Top 55 Kubernetes Interview Questions and Answers (2023)

By

David Taylor

Updated

May 13, 2023

Here are Kubernetes Interview Questions for fresher as well as experienced candidates to get the dream job.

1) What is Kubernetes?

Kubernetes is a container management system developed in the Google platform. The purpose of Kubernetes is to manage a containerized application in various types of physical, virtual, and cloud environments. Google Kubernetes is a highly flexible container tool to deliver even complex applications, consistently. Applications run on clusters of hundreds to thousands of individual servers.

2) Define node in Kubernetes

A node the smallest unit of hardware. It defines a single machine in a cluster that can be a virtual machine from a cloud provider or physical machine in the data center. Every machine available in the Kubernetes cluster can substitute other machines.

Kubernetes

3) What is the work of a kube-scheduler?

Kube-scheduler is the default scheduler for Kubernetes. It assigns nodes to newly created pods.

4) Define daemon sets

Daemon sets are a set of pods that runs on a host. They are used for host layers attributes like monitoring network or simple network.

5) Define Heapster in Kubernetes

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

6) What tasks are performed by Kubernetes?

Kubernetes is the Linux kernel which is used for distributed systems. It helps you to be abstract the underlying hardware of the nodes(servers) and offers a consistent interface for applications that consume the shared pool of resources.

7) Define Kubernetes controller manager

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

8) Why use namespace in Kubernetes?

Namespaces in Kubernetes are used for dividing cluster resources between users. It helps the environment where more than one user spread projects or teams and provides a scope of resources.

9) Why use Kubernetes?

Kubernetes is used because:

Kubernetes can run on-premises bare metal, OpenStack, public clouds Google, Azure, AWS, etc.

It helps you to 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 will enable applications that need to be released and updated without any downtime.

Kubernetes allows you to assure those containerized apps run where and when you want and help you to find resources and tools which you want to work.

10) What are the features of Kubernetes?

The features of Kubernetes are:

Automated Scheduling

Self-Healing Capabilities

Automated rollouts & rollback

Horizontal Scaling & Load Balancing

Offers environment consistency for development, testing, and production

Infrastructure is loosely coupled to each component can act as a separate unit

Provides a higher density of resource utilization

Offers enterprise-ready features

Application-centric management

Auto-scalable infrastructure

You can create predictable infrastructure

11) Mention the types of controller managers

Types of controller managers are: 1) endpoints controller, 2) service accounts controller, 3) node controller, 4) namespace controller, 5) replication controller, 6) token controller.

12) Explain Kubernetes Architecture

Kubernetes Architecture Diagram

Master Node: The master node is the first and most vital component which is responsible for the management of 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 acts as an entry point for all the REST commands used for controlling the cluster.

Scheduler: The scheduler schedules the tasks to the slave node. It stores the resource usage information for every slave node. It is responsible for distributing the workload.

Etcd: etcd components, store configuration detail, and wright values. It communicates with the most component to receive commands and work. It also manages network rules and port forwarding activity.

Worker/Slave nodes: Worker nodes are another essential component that contains 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 containers.

Kubelet: It gets the configuration of a Pod from the API server and ensures that the described containers are up and running.

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

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

13) List various services available in Kubernetes

Various services available in Kubernetes are 1) Cluster IP service, 2) Load Balancer service, 3) Node Port service, 4) External Name Creation service.

14) Define Cluster IP

The Cluster IP is a Kubernetes service that offers a service inside the cluster that other apps inside cluster can access.

15) Explain node port

The node port service is a fundamental way to get external traffic to your service. It opens a particular port on all nodes and forwards network traffic sent to this port.

16) Define kubelet

The kubelet is a service agent which controls and maintains group pf pods by checking pod specification using Kubernetes. The kubelet runs on each node and allows to communicate between a master node and a slave node.

17) What are the disadvantages of Kubernetes?

Kubernetes dashboard is not as helpful as it should be

Security is not very effective.

It is very complex and can reduce productivity

Kubernetes is more costly than its alternatives.

18) What is Kube-proxy?

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

19) What is the difference between Kubernetes and Docker Swarm?

The difference between Kubernetes and Docker Swarm is:

Kubernetes	Docker Swarm
Kubernetes Provides an auto-scaling feature.	Docker Swarm does not provide an auto-scaling feature.
Manually configure your load balancing settings.	Does auto load balancing
Installation is complicated & time-consuming.	Installation is easy & fast.
GUI is available.	GUI not available.
It provides a built-in load balancing technique.	Process scheduling is done to maintain services while updating.

