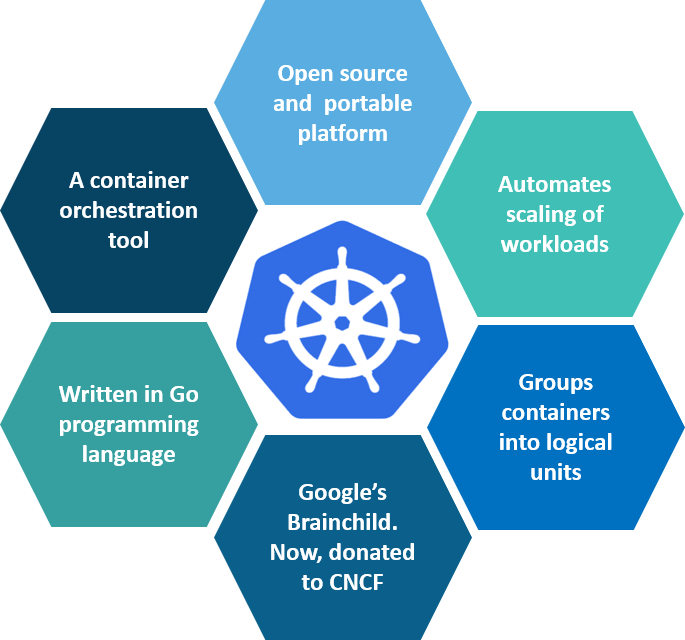
### ****Q1. How is Kubernetes different from Docker Swarm?****

|  |  |  |
| --- | --- | --- |
| ****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/" \t "https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/_blank). | 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?****



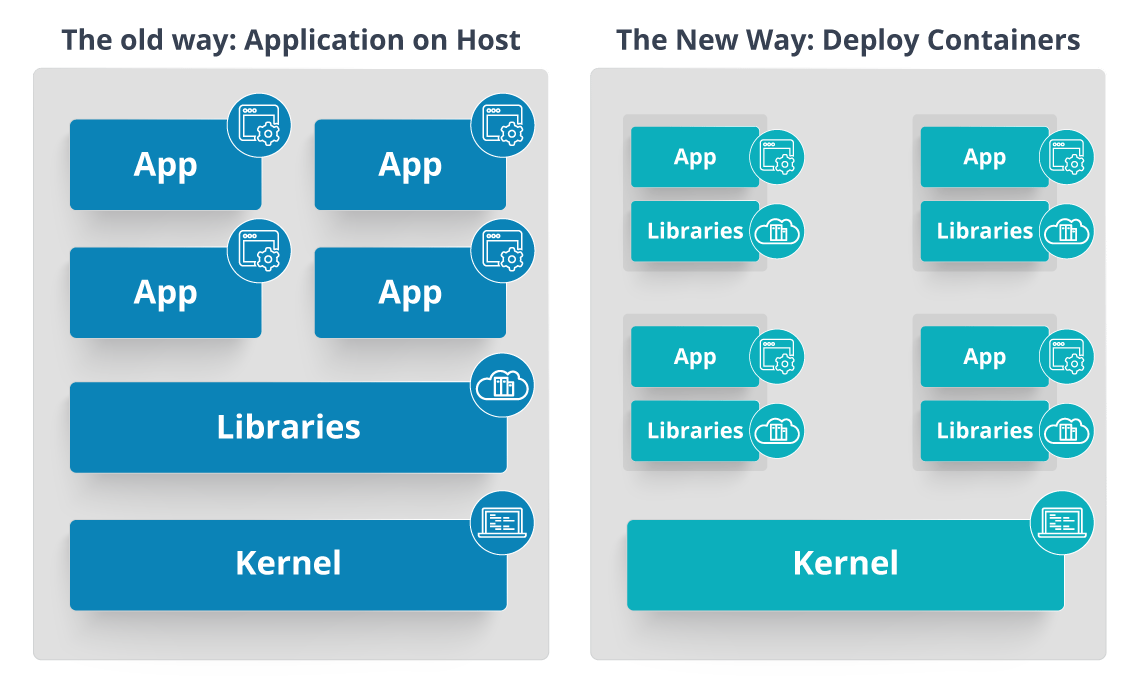
### ****Fig 1:**** What is Kubernetes – Kubernetes Interview Questions

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.

