## What's Kubernetes?

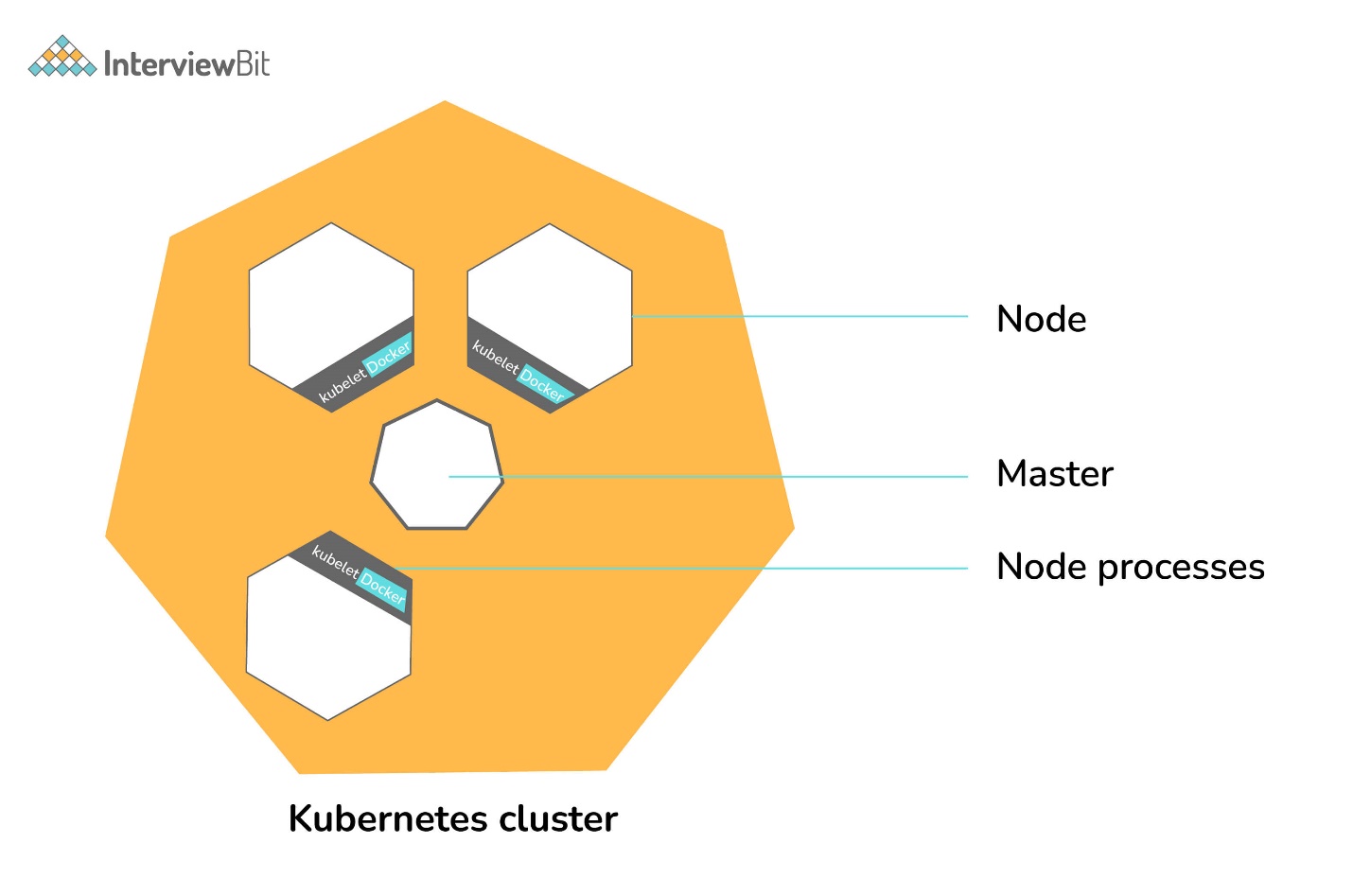
[Kubernetes](https://www.interviewbit.com/blog/kubernetes-architecture/) is a distributed open-source technology that helps us in scheduling and executing application containers within and across clusters. A Kubernetes cluster consists of two types of resources:

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

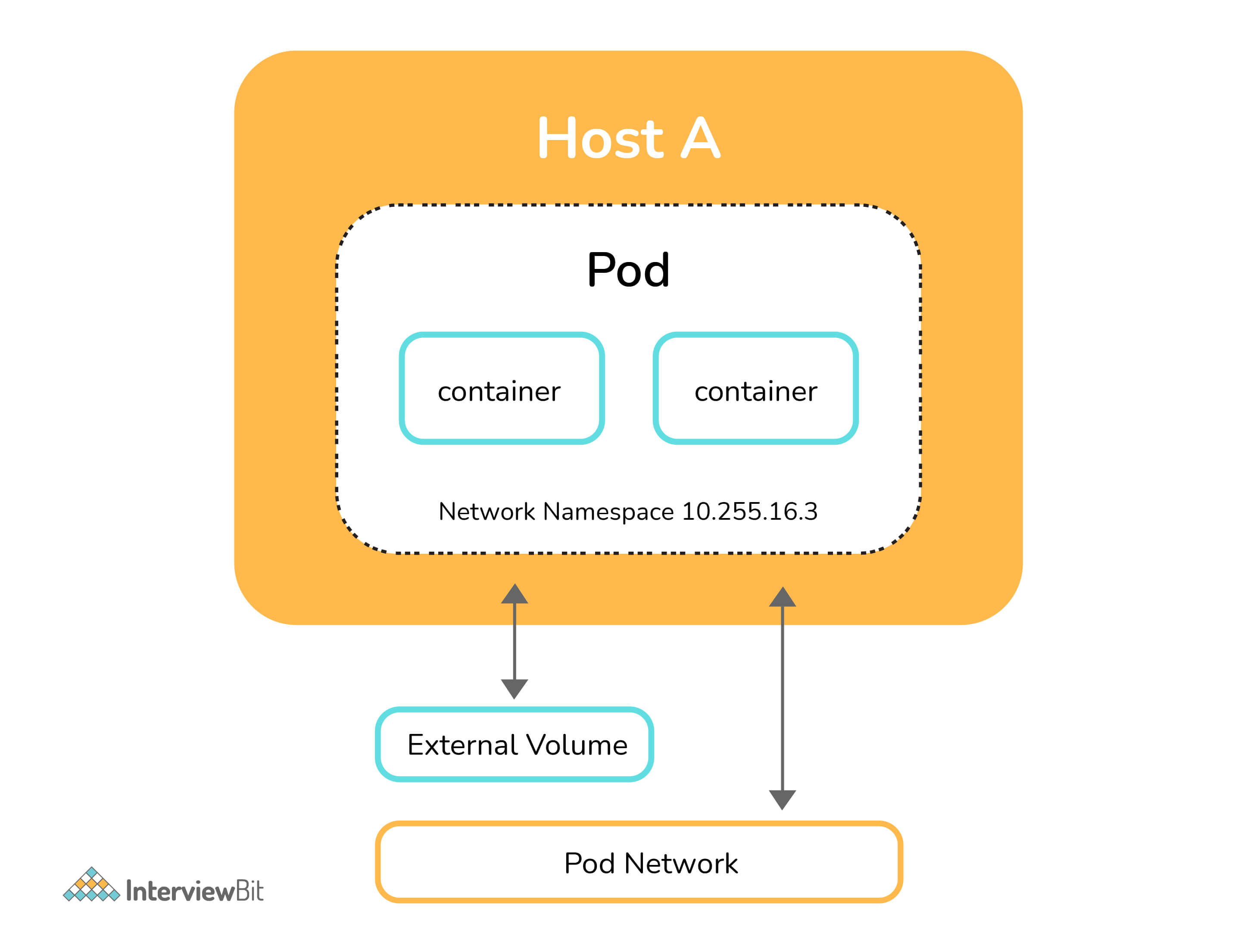
Kubernetes Cluster

 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.

