?

In Kubernetes, the cluster IP is a service inside the cluster that enables other apps to be accessed within the cluster.

29) What are the various services available in Kubernetes?

Following is the list of various services available in Kubernetes:

- Cluster IP service
- Load Balancer service
- Node Port service
- External Name Creation service etc.

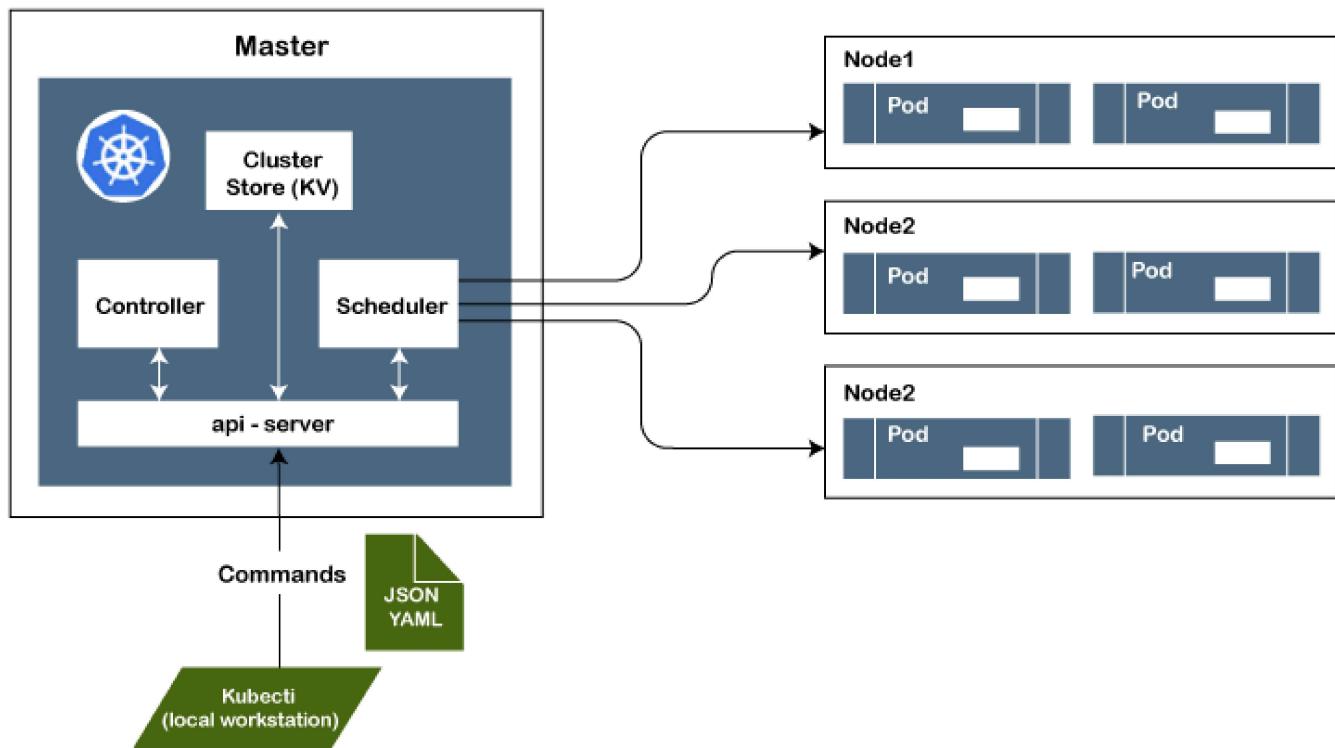
30) What do you understand by Ingress Network?

Ingress network is a set of rules which allow permission for connections into the Kubernetes cluster.

31) Explain the architecture of Kubernetes with a diagram?

Architectural Diagram:

↑ SCROLL TO TOP



Explanation of the critical components used in the diagram:

Master Node: The master node is the first and most crucial component of the Kubernetes' architecture. It is used to manage the Kubernetes cluster. It is the entry point for all kinds of administrative tasks.

There may be more than one master node in the cluster to check for fault tolerance.

API Server: The API server is an entry point for all the REST commands. It is used to control the cluster.