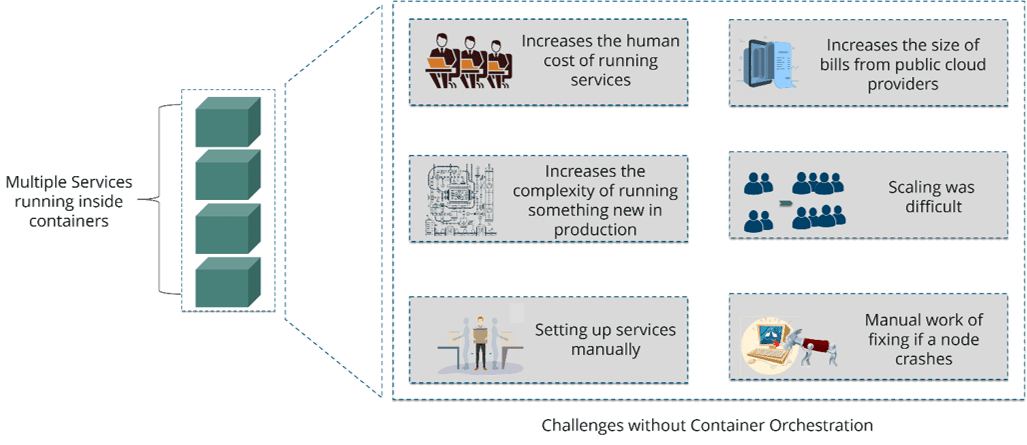
The individual blocks that you can see on the right side of the diagram are basically containerized and these are isolated from other applications. So, the applications have the necessary libraries and binaries isolated from the rest of the system, and cannot be encroached by any other application.

### ****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.

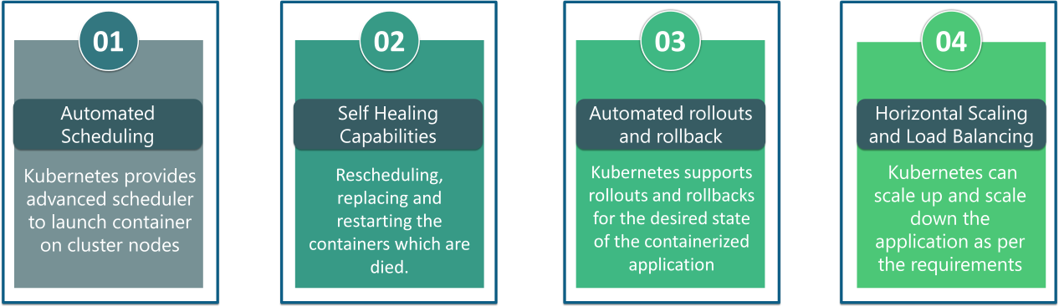


### ****Fig 3:**** Challenges Without Container Orchestration – Kubernetes Interview Questions

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:



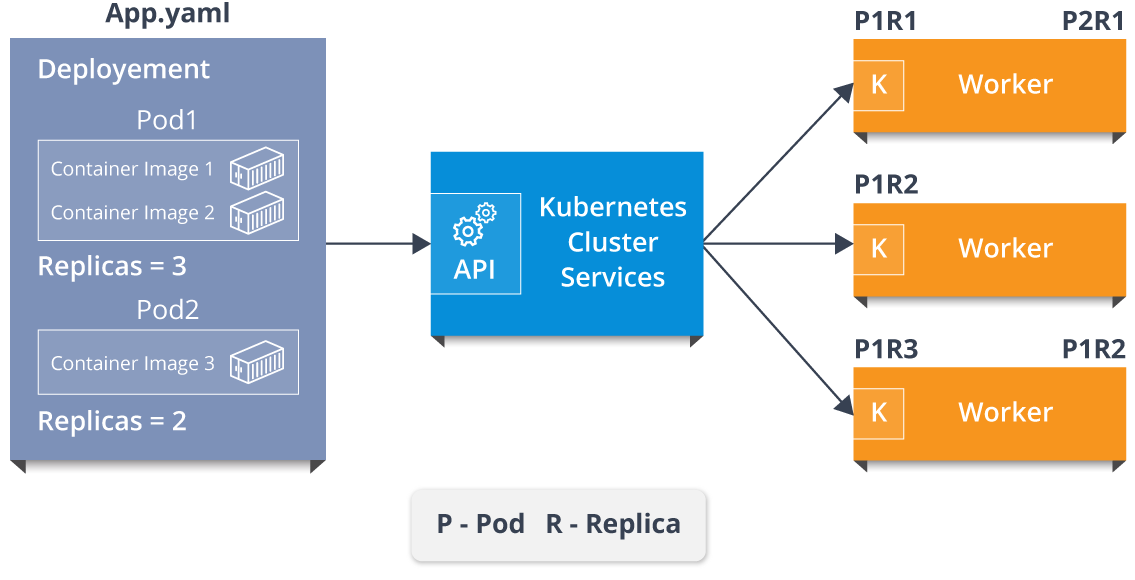
### ****Fig 4:**** Features Of Kubernetes – Kubernetes Interview Questions