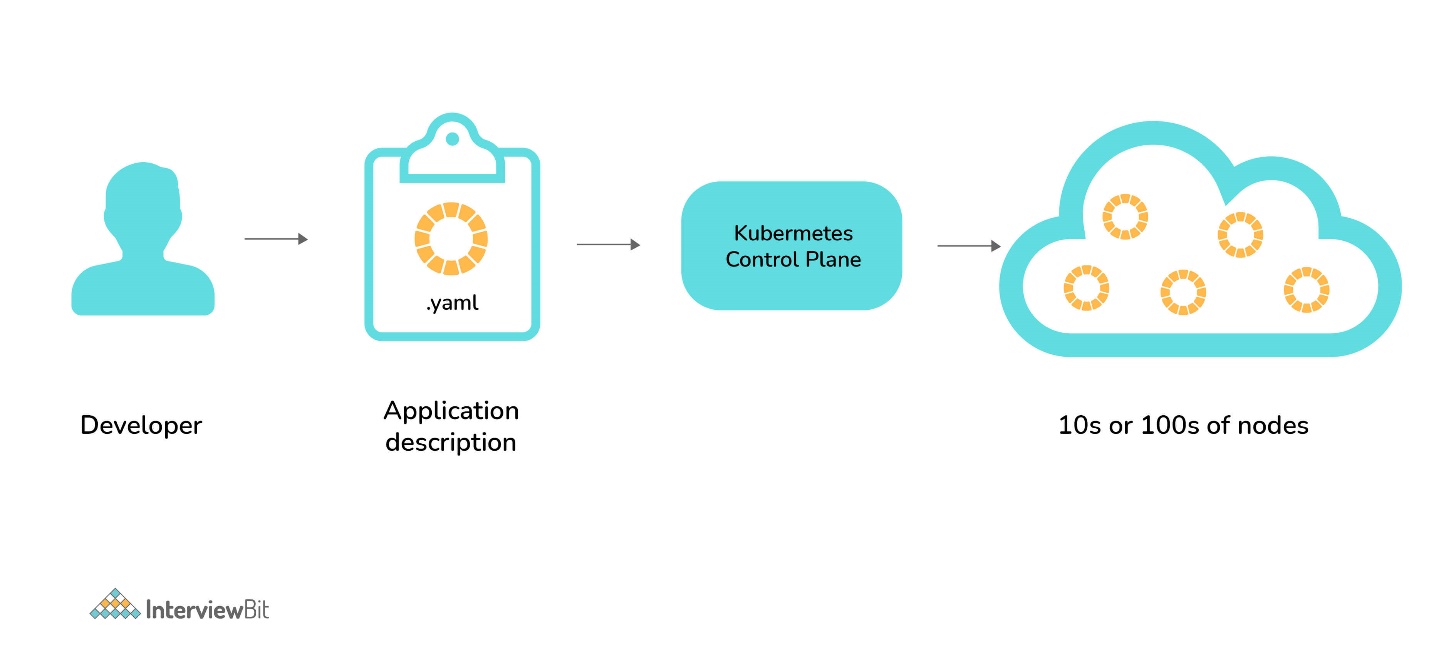
Pods are the smallest unit of objects that can be deployed on Kubernetes, Kubernetes packages one or more containers into a higher-level structure called a pod. Pod runs one level higher to the container.

A POD always runs on a Node but they share a 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, manipulate the state of an object in Kubernetes.

POD

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: "\_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?

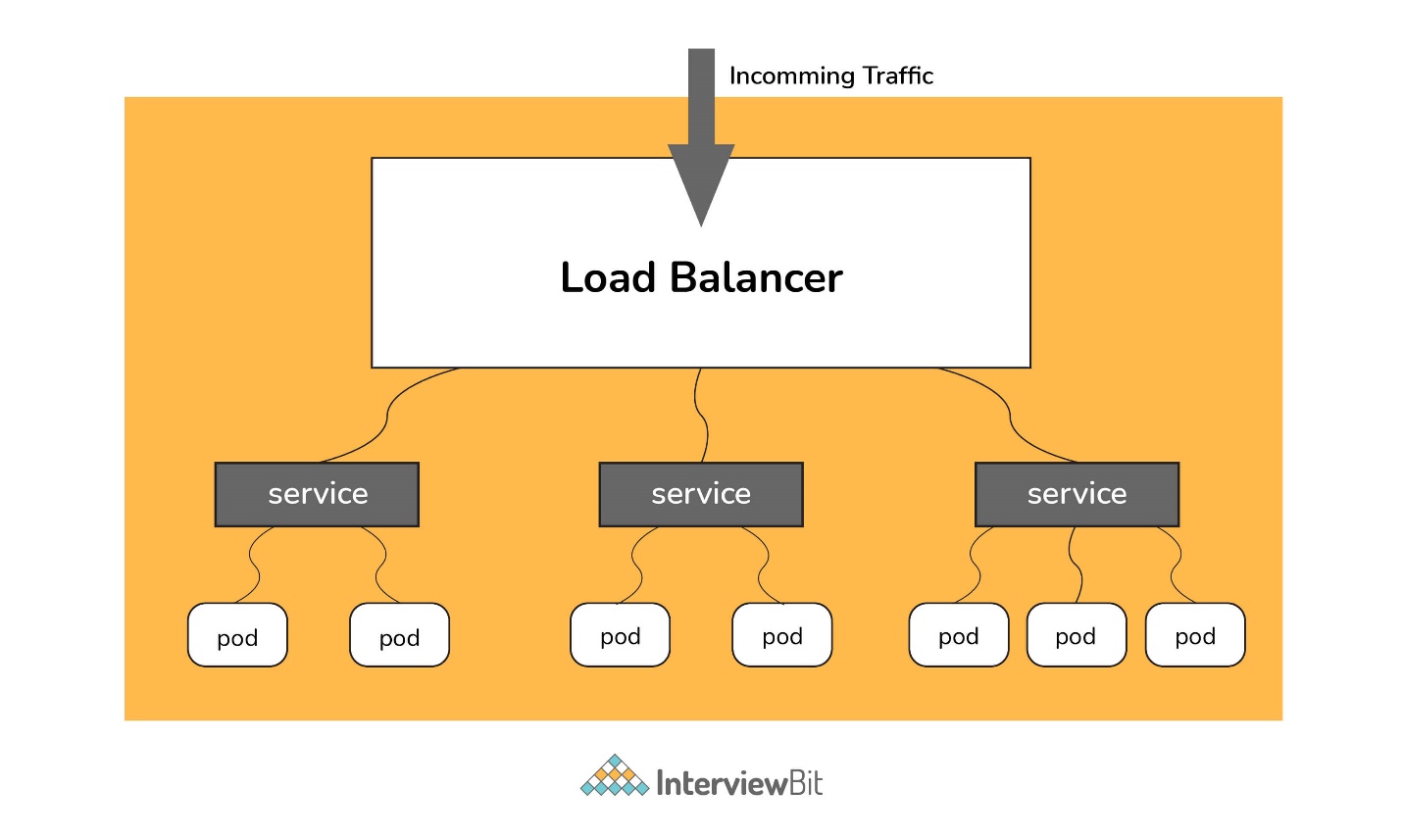
 init containers will set a stage for you before running the actual POD.

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

# Top 50 Kubernetes Interview Questions and Answer for 2023

Kubernetes has been the buzzword in today’s market and is the best orchestration tool. It attracts many experienced professionals who want to advance their careers by a notch. Multinational companies such as Huwaei, Pokemon, eBay, Yahoo Japan, SAP, Open AI, and Sound Cloud use Kubernetes in their day-to-day activities. But there is a lack of Kubernetes Certified professionals in the market. I believe you already know these facts, which have made you land on this Kubernetes Interview Questions article, which will help you know the top questions asked in interviews. For further details, refer to the [Kubernetes Certification Course](https://www.edureka.co/kubernetes-certification).