Scheduler: The scheduler is used to schedule the tasks to the slave node and distribute the workload. It also stores the resource usage information for every slave node.

Etcd: The etcd component is used to store configuration detail and good values. It communicates with the most component and receives commands and work. It is also responsible for managing network rules and port forwarding activity.

Worker/Slave nodes: Worker nodes or Slave nodes are another essential components that contain all the required services to manage the networking between the containers, communicate with the master node, which allows you to assign resources to the scheduled boxes.

↳ Pod's configuration from the API server and ensures that the described
 ↑ SCROLL TO TOP ↳ running.

Docker Container: The Docker container runs on each of the worker nodes, which runs the configured pods.

Pods: A pod is specified as a combination of single or multiple containers that logically run together on nodes.

32) What do you understand by Kube-proxy in Kubernetes?

In Kubernetes, Kube-proxy is an implementation of both a network proxy and a load balancer. It is used to support service abstraction with other networking operations and responsible for directing traffic to the container depending on IP and the port number.

33) What do you understand by Kubectl in Kubernetes? What is it used for?

In Kubernetes, Kubectl is software that is used to control Kubernetes clusters. The 'ctl' in "Kubectl" stands for control, a command-line interface to pass the command to the cluster and manage the Kubernetes component.

34) What is the full form of GKE, and what is its use?

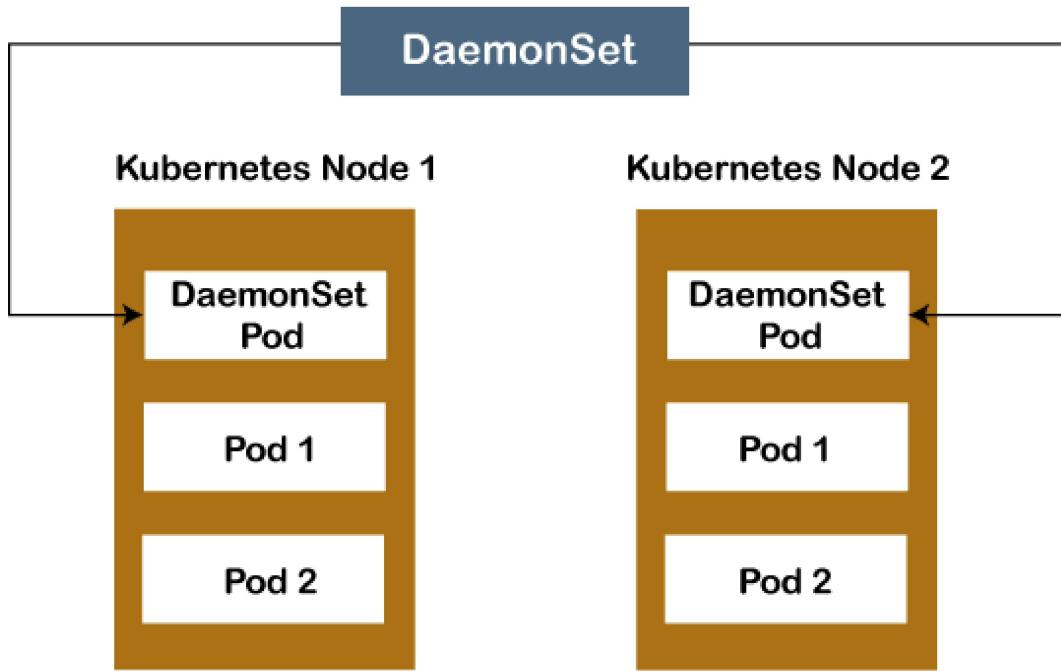
The full form of GKE is Google Container Engine. It is a management platform that supports clusters and Docker containers that run within the public cloud services of Google.

35) What is the difference between a daemon set, a deployment, and a replication controller?

Following is the difference between a daemon set, a deployment, and a replication controller:

Daemon set: A daemon set is used to ensure that all nodes you have selected are running precisely one copy of a pod.

↑ SCROLL TO TOP



Deployment: A deployment is a resource object in Kubernetes. It is used to provide declarative updates to applications. It also manages the scheduling and lifecycle of pods. It offers many key features for managing pods, such as pod health checks, rolling updates of pods, the ability to roll back and quickly scale pods horizontally.