### ****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?**** IMG_261

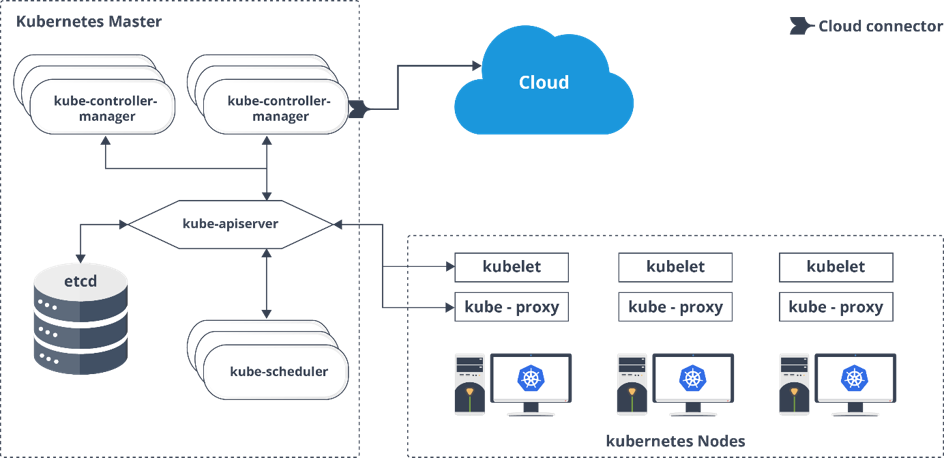
### ****Fig 6:****Node In Kubernetes – Kubernetes Interview Questions

## ****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/" \t "https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/_blank) 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.



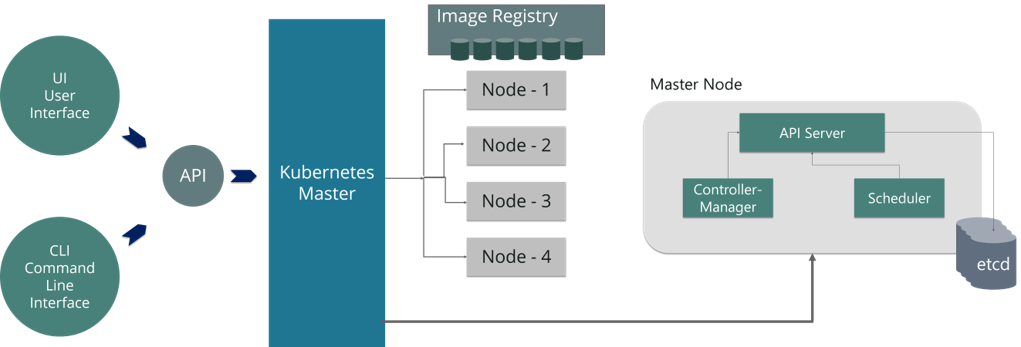
### ****Fig 7:**** Architecture Of Kubernetes – Kubernetes Interview Questions