**Top Kubernetes Interview Questions**

1. [How is Kubernetes different from Docker Swarm?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#one)
2. [What is Kubernetes?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#two)
3. [How is Kubernetes related to Docker?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#three)
4. [What is the difference between deploying applications on hosts and containers?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#four)
5. [What is Container Orchestration?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#five)
6. [What is the need for Container Orchestration?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#six)
7. [What are the features of Kubernetes?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#seven)
8. [How does Kubernetes simplify containerized Deployment?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#eight)
9. [What do you know about clusters in Kubernetes?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#nine)
10. [What is Google Container Engine?](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#ten)

In this blog on Kubernetes Interview Questions, I will be discussing the top Kubernetes related questions asked in your interviews. So, for your better understanding I have divided this blog into the following 4 sections:

* [Kubernetes Basic Interview Questions](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#Basic%20Kubernetes%20Interview%20Questions)
* [Architecture-Based Kubernetes Interview Questions](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#Architecture-Based%20Interview%20Questions)
* [Scenario-Based Kubernetes Interview Questions](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#Scenario-Based%20Interview%20Questions)
* [Multiple Choice Kubernetes Interview Questions](https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/#Multiple%20Choice%20Questions)

## Kubernetes Full Course in 7 Hours | Kubernetes Tutorial | Kubernetes Training | Edureka

This Edureka Kubernetes Full Course video will help you understand and learn the fundamentals of Kubernetes. This Kubernetes Tutorial is ideal for both beginners as well as professionals who want to master the fundamentals of Kubernetes.

So let’s get started guys!!

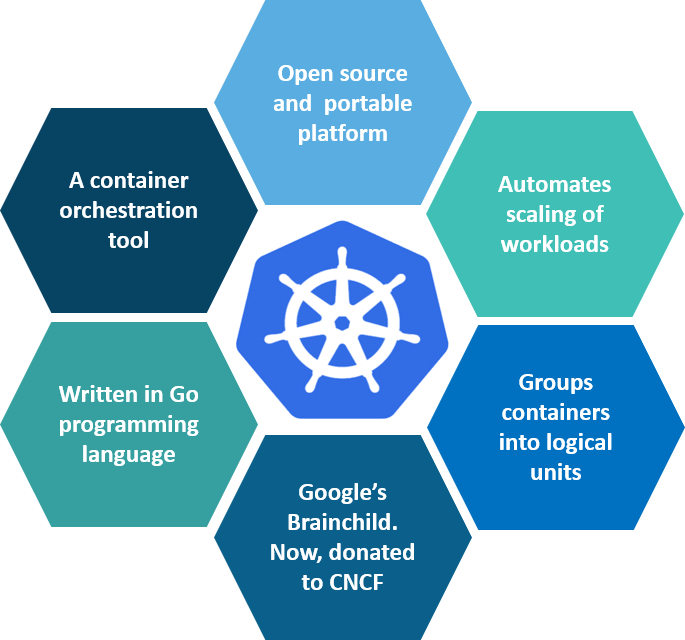
**Basic Kubernetes Interview Questions**

This section of questions will consist of all those basic questions that you need to know related to the working of Kubernetes.

### **Q1. How is Kubernetes different from Docker Swarm?**

|  |  |  |
| --- | --- | --- |
| **Features** | **Kubernetes** | **Docker Swarm** |
| **Installation & Cluster Config** | Setup is very complicated, but once installed cluster is robust. | Installation is very simple, but the cluster is not robust. |
| **GUI** | GUI is the [Kubernetes Dashboard](https://www.edureka.co/blog/kubernetes-dashboard/). | There is no GUI. |
| **Scalability** | Highly scalable and scales fast. | Highly scalable and scales 5x faster than Kubernetes. |
| **Auto-scaling** | Kubernetes can do auto-scaling. | Docker swarm cannot do auto-scaling. |
| **Load Balancing** | Manual intervention needed for load balancing traffic between different containers and pods. | Docker swarm does auto load balancing of traffic between containers in the cluster. |
| **Rolling Updates & Rollbacks** | Can deploy rolling updates and does automatic rollbacks. | Can deploy rolling updates, but not automatic rollback. |
| **DATA Volumes** | Can share storage volumes only with the other containers in the same pod. | Can share storage volumes with any other container. |
| **Logging & Monitoring** | In-built tools for logging and monitoring. | 3rd party tools like ELK stack should be used for logging and monitoring. |

### **Q2. What is Kubernetes?**



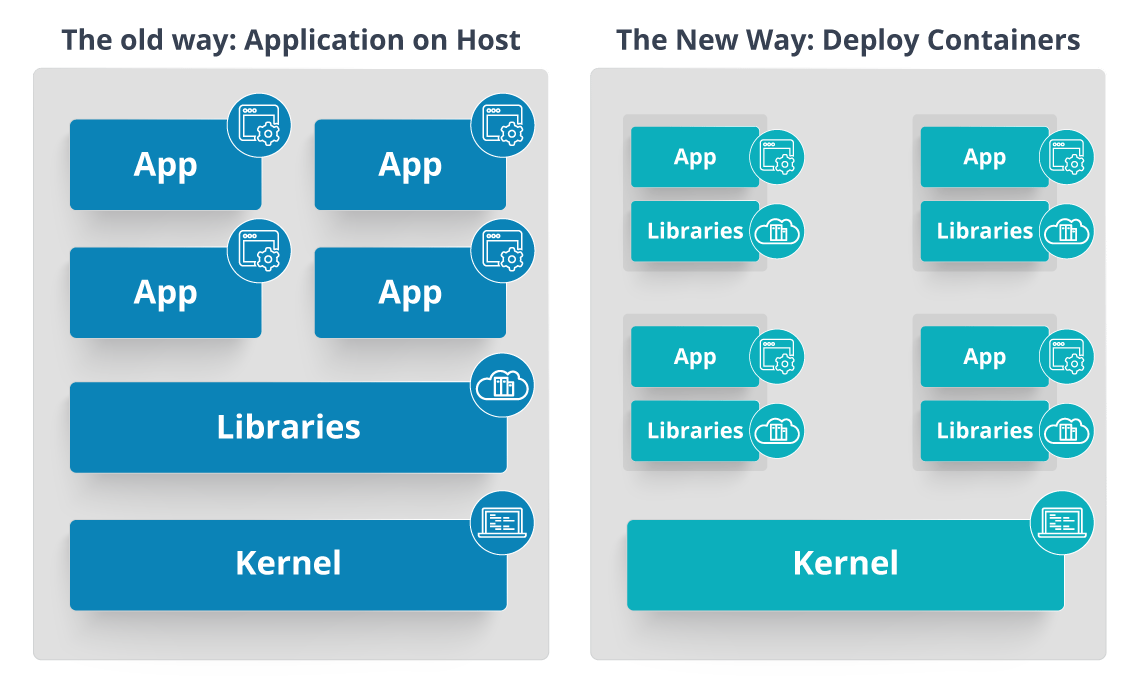
### **Fig 1: What is Kubernetes – Kubernetes Interview Questions**

Kubernetes is an open-source container management tool that holds the responsibilities of container deployment, scaling & descaling of containers & load balancing. Being Google’s brainchild, it offers excellent community and works brilliantly with all the cloud providers. So, we can say that Kubernetes is not a containerization platform, but it is a multi-container management solution.

### **Q3. How is Kubernetes related to Docker?**

It’s a known fact that Docker provides the lifecycle management of containers and a Docker image builds the runtime containers. But, since these individual containers have to communicate, Kubernetes is used. So, Docker builds the containers and these containers communicate with each other via Kubernetes. So, containers running on multiple hosts can be manually linked and orchestrated using Kubernetes.