20) Define Ingress Network

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

21) What is Kubectl used for?

Kubectl is a software for controlling Kubernetes clusters. Ctl stands for control, which is a command-line interface to pass the command to the cluster and manage the Kubernetes component.

22) What is GKE?

GKE or Google Container Engine is a management platform that supports clusters and Docker containers that run within public cloud services of Google.

23) Why load balancer is needed?

A load balancer is needed because it gives a standard way to distribute network traffic among different services, which runs in the backend.

24) How to run Kubernetes locally?

Kubernetes can be run locally using the Minikube tool. It runs a single-node cluster in a VM (virtual machine) on the computer. Therefore, it offers the ideal way for users who have just started learning Kubernetes.

25) What are the tools that are used for container monitoring?

Tools that are used for container monitoring are:

Heapster

cAdvisor

Prometheus

InfluxDB

Grafana

26) List components of Kubernetes

There are three components of Kubernetes, they are:

Addons

Node components

Master Components

27) Define headless service

Headless service is defined as a service that uses IP address, but instead of load balancing, it returns of associated pods.

28) What are the important components of node status?

The important component of node status are:

Condition

Capacity

Info

Address

29) What is minikube?

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

30) Mention the uses of GKE

The uses of the GKE (Google Kubernetes Engine) are:

It can be used to create docker container clusters

Resize application controllers

Update and then upgrade the clusters of container

Debug cluster of the container.

GKE can be used to creates a replication controller, jobs, services, container pods, or load balancer.

31) Define orchestration in Kubernetes

Orchestration in Kubernetes defines as an automatic method of scheduling the work of every container. It is used for applications that are based on microservices within clusters.

32) Explain Prometheus in Kubernetes

Prometheus is an application that is used for monitoring and alerting. It can be called out to your systems, grab real-time metrics, compress it, and stores properly in a database.

33) List tools for container orchestration

The tools for container orchestration are 1) Docker swarm, 2) Apache Mesos, and 3) Kubernetes.

34) Mention the list of objects of Kubernetes?

Objects that are used in Kubernetes are: 1) Pods, 2) Replication sets and controllers, 3) Jobs and cron jobs, 4) Daemon sets, 5) Distinctive identities, 6) Deployments, 7) and Stateful sets.

35) Define Stateful sets in Kubernetes

The stateful set is a workload API object that is used to manage the stateful application. It can also be used to manage the deployments and scaling the sets of pods. The state information and other data of stateful pods are store in the disk storage, which connects with stateful set.

36) Why use Daemon sets?

Daemon sets are used because:

It enables to runs storage platforms like ceph and glusterd on each node.

Daemon sets run the logs collection on every node such as filebeat or fluentd.

It performs node monitoring on each and every node.

37) Explain Replica set

A Replica set is used to keep replica pods stable. It enables us to specify the available number of identical pods. This can be considered a replacement for the replication .controller.

38) List out some important Kubectl commands:

The important Kubectl commands are:

kubectl annotate

kubectl cluster-info

kubectl attach

kubectl apply

kubectl config

kubectl autoscale

kubectl config current-context

kubectl config set.

39) Why uses Kube-apiserver?

Kube-apiserver is an API server of Kubernetes that is used to configure and validate API objects, which include services, controllers, etc. It provides the frontend to the cluster's shared region using which components interact with each other.

40) Explain the types of Kubernetes pods

There are two types of pods in Kubernetes:

Single Container Pod: It can be created with the run command.

Multicontainer pods: It can be created using the "create" command in Kubernetes.

41) What are the labels in Kubernetes?

Labels are a collection 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. They can be modified by the users at run time.

42) What are the objectives of the replication controller?

The objectives of the replication controller are:

It is responsible for controlling and administering the pod lifecycle.

It monitors and verifies whether the allowed number of replicas are running or not.

The replication controller helps the user to check the pod status.

It enables to alter a pod. The user can drag its position the way interested in it.

43) What do you mean by persistent volume?

A persistent volume is a storage unit that is controlled by the administrator. It is used to manage an individual pod in a cluster.

44) What are Secrets in Kubernetes?

Secrets are sensitive information like login credentials of the user. They are objects in Kubernetes that stores sensitive information like username and password after performing encryption.

45) What is Sematext Docker Agent?

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

46) Define OpenShift

OpenShift is a public cloud application development and hosting platform developed by Red Hat. It offers automation for management so that developers can focus on writing the code.

47) Define K8s

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

48) What are federated clusters?