### ****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.



### ****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.

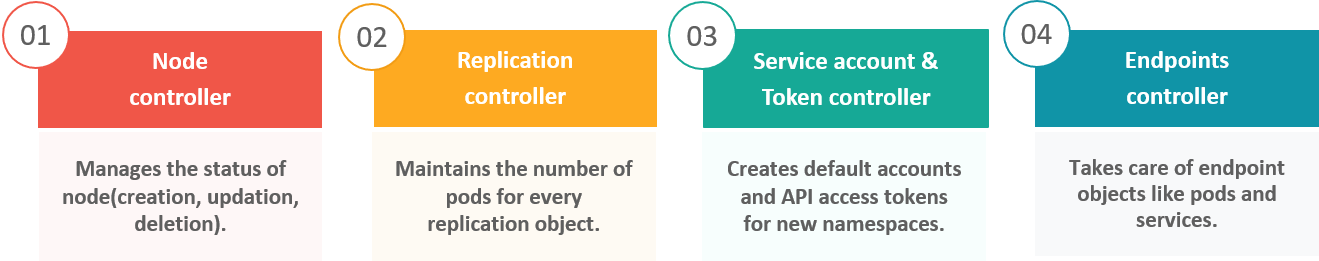
The kube-scheduler is responsible for distributing and managing the workload on the worker nodes. So, it selects the most suitable node to run the unscheduled pod based on resource requirements and keeps track of resource utilization. It ensures that the workload is not scheduled on already full nodes.

### ****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.

* *[7 Expert Support](https://www.edureka.co/kubernetes-certification?utm_source=blogbanner&utm_campaign=curriculum" \t "https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/_blank)*

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

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


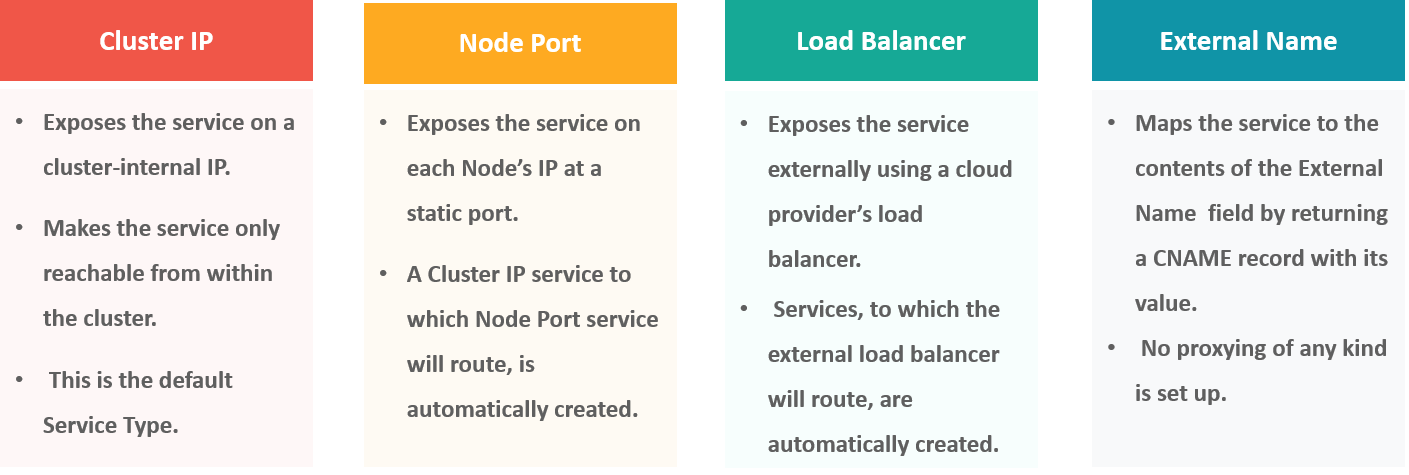
### ****Fig 9:**** Types Of Controllers – Kubernetes Interview Questions

### ****Q6.  What is ETCD?****

Etcd is written in [Go programming language](https://www.edureka.co/blog/golang-tutorial/" \t "https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/_blank) 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****