### **Q4. What is the difference between deploying applications on hosts and containers?**



### **Fig 2: Deploying Applications On Host vs Containers – Kubernetes Interview Questions**

Refer to the above diagram. The left side architecture represents deploying applications on hosts. So, this kind of architecture will have an operating system and then the operating system will have a kernel that will have various libraries installed on the operating system needed for the application. So, in this kind of framework you can have n number of applications and all the applications will share the libraries present in that operating system whereas while deploying applications in containers the architecture is a little different.

This kind of architecture will have a kernel and that is the only thing that’s going to be the only thing common between all the applications. So, if there’s a particular application that needs Java then that particular application we’ll get access to Java and if there’s another application that needs Python then only that particular application will have access to Python.

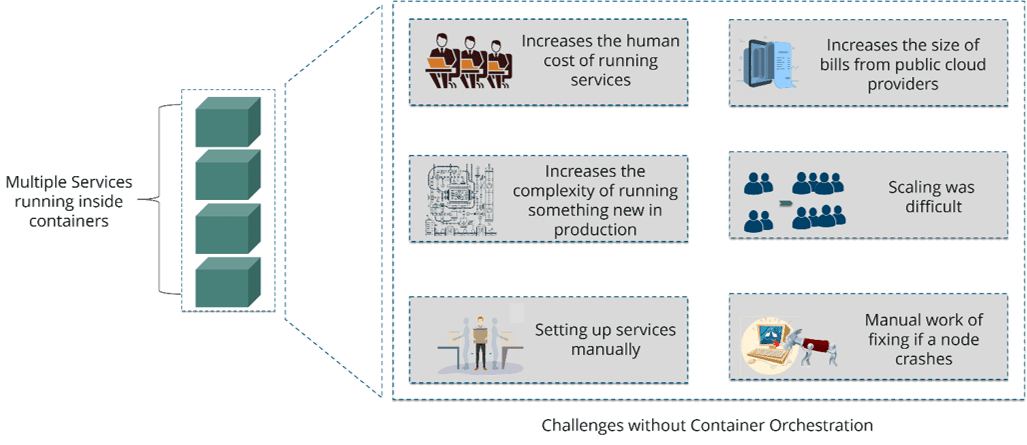
The individual blocks that you can see on the right side of the diagram are basically containerized and these are isolated from other applications. So, the applications have the necessary libraries and binaries isolated from the rest of the system, and cannot be encroached by any other application.

### **Q5. What is Container Orchestration?**

Consider a scenario where you have 5-6 microservices for an application. Now, these microservices are put in individual containers, but won’t be able to communicate without container orchestration. So, as orchestration means the amalgamation of all instruments playing together in harmony in music, similarly container orchestration means all the services in individual containers working together to fulfill the needs of a single server.

### **Q6. What is the need for Container Orchestration?**

Consider you have 5-6 microservices for a single application performing various tasks, and all these microservices are put inside containers. Now, to make sure that these containers communicate with each other we need container orchestration.

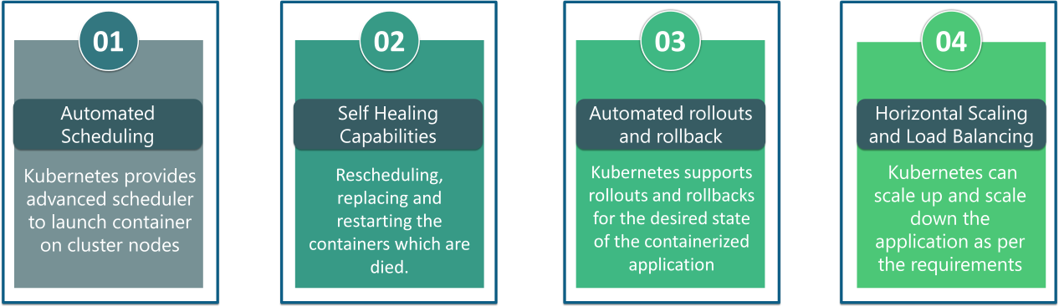


### **Fig 3: Challenges Without Container Orchestration – Kubernetes Interview Questions**

As you can see in the above diagram, there were also many challenges that came into place without the use of container orchestration. So, to overcome these challenges the container orchestration came into place.

### **Q7. What are the features of Kubernetes?**

The features of Kubernetes, are as follows:



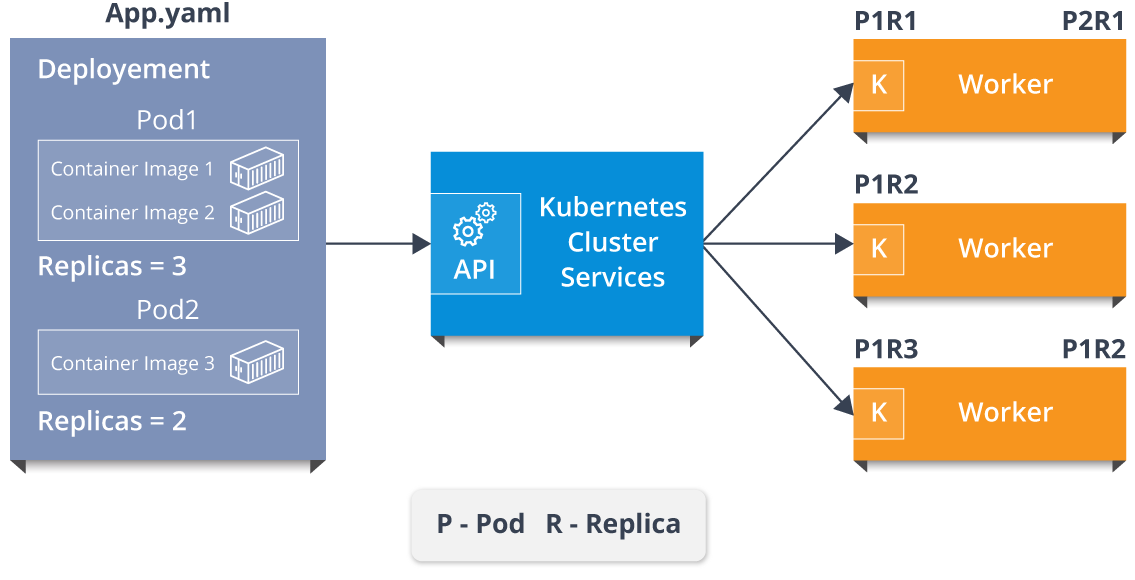
### **Fig 4: Features Of Kubernetes – Kubernetes Interview Questions**

### **Q8. How does Kubernetes simplify containerized Deployment?**

As a typical application would have a cluster of containers running across multiple hosts, all these containers would need to talk to each other. So, to do this you need something big that would load balance, scale & monitor the containers. Since Kubernetes is cloud-agnostic and can run on any public/private providers it must be your choice simplify containerized deployment.

### **Q9. What do you know about clusters in Kubernetes?**

The fundamental behind Kubernetes is that we can enforce the desired state management, by which I mean that we can feed the cluster services of a specific configuration, and it will be up to the cluster services to go out and run that configuration in the infrastructure.



### **Fig 5: Representation Of Kubernetes Cluster – Kubernetes Interview Questions**

So, as you can see in the above diagram, the deployment file will have all the configurations required to be fed into the cluster services. Now, the deployment file will be fed to the API and then it will be up to the cluster services to figure out how to schedule these pods in the environment and make sure that the right number of pods are running.

So, the API which sits in front of services, the worker nodes & the Kubelet process that the nodes run, all together make up the Kubernetes Cluster.

### **Q10. What is Google Container Engine?**

**Google Container Engine (GKE)**is an open-source management platform for Docker containers and clusters. This Kubernetes based engine supports only those clusters which run within Google’s public cloud services.

## ****Kubernetes Interview Questions****

### **Q11.  What is Heapster?**

Heapster is a cluster-wide aggregator of data provided by Kubelet running on each node. This container management tool is supported natively on Kubernetes cluster and runs as a pod, just like any other pod in the cluster. So, it basically discovers all nodes in the cluster and queries usage information from the Kubernetes nodes in the cluster, via on-machine Kubernetes agent.