Replication Controller: The replication controller is used to specify how many exact copies of a pod should be running in a cluster. It differs from a deployment in that it does not offer pod health checks, and the rolling update process is not as robust.

36) What is the use of load balancer in Kubernetes?

A load balancer is used in Kubernetes to provide a standard way to distribute network traffic among different services, which runs in the backend.

37) How can we run Kubernetes locally?

We can use the Minikube tool to run Kubernetes locally. It runs a single-node cluster in a VM (virtual machine) on the computer. So, it is best for users who have just started learning Kubernetes.

↳ [What is a container, and what is its use?](#)

↑ SCROLL TO TOP

A sidecar container is a utility container used to support the main container in a Pod. We can pair a sidecar container with one or more main containers and enhance those primary containers' functionality. It is mainly used for monitoring or when the system logs.

39) What do you understand by a headless service?

A headless service is a specific type of service which uses an IP address, but instead of load balancing, it returns associated pods.

40) What are the main objects used in Kubernetes?

Following is the list of objects used in Kubernetes:

- Pods
- Replication sets and controllers
- Jobs and cron jobs
- Daemon sets
- Distinctive identities
- Deployments
- Stateful sets.

41) What are the different types of pods in Kubernetes?

↑ SCROLL TO TOP

There are mainly two types of pods in Kubernetes:

- **Single Container Pods:** We can create Single Container Pods by using the run command in Kubernetes.
- **multi-container pods:** We can create multi-container pods by using the "create" command in Kubernetes.

42) What do you understand by Prometheus in Kubernetes?

Prometheus is an application that is used for the monitoring and alerting process. We can call out Prometheus to your systems, grab real-time metrics, compress it, and stores them properly in a database.

43) What is the difference between a replica set and a replication controller?

A Replica set and a Replication Controller both are used to do almost the same thing. Both of them ensure that a specified number of pod replicas are running at any given time.

The usage of selectors can distinguish the difference between them to replicate pods. The Replica Set uses Set-Based selectors while replication controllers use Equity-Based selectors.

44) What is the difference between the Equity-Based Selectors and the Selector-Based Selectors?

See the differences between the Equity-Based Selectors and the Selector-Based Selectors:

Equity-Based Selectors: Equity-Based Selectors are the type of selector that allows filtering by label key and values. This selector will only look for the pods, which will have the same phrase as that of the label. For example, if your label key says app=nginx, then, with this selector, you can only look for those pods with label app equal to Nginx.

Selector-Based Selectors: The Selector-Based Selectors are used to allow filtering keys according to a set of values. You can say that these selectors look for pods whose label has been mentioned in the set. For example, if your label key says app in (Nginx, NPS, Apache). With this selector, if your app is equal to any of Nginx, NPS, or Apache, then the selector will take it as a true result.

↑ SCROLL TO TOP

45) What are some examples of recommended security measures for Kubernetes?

Following are some examples of recommended security measures for Kubernetes:

- Defining resource quota
- Support for auditing
- Restriction of etcd access
- Regular security updates to the environment
- Network segmentation
- Definition of strict resource policies
- Continuous scanning for security vulnerabilities
- Using images from authorized repositories etc.

46) How can we get a static IP for a Kubernetes load balancer?

We can achieve a static IP for the Kubernetes load balancer by changing DNS records because the Kubernetes Master can assign a new static IP address.

47) What is the role of kube-apiserver and kube-scheduler?

The kube - apiserver is the master node control panel's front-end and follows the scale-out architecture. This is used to expose all the APIs of the Kubernetes Master node components. It is responsible for establishing communication between Kubernetes Node and the Kubernetes master components.

The kube-scheduler is used to distribute and manage the workload on the worker nodes. It chooses the most suitable node to run the unscheduled pod based on resource requirement and keeps track of resource utilization. It makes sure that the workload is not scheduled on nodes that are already full.

48) What do you understand by minikube in Kubernetes?

↑ SCROLL TO TOP

Minikube is software that facilitates users to run Kubernetes. It runs on the single nodes within the VM on your computer. This tool is also used by programmers who are developing an application using Kubernetes.

