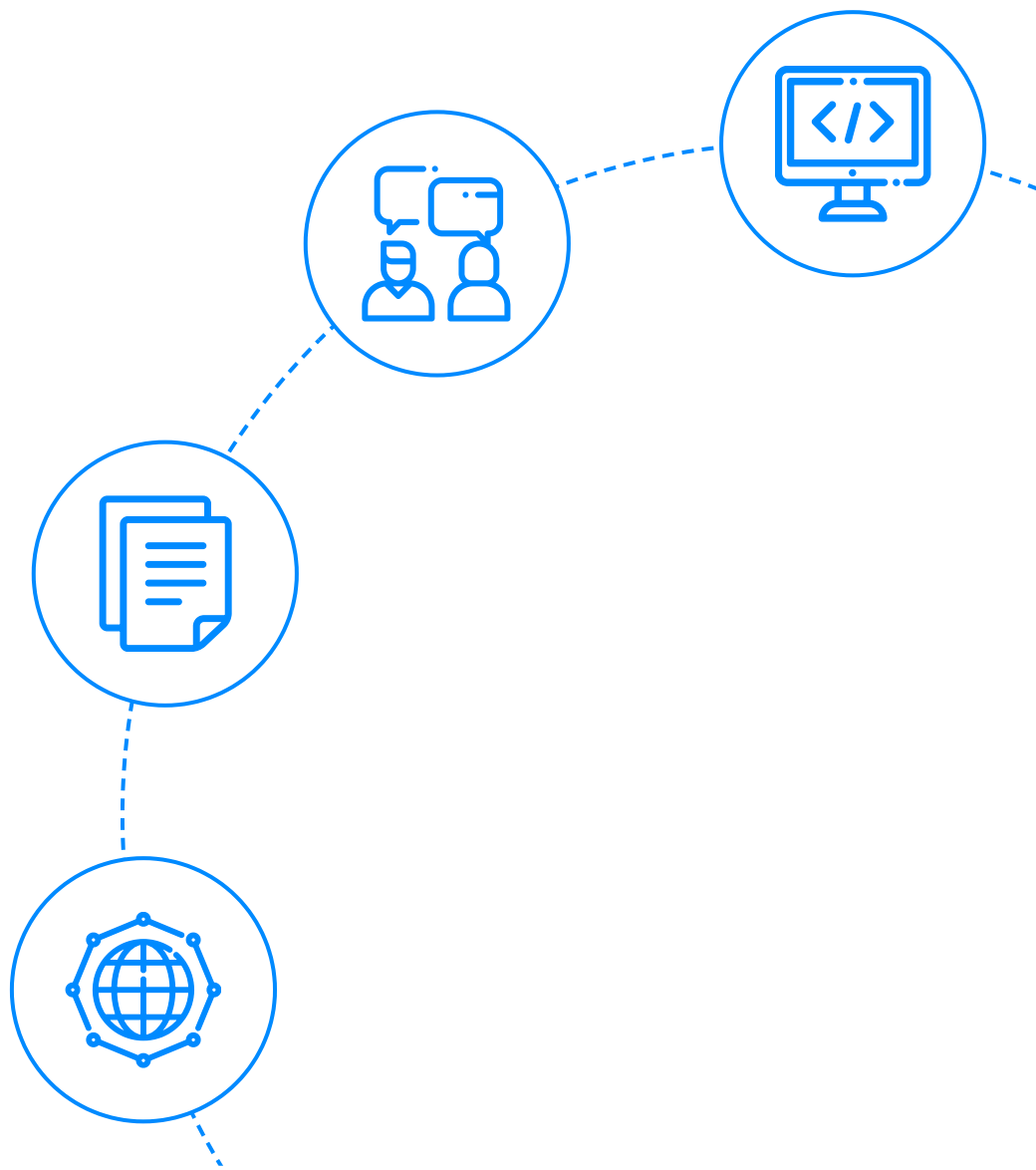




InterviewBit

Kubernetes Interview Questions



To view the live version of the page, [click here.](#)

© Copyright by Interviewbit

Contents

Basic Kubernetes Interview Questions

1. How to do maintenance activity on the K8 node?
2. How do we control the resource usage of POD?
3. What are the various K8's services running on nodes and describe the role of each service?
4. What is PDB (Pod Disruption Budget)?
5. What's the init container and when it can be used?
6. What is the role of Load Balance in Kubernetes?
7. What are the various things that can be done to increase Kubernetes security?
8. How to monitor the Kubernetes cluster?
9. How to get the central logs from POD?

Intermediate Interview Questions

10. How to turn the service defined below in the spec into an external one?
11. Complete the following configurationspec file to make it Ingress
12. How to configure TLS with Ingress?
13. Why use namespaces? What is the problem with using the default namespace?
14. In the following file which service and in which namespace is referred?
15. What is an Operator?
16. Why do we need Operators?
17. What is GKE?
18. What is Ingress Default Backend?