### **Q12.  What is Minikube?**

Minikube is a tool that makes it easy to run Kubernetes locally. This runs a single-node Kubernetes cluster inside a virtual machine.

### **Q13.  What is Kubectl?**

Kubectl is the platform using which you can pass commands to the cluster. So, it basically provides the CLI to run commands against the Kubernetes cluster with various ways to create and manage the Kubernetes component.

### **Q14.  What is Kubelet?**

This is an agent service which runs on each node and enables the slave to communicate with the master. So, Kubelet works on the description of containers provided to it in the PodSpec and makes sure that the containers described in the PodSpec are healthy and running.

### **Q15. What do you understand by a node in Kubernetes? Kubernetes Node - Kubernetes Interview Questions - Edureka**

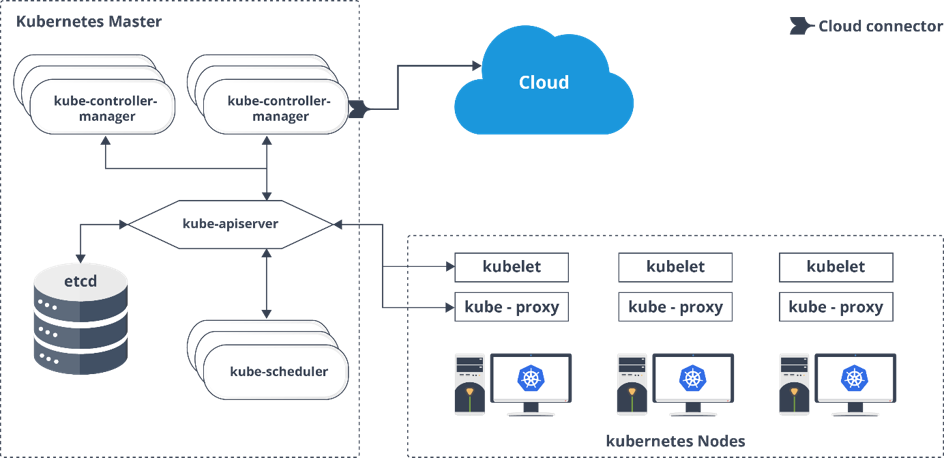
### **Fig 6: Node In Kubernetes – Kubernetes Interview Questions**

**Architecture-Based Kubernetes Interview Questions**

This section of questions will deal with the questions related to the architecture of Kubernetes.

### **Q1. What are the different components of Kubernetes Architecture?**

The [Kubernetes Architecture](https://www.edureka.co/blog/kubernetes-architecture/) has mainly 2 components – the master node and the worker node. As you can see in the below diagram, the master and the worker nodes have many inbuilt components within them. The master node has the kube-controller-manager, kube-apiserver, kube-scheduler, etcd. Whereas the worker node has kubelet and kube-proxy running on each node.



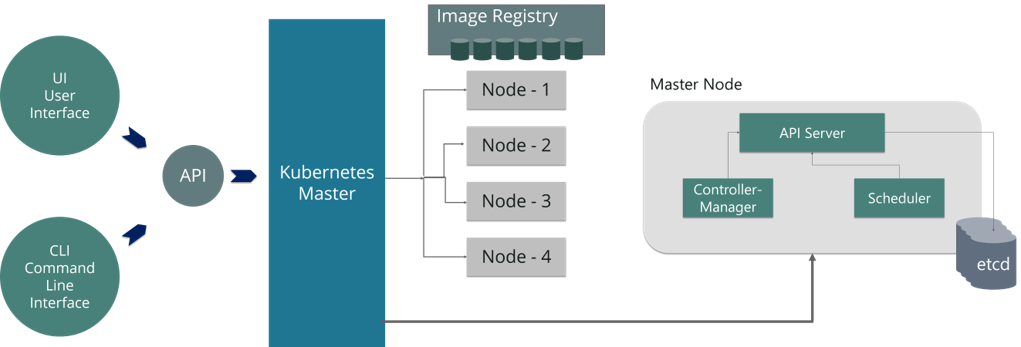
### **Fig 7: Architecture Of Kubernetes – Kubernetes Interview Questions**

### **Q2. What do you understand by Kube-proxy?**

Kube-proxy can run on each and every node and can do simple TCP/UDP packet forwarding across backend network service. So basically, it is a network proxy that reflects the services as configured in Kubernetes API on each node. So, the Docker-linkable compatible environment variables provide the cluster IPs and ports which are opened by proxy.

### **Q3.  Can you brief on the working of the master node in Kubernetes?**

Kubernetes master controls the nodes and inside the nodes the containers are present. Now, these individual containers are contained inside pods and inside each pod, you can have a various number of containers based upon the configuration and requirements. So, if the pods have to be deployed, then they can either be deployed using user interface or command-line interface. Then, these pods are scheduled on the nodes, and based on the resource requirements, the pods are allocated to these nodes. The kube-apiserver makes sure that there is communication established between the Kubernetes node and the master components.



[[](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum)](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

### [Kubernetes Certification Training Course: Administrator (CKA)](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

* *[Instructor-led Sessions](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[Assessments](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[24 x 7 Expert Support](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

### **Fig 8: Representation Of Kubernetes Master Node – Kubernetes Interview Questions**

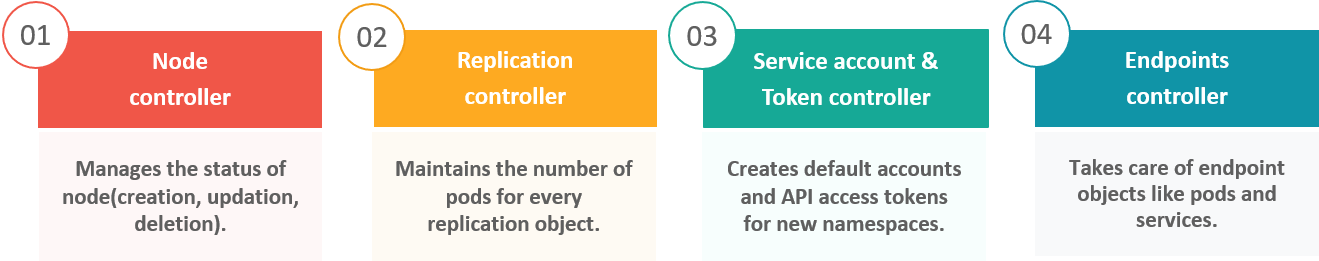
### **Q4.  What is the role of kube-apiserver and kube-scheduler?**

The kube – apiserver follows the scale-out architecture and is the front end of the master node control panel. This exposes all the APIs of the Kubernetes Master node components and is responsible for establishing communication between Kubernetes Node and the Kubernetes master components.

The kube-scheduler is responsible for distributing and managing the workload on the worker nodes. So, it selects the most suitable node to run the unscheduled pod based on resource requirements and keeps track of resource utilization. It ensures that the workload is not scheduled on already full nodes.

### **Q5.  Can you brief me about the Kubernetes controller manager?**

Multiple controller processes run on the master node but are compiled together to run as a single process: the Kubernetes Controller Manager. So, Controller Manager is a daemon that embeds controllers and does namespace creation and garbage collection. It owns the responsibility and communicates with the API server to manage the end-points.