49) Which are the most important Kubectl commands?

Following is the list of some important Kubectl commands:

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

50) What do you understand by the labels in Kubernetes?

Labels are a set of keys that contain some values. The key values are connected to pods, replication controllers, and associated services. Generally, labels are added to some object during its creation time. We can easily modify them at the run time.

51) What are the main objectives of the replication controller?

Following are the main objectives of the replication controller:

- A replication controller is mainly responsible for controlling and administering the pod lifecycle.
- It is used to monitor and verify whether the allowed numbers of replicas are running or not.
- It also helps the users to check the pod status.
- It enables to alter a pod. The user can drag its position the way interested in it.

↑ SCROLL TO TOP

52) What do you understand by Sematext Docker Agent?

The Sematext Docker agent is a log collection agent with events and metrics, which runs as a small container in each Docker host. These agents are responsible for gathering metrics, events, and logs for all cluster nodes and containers.

53) What do you understand by persistent volume in Kubernetes?

In Kubernetes, a persistent volume is a storage unit that the administrator controls. It is used to manage an individual pod in a cluster.

54) Do all of the nodes occupy the same size in the cluster?

No, all the nodes don't occupy the same size in the cluster. The Kubernetes components, such as kubelet, take up resources on your nodes, and you still need more capacity for the node to do any task. In the larger cluster, it is always preferred to create a mix of different instance sizes. So, the Kubernetes can easily schedule the pods that require a lot of memory with intensive compute workloads on large nodes, and smaller nodes can handle smaller pods.

55) What is the ContainerCreating pod in Kubernetes?

A ContainerCreating pod is a specific type of pod scheduled on a node but can't start up properly.

56) What are the different types of Kubernetes Volume?

Following are the different types of Kubernetes Volume:

- EmptyDir
- GCE persistent disk
- Flocker
- HostPath
- NFS
- ISCSI

↑ SCROLL TO TOP

- o PersistentVolumeClaim
- o downwardAPI

57) What are the Secrets in Kubernetes?

Secrets are the objects in Kubernetes that stores sensitive information like username and password after performing encryption.

58) What do you understand by OpenShift in Kubernetes?

OpenShift is a public cloud application development and hosting platform developed by Red Hat. It is used to provide automation for management so that developers can easily write codes.

59) What do you understand by K8s in Kubernetes?

K8s (K-eight characters-S) is a term used for Kubernetes. It is an open-source orchestration framework used for the containerized applications.

60) What do you understand by federated clusters?

Federated clusters are a set of multiple clusters that are managed as a single cluster.

61) What is the difference between Kubernetes Volumes and Docker volumes?

The main differences between Kubernetes Volumes and Docker volumes are:

Kubernetes Volumes	Docker Volumes
The Kubernetes volumes are not limited to any container.	The Docker volumes are limited to a pod in the container.
Kubernetes volumes support all containers deployed in a pod of Kubernetes.	Docker volumes do not support all containers deployed in Docker.

↑ SCROLL TO TOP

62) What are the ways to provide API-Security on Kubernetes?

There are the following ways to provide API-Security on Kubernetes:

- Using correct auth mode with API server authentication mode.
- Making kubeless that protects its API via authorization-mode.
- Ensuring the kube-dashboard uses a restrictive RBAC (Role-Based Access Control) policy.

63) What do you understand by PVC? What is its full form?

The full form of PVC is Persistent Volume Claim. It is storage requested by Kubernetes for pods. In PVC, it doesn't require knowing the underlying provisioning. You can create the claim in the same namespace where the pod is created.

64) What do you understand by container resource monitoring?

The users need to understand the application's performance and resource utilization at all the different abstraction layers. Kubernetes create abstraction at different levels like containers, pods, services, and whole cluster to monitor them well. This process is called container resource monitoring.

Following is a list of some container resource monitoring tools:

- **Heapster:** It is used to gather data and events from the containers and pods within the cluster.
- **InfluxDB:** It is used along with heapster for visualizing data within the Kubernetes environment.
- **Grafana:** It is a time-series database used to store the data captured by all heapster pods.
- **CAdvisor:** It is a built-in tool in a kubelet that automatically discovers all the active containers and monitors them.
- **Prometheus:** It specifies a project of CNCF, which provides powerful querying, alerting, and visualization capabilities.