Kubernetes Interview Questions For Experienced

19. How to run Kubernetes locally?
20. What is Kubernetes Load Balancing?
21. What the following in the Deployment configuration file mean?
22. What is the difference between Docker Swarm and Kubernetes?
23. How to troubleshoot if the POD is not getting scheduled?
24. How to run a POD on a particular node?
25. What are the different ways to provide external network connectivity to K8?
26. How can we forward the port '8080 (container) -> 8080 (service) -> 8080 (ingress) -> 80 (browser) and how it can be done?

Let's get Started

What's Kubernetes?

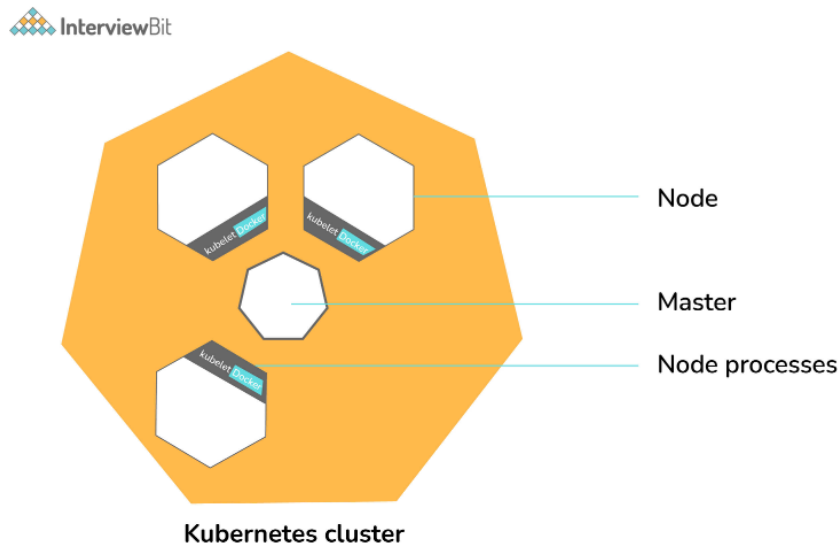
Kubernetes is a distributed open-source technology which helps us in scheduling and executing application container within and across clusters. A Kubernetes cluster consists of two types of resources:

The Master => Coordinates all activities in the cluster, for example, => scheduling applications, maintaining applications' state, scaling applications, and rolling out new updates

Nodes => A node is an instance of an OS that serves as a worker machine in a Kubernetes cluster.

Also, Node will have two components

- Kubelet => Agent for managing and communicating with the master
- Tool (Docker/containers) => Tools for running container operations



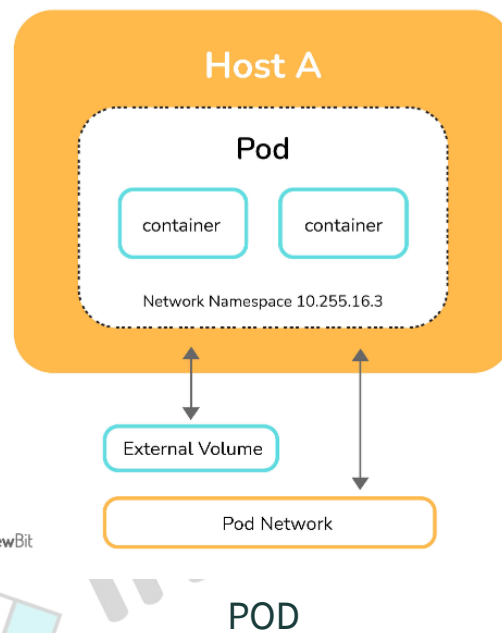
Kubernetes Cluster

It is designed based on ground-up as a loosely coupled collection of containers centered around deploying, maintaining, and scaling workloads. Works as an engine for resolving state by converging actual and the desired state of the system (self-healing). Hidden from the underlying hardware of the nodes and provides a uniform interface for workloads to be both deployed and consume the shared pool of resources(hardware) in order to simplify deployment.

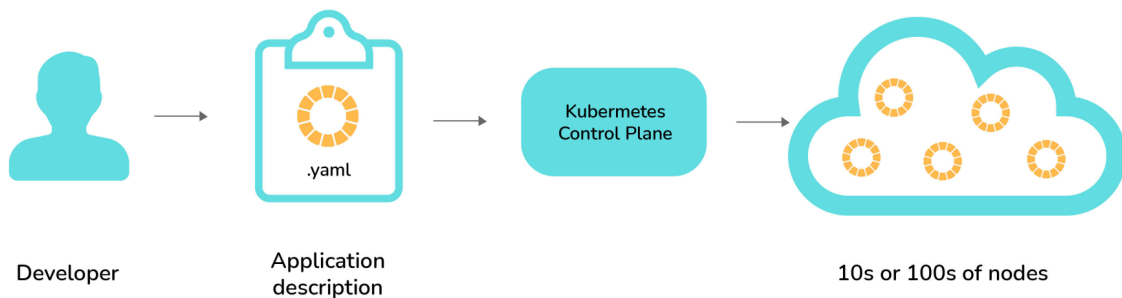
Pods are the smallest unit of objects that can be deployed on Kubernetes, Kubernetes packages one or more containers into a higher-level structure called a pod. Pod runs one level higher to the container.