So, the different types of controller manager running on the master node are :  


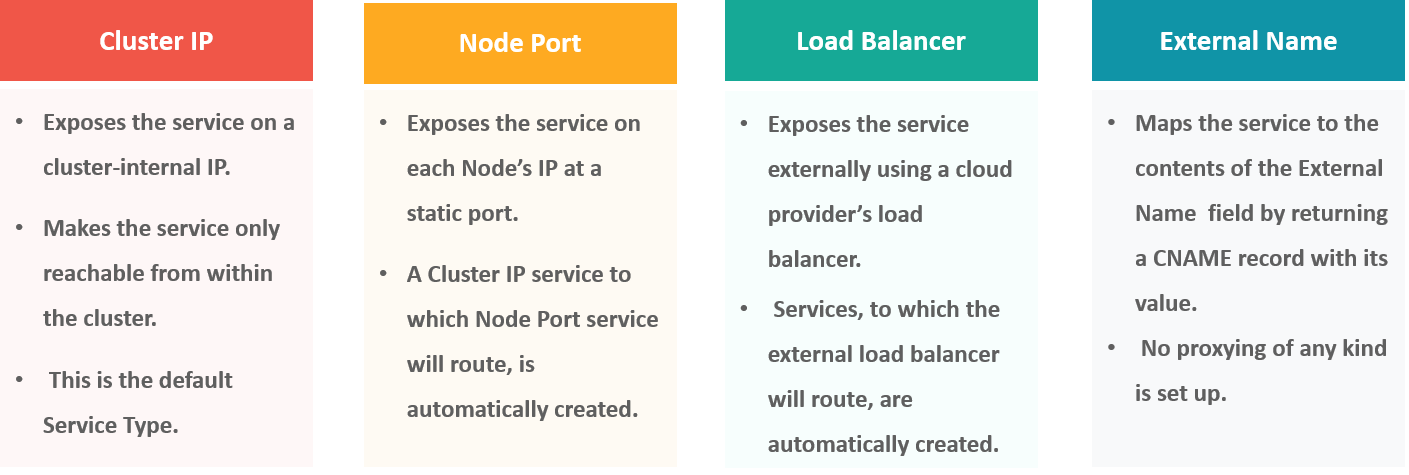
### **Fig 9: Types Of Controllers – Kubernetes Interview Questions**

### **Q6.  What is ETCD?**

Etcd is written in [Go programming language](https://www.edureka.co/blog/golang-tutorial/) and is a distributed key-value store used for coordinating distributed work. So, Etcd stores the configuration data of the Kubernetes cluster, representing the state of the cluster at any given point in time.

### **Q7. What are the different types of services in Kubernetes?**

The following are the different types of services used:



### **Fig 10: Types Of Services – Kubernetes Interview Questions**

### **Q8. What do you understand by load balancer in Kubernetes?**

A load balancer is one of the most common and standard ways of exposing service. There are two types of load balancer used based on the working environment i.e. either the Internal Load Balancer or the External Load Balancer. The Internal Load Balancer automatically balances load and allocates the pods with the required configuration whereas the External Load Balancer directs the traffic from the external load to the backend pods.

## ****Kubernetes Interview Questions****

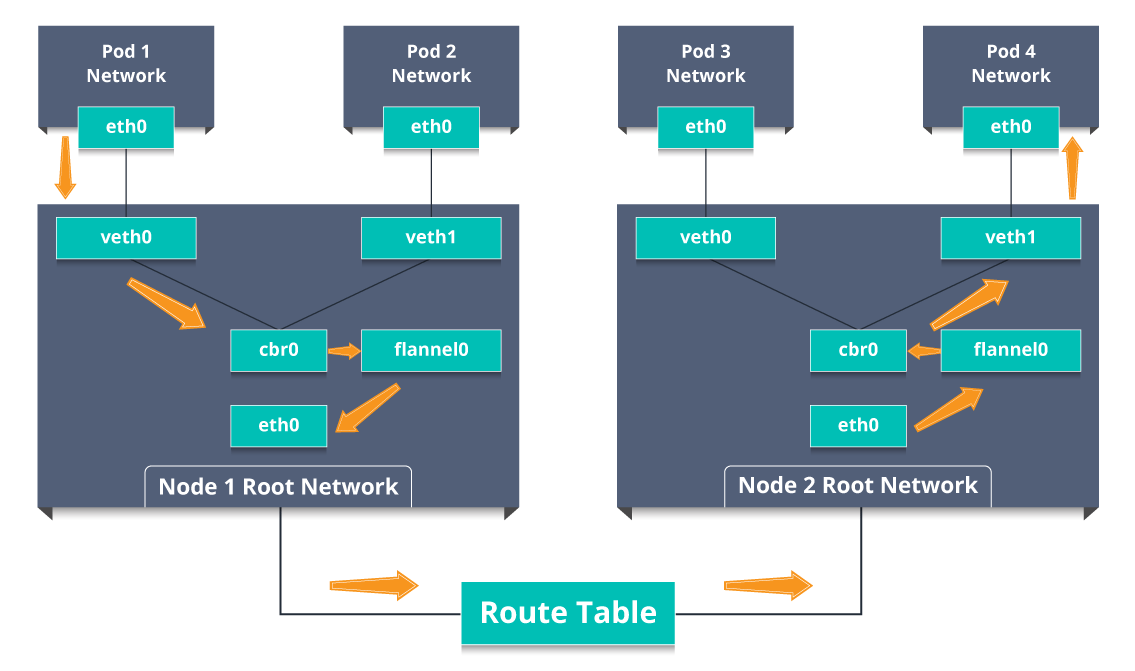
### **Q9. What is Ingress network, and how does it work?**

Ingress network is a collection of rules that acts as an entry point to the Kubernetes cluster. This allows inbound connections, which can be configured to give services externally through reachable URLs, load balance traffic, or by offering name-based virtual hosting. So, Ingress is an API object that manages external access to the services in a cluster, usually by HTTP and is the most powerful way of exposing service.

Now, let me explain to you the working of Ingress network with an example.

There are 2 nodes having the pod and root network namespaces with a Linux bridge. In addition to this, there is also a new virtual ethernet device called flannel0(network plugin) added to the root network.

Now, suppose we want the packet to flow from pod1 to pod 4. Refer to the below diagram.



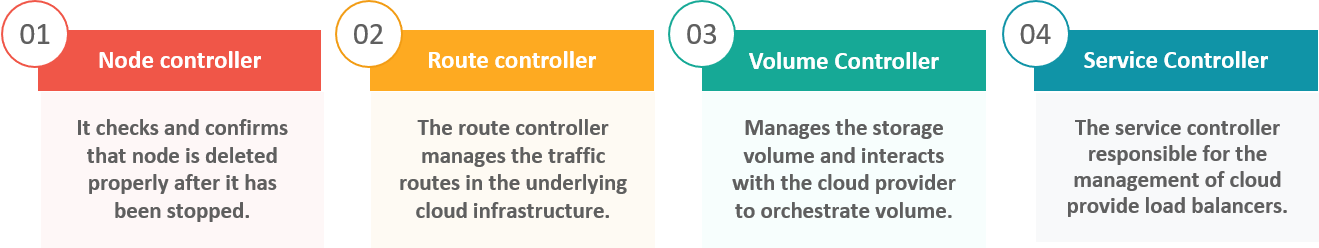
### **Fig 11: Working Of Ingress Network – Kubernetes Interview Questions**

* So, the packet leaves pod1’s network at eth0 and enters the root network at veth0.
* Then it is passed on to cbr0, which makes the ARP request to find the destination and it is found out that nobody on this node has the destination IP address.
* So, the bridge sends the packet to flannel0 as the node’s route table is configured with flannel0.
* Now, the flannel daemon talks to the API server of Kubernetes to know all the pod IPs and their respective nodes to create mappings for pods IPs to node IPs.
* The network plugin wraps this packet in a UDP packet with extra headers changing the source and destination IP’s to their respective nodes and sends this packet out via eth0.
* Now, since the route table already knows how to route traffic between nodes, it sends the packet to the destination node2.
* The packet arrives at eth0 of node2 and goes back to flannel0 to de-capsulate and emits it back in the root network namespace.
* Again, the packet is forwarded to the Linux bridge to make an ARP request to find out the IP that belongs to veth1.
* The packet finally crosses the root network and reaches the destination Pod4.