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

49) Mention the difference between Docker volumes and Kubernetes Volumes

Kubernetes Volumes	Docker Volumes
Volumes are not limited to any container.	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.

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

The ways to provide API-Security on Kubernetes are:

Using correct auth mode with API server authentication mode= Node.

Make kubeless that protects its API via authorization-mode=Webhook.

Ensure the kube-dashboard uses a restrictive RBAC (Role-Based Access Control) policy

51) What is ContainerCreating pod?

A ContainerCreating pod is one that can be scheduled on a node but can’t start up properly.

52) What are the types of Kubernetes Volume?

The types of Kubernetes Volume are:

EmptyDir

GCE persistent disk

Flocker

HostPath

NFS

ISCSI

rbd

PersistentVolumeClaim

downwardAPI

53) Explain PVC

The full form of PVC stands for Persistent Volume Claim. It is storage requested by Kubernetes for pods. The user does not require to know the underlying provisioning. This claim should be created in the same namespace where the pod is created.

54) What is the Kubernetes Network Policy?

Network Policy defines how the pods in the same namespace would communicate with each other and the network endpoint.

55) What is Kubernetes proxy service?

Kubernetes proxy service is a service which runs on the node and helps in making it available to an external host.

These interview questions will also help in your viva(orals)

Basic Kubernetes Interview Questions

1. How to do maintenance activity on the K8 node?

Whenever there are security patches available the Kubernetes administrator has to perform the maintenance task to apply the security patch to the running container in order to prevent it from vulnerability, which is often an unavoidable part of the administration. The following two commands are useful to safely drain the K8s node.

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

The first command moves the node to maintenance mode or makes the node unavailable, followed by kubectl drain which will finally discard the pod from the node. After the drain command is a success you can perform maintenance.

Note: If you wish to perform maintenance on a single pod following two commands can be issued in order:

kubectl get nodes: to list all the nodes

kubectl drain <node name>: drain a particular node

2. How do we control the resource usage of POD?

With the use of limit and request resource usage of a POD can be controlled.

Request: The number of resources being requested for a container. If a container exceeds its request for resources, it can be throttled back down to its request.

Limit: An upper cap on the resources a single container can use. If it tries to exceed this predefined limit it can be terminated if K8's decides that another container needs these resources. If you are sensitive towards pod restarts, it makes sense to have the sum of all container resource limits equal to or less than the total resource capacity for your cluster.

Example:

```
apiVersion: v1
kind: Pod
metadata:
  name: demo
spec:
  containers:
  - name: example1
  image:example/example1
  resources:
      requests:
          memory: "_Mi"
          cpu: "_m"
      limits:
          memory: "_Mi"
          cpu: "_m"
```

3. What are the various K8's services running on nodes and describe the role of each service?

Mainly K8 cluster consists of two types of nodes, executor and master.

Executor node: (This runs on master node)

Kube-proxy: This service is responsible for the communication of pods within the cluster and to the outside network, which runs on every node. This service is responsible to maintain network protocols when your pod establishes a network communication.

kubelet: Each node has a running kubelet service that updates the running node accordingly with the configuration(YAML or JSON) file. NOTE: kubelet service is only for containers created by Kubernetes.

Master services:

Kube-apiserver: Master API service which acts as an entry point to K8 cluster.

Kube-scheduler: Schedule PODs according to available resources on executor nodes.

Kube-controller-manager: is a control loop that watches the shared state of the cluster through the apiserver and makes changes attempting to move the current state towards the desired stable state

You can download a PDF version of Kubernetes Interview Questions.

Download PDF

4. What is PDB (Pod Disruption Budget)?

A Kubernetes administrator can create a deployment of a kind: PodDisruptionBudget for high availability of the application, it makes sure that the minimum number is running pods are respected as mentioned by the attribute minAvailable spec file. This is useful while performing a drain where the drain will halt until the PDB is respected to ensure the High Availability(HA) of the application. The following spec file also shows minAvailable as 2 which implies the minimum number of an available pod (even after the election).

Example: YAML Config using minAvailable =>

```
apiVersion: policy/v1beta1

kind: PodDisruptionBudget

metadata:
  name: zk-pdb
spec:
  minAvailable: 2
  selector:
    matchLabels:
      app: zookeeper
```

5. What’s the init container and when it can be used?

init containers will set a stage for you before running the actual POD.

Wait for some time before starting the app Container with a command like sleep 60.

Clone a git repository into a volume.