Company Interview Questions & Procedure	JavaScript Interview Questions
Java Basics Interview Questions	Java OOPs Interview Questions
Servlet Interview Questions	JSP Interview Questions
Spring Interview Questions	Hibernate Interview Questions
PL/SQL Interview Questions	SQL Interview Questions
Oracle Interview Questions	Android Interview Questions
jQuery Interview Questions	MySQL Interview Questions

You may also like:

- [Java Interview Questions](#)
- [SQL Interview Questions](#)
- [Python Interview Questions](#)
- [JavaScript Interview Questions](#)
- [Angular Interview Questions](#)
- [Selenium Interview Questions](#)

[↑ SCROLL TO TOP](#) [view Questions](#)

- HR Interview Questions
- C Programming Interview Questions
- C++ Interview Questions
- Data Structure Interview Questions
- DBMS Interview Questions
- HTML Interview Questions
- IAS Interview Questions
- Manual Testing Interview Questions
- OOPs Interview Questions
- .Net Interview Questions
- C# Interview Questions
- ReactJS Interview Questions
- Networking Interview Questions
- PHP Interview Questions
- CSS Interview Questions
- Node.js Interview Questions
- Spring Interview Questions
- Hibernate Interview Questions
- AWS Interview Questions
- Accounting Interview Questions

Learn Latest Tutorials



Splunk



SPSS



Swagger



Transact-SQL



Tumblr



ReactJS



Regex



Reinforcement Learning

↑ SCROLL TO TOP



R Programming



RxJS



React Native



Python Design Patterns



Python Pillow



Python Turtle



Keras

Preparation



Aptitude



Logical Reasoning



Verbal Ability



Interview Questions



Company Interview Questions

Company Questions

Trending Technologies



Artificial Intelligence



AWS



Selenium tutorial



Cloud Computing tutorial



Hadoop



ReactJS



Data Science



Angular 7 Tutorial

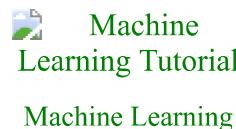
[↑ SCROLL TO TOP](#)



Blockchain
Tutorial
Blockchain



Git
Tutorial



Machine
Learning
Tutorial
Machine Learning

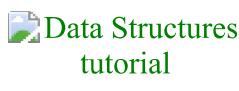


DevOps
Tutorial
DevOps

B.Tech / MCA



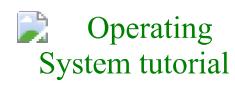
DBMS



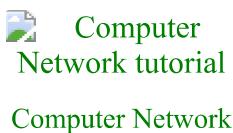
Data Structures
tutorial



DAA



Operating
System tutorial
Operating System



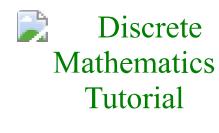
Computer
Network
tutorial
Computer Network



Compiler
Design
tutorial
Compiler Design



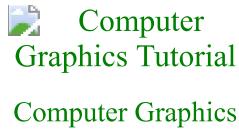
Computer
Organization and
Architecture
Computer
Organization



Discrete
Mathematics
Tutorial
Discrete
Mathematics



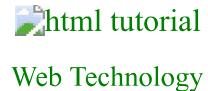
Ethical Hacking
Tutorial
Ethical Hacking



Computer
Graphics
Tutorial
Computer Graphics



Software
Engineering
Tutorial
Software
Engineering



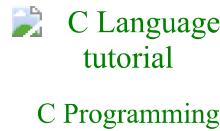
Web Technology



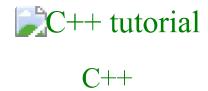
Cyber Security



Automata
Tutorial
Automata



C Language
tutorial
C Programming



C++



Java



.Net
Framework
tutorial
.Net



Python



List of
Programs
Programs



Control
Systems



Data Mining
Tutorial
Data Mining



Data
Warehouse
Tutorial

↑ SCROLL TO TOP

Data Warehouse

↑ SCROLL TO TOP