### **Q10.  What do you understand by Cloud controller manager?**

The Cloud Controller Manager is responsible for persistent storage, network routing, abstracting the cloud-specific code from the core Kubernetes specific code, and managing the communication with the underlying cloud services. It might be split out into several different containers depending on which cloud platform you are running on and then it enables the cloud vendors and Kubernetes code to be developed without any inter-dependency. So, the cloud vendor develops their code and connects with the Kubernetes cloud-controller-manager while running the Kubernetes.

The various types of cloud controller manager are as follows:

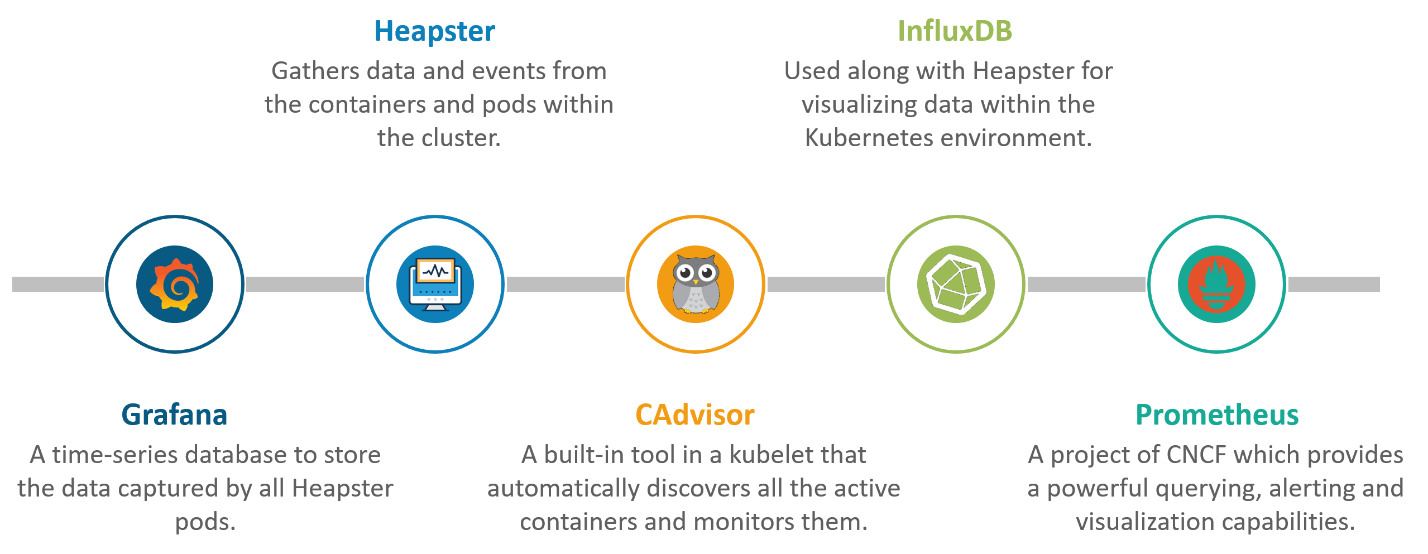


### **Fig 12: Types Of Cloud Controller Manager – Kubernetes Interview Questions**

### **Q11. What is Container resource monitoring?**

As for users, it is really important to understand the performance of the application and resource utilization at all the different abstraction layer, Kubernetes factored the management of the cluster by creating abstraction at different levels like container, pods, services and whole cluster. Now, each level can be monitored and this is nothing but Container resource monitoring.

The various container resource monitoring tools are as follows:



### **Fig 13: Container Resource Monitoring Tools – Kubernetes Interview Questions**

### **Q12. What is the difference between a replica set and a replication controller?**

Replica Set and Replication Controller do almost the same thing. Both ensure that a specified number of pod replicas are running at any given time. The difference comes with the usage of selectors to replicate pods. Replica Set uses Set-Based selectors while replication controllers use Equity-Based selectors.

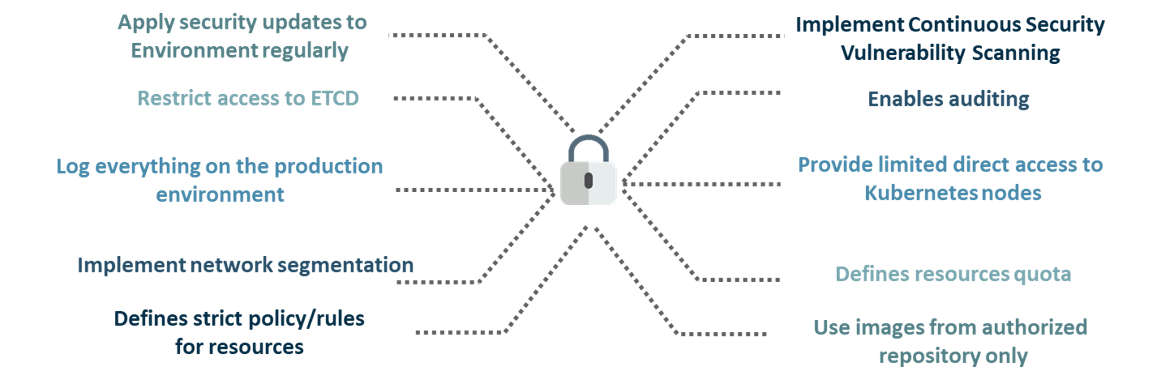
* **Equity-Based Selectors:**This type of selector allows filtering by label key and values. So, in layman’s terms, the equity-based selector will only look for the pods with the exact same phrase as the label.  
  **Example**: Suppose your label key says app=nginx; then, with this selector, you can only look for those pods with label app equal to nginx.
* **Selector-Based Selectors:**This type of selector allows filtering keys according to a set of values. So, in other words, the selector-based selector will look for pods whose label has been mentioned in the set.  
  **Example:** Say your label key says app in (Nginx, NPS, Apache). Then, with this selector, if your app is equal to any of Nginx, NPS, or Apache, the selector will take it as a true result.

### **Q13. What is a Headless Service?**

Headless Service is similar to that of a ‘Normal’ service but does not have a Cluster IP. This service enables you to directly reach the pods without the need to access them through a proxy.

### **Q14. What are the best security measures that you can take while using Kubernetes?**

The following are the best security measures that you can follow while using Kubernetes:

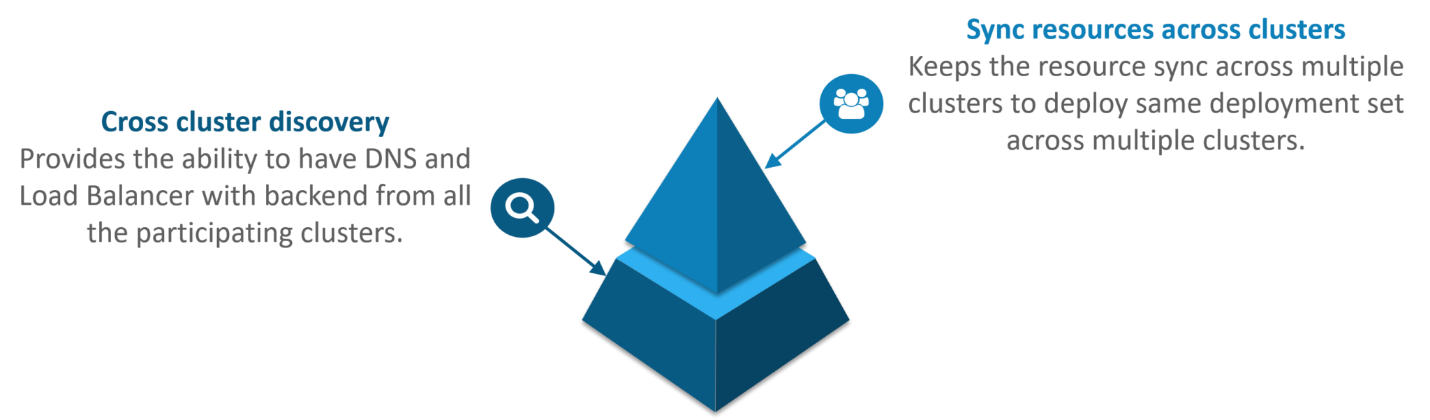


### **Fig 14: Best Security Measures – Kubernetes Interview Questions**

### **Q15. What are federated clusters?**

Multiple Kubernetes clusters can be managed as a single cluster with the help of federated clusters. So, you can create multiple Kubernetes clusters within a data center/cloud and use federation to control/manage them all at one place.