6. What is the role of Load Balance in Kubernetes?

Load balancing is a way to distribute the incoming traffic into multiple backend servers, which is useful to ensure the application available to the users.

Load Balancer

In Kubernetes, as shown in the above figure all the incoming traffic lands to a single IP address on the load balancer which is a way to expose your service to outside the internet which routes the incoming traffic to a particular pod (via service) using an algorithm known as round-robin. Even if any pod goes down load balances are notified so that the traffic is not routed to that particular unavailable node. Thus load balancers in Kubernetes are responsible for distributing a set of tasks (incoming traffic) to the pods

7. What are the various things that can be done to increase Kubernetes security?

By default, POD can communicate with any other POD, we can set up network policies to limit this communication between the PODs.

RBAC (Role-based access control) to narrow down the permissions.

Use namespaces to establish security boundaries.

Set the admission control policies to avoid running the privileged containers.

Turn on audit logging.

8. How to monitor the Kubernetes cluster?

Prometheus is used for Kubernetes monitoring. The Prometheus ecosystem consists of multiple components.

Mainly Prometheus server which scrapes and stores time-series data.

Client libraries for instrumenting application code.

Push gateway for supporting short-lived jobs.

Special-purpose exporters for services like StatsD, HAProxy, Graphite, etc.

An alert manager to handle alerts on various support tools.

9. How to get the central logs from POD?

This architecture depends upon the application and many other factors. Following are the common logging patterns

Node level logging agent.

Streaming sidecar container.

Sidecar container with the logging agent.

Export logs directly from the application.

In the setup, journalbeat and filebeat are running as daemonset. Logs collected by these are dumped to the kafka topic which is eventually dumped to the ELK stack.

The same can be achieved using EFK stack and fluentd-bit.

Intermediate Interview Questions

10. How to turn the service defined below in the spec into an external one?

```
spec:
    selector:
        app: some-app
    ports:
        - protocol: UDP
            port: 8080
            targetPort: 8080
```

Explanation -

Adding type: LoadBalancer and nodePort as follows:

```
spec:
  selector:
      app: some-app
  type: LoadBalancer
  ports:
      - protocol: UDP
          port: 8080
          targetPort: 8080
          nodePort: 32412
```

11. Complete the following configurationspec file to make it Ingress

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

```
spec:
```

Explanation -

One of the several ways to answer this question.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: someapp-ingress
spec:
  rules:
  - host: my.host
      http:
          paths:
          - backend:
                  serviceName: someapp-internal-service
                  servicePort: 8080
```

12. How should TLS be configured with Ingress?

Add tls and secretName entries.

```
spec:
  tls:
  - hosts:
      - some_app.com
      secretName: someapp-secret-tls
```

13. Why should namespaces be used? How does using the default namespace cause problems?

Over the course of time, using the default namespace alone is proving to be difficult, since you are unable to get a good overview of all the applications you can manage within the cluster as a whole. The namespaces allow applications to be organized into groups that make sense, such as a namespace for all monitoring applications and another for all security applications.

Additionally, namespaces can be used for managing Blue/Green environments, in which each namespace contains its own version of an app as well as sharing resources with other namespaces (such as logging or monitoring). It is also possible to have one cluster with multiple teams using namespaces. The use of the same cluster by multiple teams may lead to conflict. Suppose they end up creating an app that has the same name, this means that one team will override the app created by the other team as Kubernetes prohibits two apps with the same name (within the same namespace).

14. What service and namespace are referred to in the following file?

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: some-configmap

data:

  some_url: silicon.chip
```

It is clear from the above file that the service "silicon" is a reference to a namespace called "chip".

15. What is an Operator?

As an extension to K8, the operator provides the capability of managing applications and their components using custom resources. Operators generally comply with all the principles relating to Kubernetes, especially those relating to the control loops.

16. What is the purpose of operators?

As compared to stateless applications, achieving desired status changes and upgrades are handled the same way for every replica, managing Kubernetes applications is more challenging. The stateful nature of stateful applications may require different handling for upgrading each replica, as each replica might be in a different state. Therefore, managing stateful applications often requires a human operator. This is supposed to be assisted by Kubernetes Operator. Moreover, this will pave the way for a standard process to be automated across several Kubernetes clusters.

17. What is GKE?

GKE is Google Kubernetes Engine that is used for managing and orchestrating systems for Docker containers. With the help of Google Public Cloud, we can also orchestrate the container cluster.