A POD always runs on a Node but they share few resources which can be Shared Volumes, Cluster Unique IP, Info about how to run each container. All containers in the pod are going to be scheduled on an equivalent node.

Services are the unified way of accessing the workloads on the pods, Control plane which is the core of Kubernetes is an API server that lets you query, manipulates the state of an object in Kubernetes.



The following image describes the work-flow of the Kubernetes from a high level, wherein the application description is a YAML file also known as configuration or spec file with the help of which we can deploy applications bundled in the form of pods in cluster or node



Kubernetes Flow

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: "1Mi"
        cpu: "1m"
      limits:
        memory: "1Mi"
        cpu: "1m"
```

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

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?

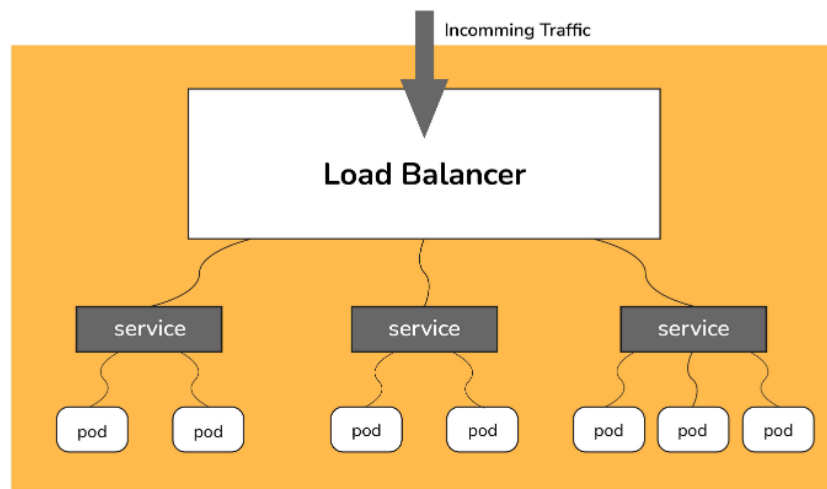
init containers will set a stage for you before running the actual POD.

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.



InterviewBit

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 balancers 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 to configure TLS with Ingress?

Add `tls` and `secretName` entries.

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

13. Why use namespaces? What is the problem with using the default namespace?

While using the default namespace alone, it becomes hard over time to get an overview of all the applications you can manage in your cluster. Namespaces make it easier to organize the applications into groups that make sense, like a namespace of all the monitoring applications and a namespace for all the security applications, etc.

Namespaces can also be useful for managing Blue/Green environments where each namespace can include a different version of an app and also share resources that are in other namespaces (namespaces like logging, monitoring, etc.).

Another use case for namespaces is one cluster with multiple teams. When multiple teams use the same cluster, they might end up stepping on each other's toes. For example, if they end up creating an app with the same name it means one of the teams overrides the app of the other team because there can't be two apps in Kubernetes with the same name (in the same namespace).

14. In the following file which service and in which namespace is referred?

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: some-configmap
data:
  some_url: silicon.chip
```

Answer - It's referencing the service "silicon" in the namespace called "chip".

15. What is an Operator?

"Operators are software extensions to K8s which make use of custom resources to manage applications and their components. Operators follow Kubernetes principles, notably the control loop."

16. Why do we need Operators?

The process of managing applications in Kubernetes isn't as straightforward as managing stateless applications, where reaching the desired status and upgrades are both handled the same way for every replica. In stateful applications, upgrading each replica might require different handling due to the stateful nature of the app, each replica might be in a different status. As a result, we often need a human operator to manage stateful applications. Kubernetes Operator is supposed to assist with this.

This will also help with automating a standard process on multiple 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. What is the difference 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.

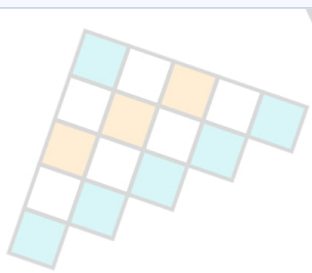
```
$ kubectl proxy --port=8080 $  
http://localhost:8080/api/v1/proxy/namespaces//services/:
```

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
```



Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)