The federated clusters can achieve this by doing the following two things. Refer to the below diagram.



### **Fig 15: Federated Clusters – Kubernetes Interview Questions**

## ****Scenario-Based Kubernetes Interview Questions****

This section of questions will consist of various scenario-based questions that you may face in your interviews.

**Scenario 1:** Suppose a company built on monolithic architecture handles numerous products. Now, as the company expands in today’s scaling industry, their monolithic architecture started causing problems.

How do you think the company shifted from monolithic to microservices and deploy their services containers?

**Solution:**

As the company’s goal is to shift from their monolithic application to microservices, they can end up building piece by piece, in parallel and just switch configurations in the background. Then they can put each of these built-in microservices on the Kubernetes platform. So, they can start by migrating their services once or twice and monitor them to make sure everything is running stable. Once they feel everything is going good, then they can migrate the rest of the application into their Kubernetes cluster.

**Scenario 2:**Consider a multinational company with a very much distributed system, with a large number of data centers, virtual machines, and many employees working on various tasks.

How do you think can such a company manage all the tasks in a consistent way with Kubernetes?

**Solution:**

As all of us know that I.T. departments launch thousands of containers, with tasks running across a numerous number of nodes across the world in a distributed system.

In such a situation the company can use something that offers them agility, scale-out capability, and DevOps practice to the cloud-based applications.

So, the company can, therefore, use Kubernetes to customize their scheduling architecture and support multiple container formats. This makes it possible for the affinity between container tasks that gives greater efficiency with an extensive support for various container networking solutions and container storage.

Next

**Scenario 3:**Consider a situation, where a company wants to increase its efficiency and the speed of its technical operations by maintaining minimal costs.

How do you think the company will try to achieve this?

**Solution:**

The company can implement the DevOps methodology, by building a CI/CD pipeline, but one problem that may occur here is the configurations may take time to go up and running. So, after implementing the CI/CD pipeline the company’s next step should be to work in the cloud environment. Once they start working on the cloud environment, they can schedule containers on a cluster and can orchestrate with the help of Kubernetes. This kind of approach will help the company reduce their deployment time, and also get faster across various environments.

**Scenario 4:** Suppose a company wants to revise it’s deployment methods and wants to build a platform which is much more scalable and responsive.

How do you think this company can achieve this to satisfy their customers?

**Solution:**

In order to give millions of clients the digital experience they would expect, the company needs a platform that is scalable, and responsive, so that they could quickly get data to the client website. Now, to do this the company should move from their private data centers (if they are using any) to any cloud environment such as AWS. Not only this, but they should also implement the microservice architecture so that they can start using Docker containers. Once they have the base framework ready, then they can start using the best orchestration platform available i.e. Kubernetes. This would enable the teams to be autonomous in building applications and delivering them very quickly.

**Scenario 5:** Consider a multinational company with a very much distributed system, looking forward to solving the monolithic code base problem.

How do you think the company can solve their problem?

**Solution**

Well, to solve the problem, they can shift their monolithic code base to a microservice design and then each and every microservices can be considered as a container. So, all these containers can be deployed and orchestrated with the help of Kubernetes.

Want to get Kubernetes Certified? [View Batches Now](https://www.edureka.co/kubernetes-certification)

## ****Kubernetes Interview Questions****

**Scenario 6:**All of us know that the shift from monolithic to microservices solves the problem from the development side, but increases the problem at the deployment side.

How can the company solve the problem on the deployment side?

**Solution**

The team can experiment with container orchestration platforms, such as Kubernetes and run it in data centers. So, with this, the company can generate a templated application, deploy it within five minutes, and have actual instances containerized in the staging environment at that point. This kind of Kubernetes project will have dozens of microservices running in parallel to improve the production rate as even if a node goes down, then it can be rescheduled immediately without performance impact.

**Scenario 7:**Suppose a company wants to optimize the distribution of its workloads, by adopting new technologies.

How can the company achieve this distribution of resources efficiently?

**Solution**

The solution to this problem is none other than Kubernetes. Kubernetes makes sure that the resources are optimized efficiently, and only those resources are used which are needed by that particular application. So, with the usage of the best container orchestration tool, the company can achieve the distribution of resources efficiently.

**Scenario 8:**Consider a carpooling company wants to increase their number of servers by simultaneously scaling their platform.

How do you think will the company deal with the servers and their installation?

**Solution**

The company can adopt the concept of containerization. Once they deploy all their application into containers, they can use Kubernetes for orchestration and use container monitoring tools like Prometheus to monitor the actions in containers. So, with such usage of containers, giving them better capacity planning in the data center because they will now have fewer constraints due to this abstraction between the services and the hardware they run on.

**Scenario 9:**Consider a scenario where a company wants to provide all the required hand-outs to its customers having various environments.

How do you think they can achieve this critical target in a dynamic manner?

**Solution**

The company can use Docker environments, to put together a cross-sectional team to build a web application using Kubernetes. This kind of framework will help the company achieve the goal of getting the required things into production within the shortest time frame. So, with such a machine running, the company can give the hands-outs to all the customers having various environments.

**Scenario 10**: Suppose a company wants to run various workloads on different cloud infrastructure from bare metal to a public cloud.

How will the company achieve this in the presence of different interfaces?

**Solution**

The company can decompose its infrastructure into microservices and then adopt Kubernetes. This will let the company run various workloads on different cloud infrastructures.

### **Multiple Choice Kubernetes Interview Questions**

This section of questions will consist of multiple-choice interview questions, that are frequently asked in interviews.

**Q1. What are minions in the Kubernetes cluster?**

1. They are components of the master node.
2. They are the work-horse / worker node of the cluster.[Ans]
3. They are monitoring engine used widely in kubernetes.
4. They are docker container service.

**Q2. Kubernetes cluster data is stored in which of the following?**

### [Kubernetes C](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=batches" \t "_blank)

1. Kube-apiserver
2. Kubelet
3. Etcd[Ans]
4. None of the above

**Q3. Which of them is a Kubernetes Controller?**

1. ReplicaSet
2. Deployment
3. Rolling Updates
4. Both ReplicaSet and Deployment[Ans]

**Q4. Which of the following are core Kubernetes objects?**

1. Pods
2. Services
3. Volumes
4. All of the above[Ans]

**Q5. The Kubernetes Network proxy runs on which node?**

1. Master Node
2. Worker Node
3. All the nodes[Ans]
4. None of the above

**Q6. What are the responsibilities of** **a node controller?**

1. To assign a CIDR block to the nodes
2. To maintain the list of nodes
3. To monitor the health of the nodes
4. All of the above[Ans]

**Q7. What are the responsibilities of Replication Controller?**

1. Update or delete multiple pods with a single command
2. Helps to achieve the desired state
3. Creates a new pod, if the existing pod crashes
4. All of the above[Ans]

**Q8. How to define a service without a selector?**

1. Specify the external name[Ans]
2. Specify an endpoint with IP Address and port
3. Just by specifying the IP address
4. Specifying the label and api-version

**Q9. What did the 1.8 version of Kubernetes introduce?**

1. Taints and Tolerations[Ans]
2. Cluster level Logging
3. Secrets
4. Federated Clusters

**Q10. The handler invoked by** **Kubelet to check if a container’s IP address is open or not is?**

1. HTTPGetAction
2. ExecAction
3. TCPSocketAction[Ans]
4. None of the above