18. What is Ingress Default Backend?

It specifies what to do with an incoming request to the Kubernetes cluster that isn't mapped to any backend i.e what to do when no rules being defined for the incoming HTTP request If the default backend service is not defined, it's recommended to define it so that users still see some kind of message instead of an unclear error.

Kubernetes Interview Questions For Experienced

19. How to run Kubernetes locally?

Kubernetes can be set up locally using the Minikube tool. It runs a single-node bunch in a VM on the computer. Therefore, it offers the perfect way for users who have just ongoing learning Kubernetes.

20. What is Kubernetes Load Balancing?

Load Balancing is one of the most common and standard ways of exposing the services. There are two types of load balancing in K8s and they are:

Internal load balancer – This type of balancer automatically balances loads and allocates the pods with the required incoming load.

External Load Balancer – This type of balancer directs the traffic from the external loads to backend pods.

21. What the following in the Deployment configuration file mean?

```
spec:
    containers:
        - name: USER_PASSWORD
            valueFrom:
                secretKeyRef:
                    name: some-secret
                    key: password
```

Explanation -

USER_PASSWORD environment variable will store the value from the password key in the secret called "some-secret" In other words, you reference a value from a Kubernetes Secret.

22. Can you explain the differences between Docker Swarm and Kubernetes?

Below are the main difference between Kubernetes and Docker:

The installation procedure of the K8s is very complicated but if it is once installed then the cluster is robust. On the other hand, the Docker swarm installation process is very simple but the cluster is not at all robust.

Kubernetes can process the auto-scaling but the Docker swarm cannot process the auto-scaling of the pods based on incoming load.

Kubernetes is a full-fledged Framework. Since it maintains the cluster states more consistently so autoscaling is not as fast as Docker Swarm.

23. How to troubleshoot if the POD is not getting scheduled?

In K8’s scheduler is responsible to spawn pods into nodes. There are many factors that can lead to unstartable POD. The most common one is running out of resources, use the commands like kubectl describe <POD> -n <Namespace> to see the reason why POD is not started. Also, keep an eye on kubectl to get events to see all events coming from the cluster.

24. How to run a POD on a particular node?

Various methods are available to achieve it.

nodeName: specify the name of a node in POD spec configuration, it will try to run the POD on a specific node.

nodeSelector: Assign a specific label to the node which has special resources and use the same label in POD spec so that POD will run only on that node.

nodeaffinities: required DuringSchedulingIgnoredDuringExecution, preferredDuringSchedulingIgnoredDuringExecution are hard and soft requirements for running the POD on specific nodes. This will be replacing nodeSelector in the future. It depends on the node labels.

25. What are the different ways to provide external network connectivity to K8?

By default, POD should be able to reach the external network but vice-versa we need to make some changes. Following options are available to connect with POD from the outer world.

Nodeport (it will expose one port on each node to communicate with it)

Load balancers (L4 layer of TCP/IP protocol)

Ingress (L7 layer of TCP/IP Protocol)

Another method is to use Kube-proxy which can expose a service with only cluster IP on the local system port.

```
$ kubectl proxy --port=8080 $ http://localhost:8080/api/v1/proxy/namespaces//services/:/
```

26. How can we forward the port '8080 (container) -> 8080 (service) -> 8080 (ingress) -> 80 (browser)and how it can be done?

The ingress is exposing port 80 externally for the browser to access, and connecting to a service that listens on 8080. The ingress will listen on port 80 by default. An "ingress controller" is a pod that receives external traffic and handles the ingress and is configured by an ingress resource For this you need to configure the ingress selector and if no 'ingress controller selector' is mentioned then no ingress controller will manage the ingress.

Simple ingress Config will look like

```
host: abc.org
http:
paths:
backend:
serviceName: abc-service
servicePort: 8080
```

Then the service will look like

```
kind: Service
apiVersion: v1
metadata:
name: abc-service
spec:
ports:
protocol: TCP
port: 8080 # port to which the service listens to
targetPort: 8080
```

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% Hike!

Master Most in Demand Skills Now !

Email Address

+91 IN INDIA

Phone Number

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:

A pod is created and the preferred state is mentioned, and without filling the node name, it is saved to etcd

The scheduler notices the new pod with no node bound

It finds a suitable node for that pod

It then informs the API server to bind the pod to the node, and next, the new desired state is saved to etcd

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

IntellipaatCloudArchitectMasterIntellipaat

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.

Start your journey as a Kubernetes Cluster Manager by enrolling in Kubernetes Cluster Management Certification.

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.

Certification in Cloud & Devops

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.

Kubernetes Architecture

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.