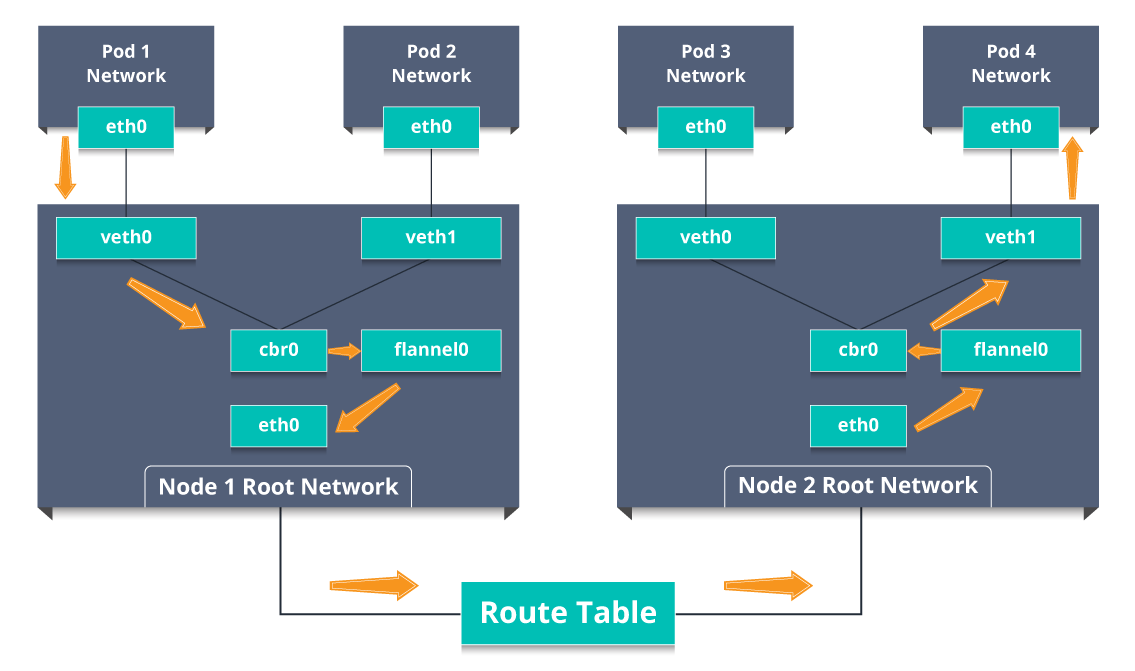
### ****Q9. What is Ingress network, and how does it work?****

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.



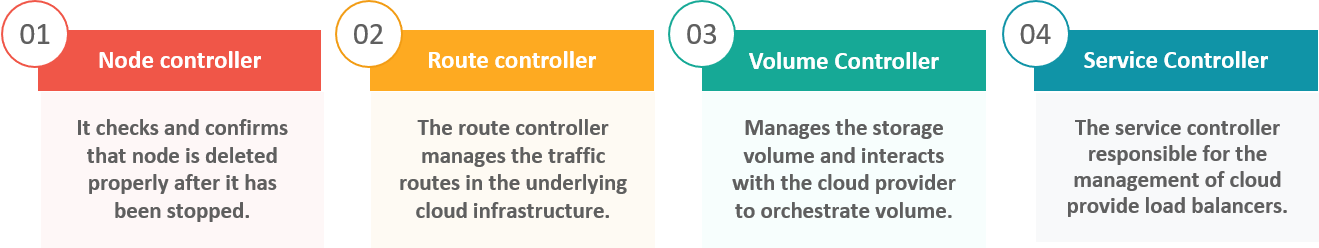
### ****Fig 11:**** Working Of Ingress Network – Kubernetes Interview Questions

* 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:

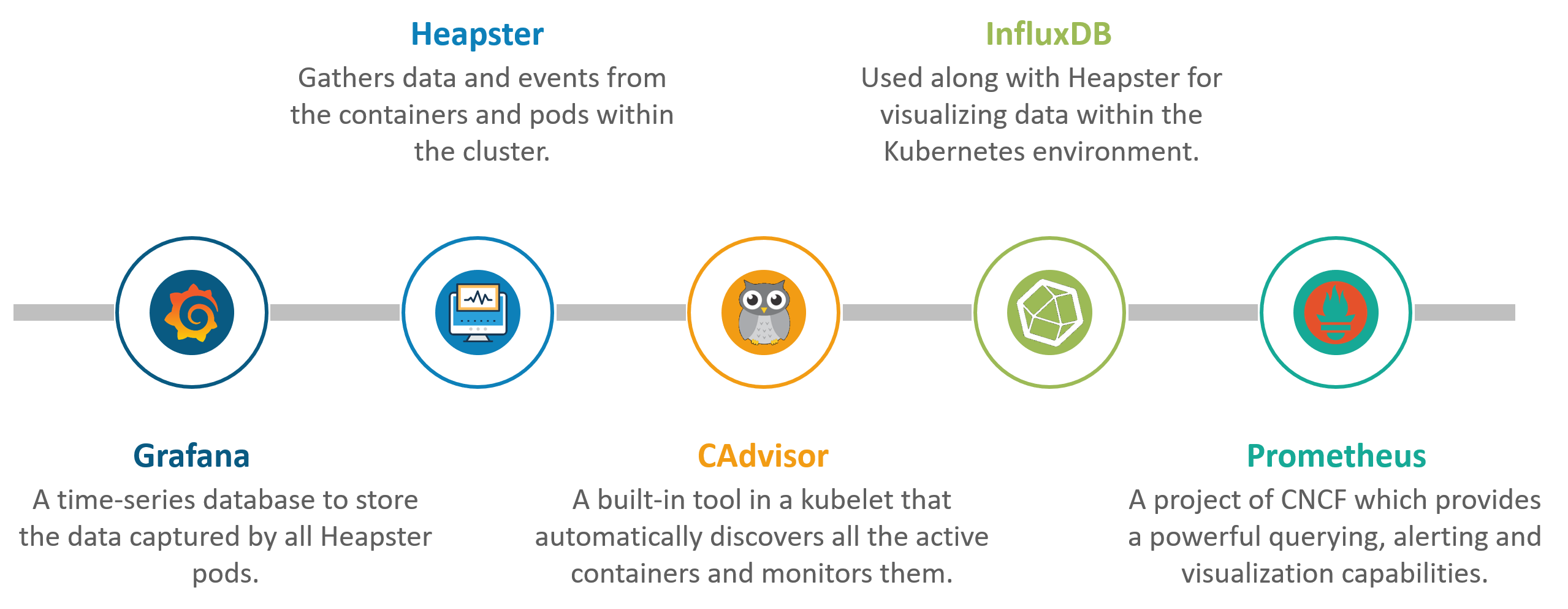


### ****Fig 12:**** Types Of Cloud Controller Manager – Kubernetes Interview Questions

### ****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.

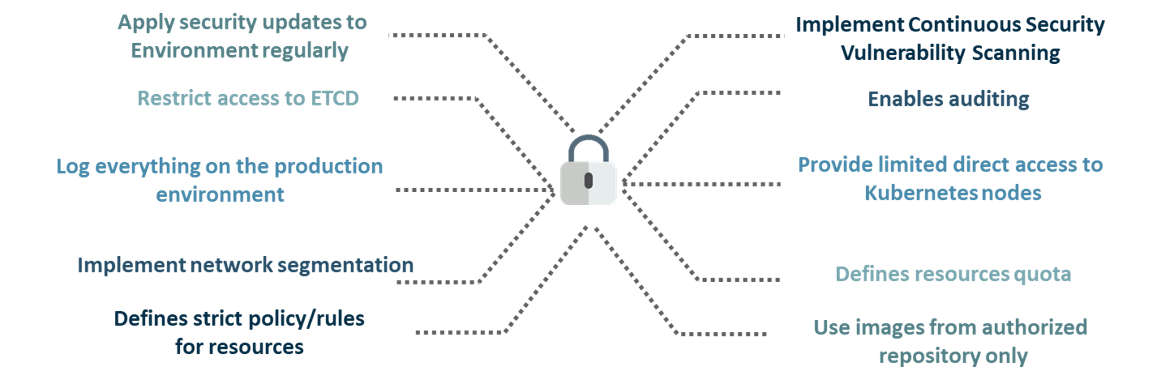
* ****Equity-Based Selectors:****This type of selector allows filtering by label key and values. So, in layman’s terms, the equity-based selector will only look for the pods with the exact same phrase as the label.  
  ****Example****: Suppose your label key says app=nginx; then, with this selector, 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:

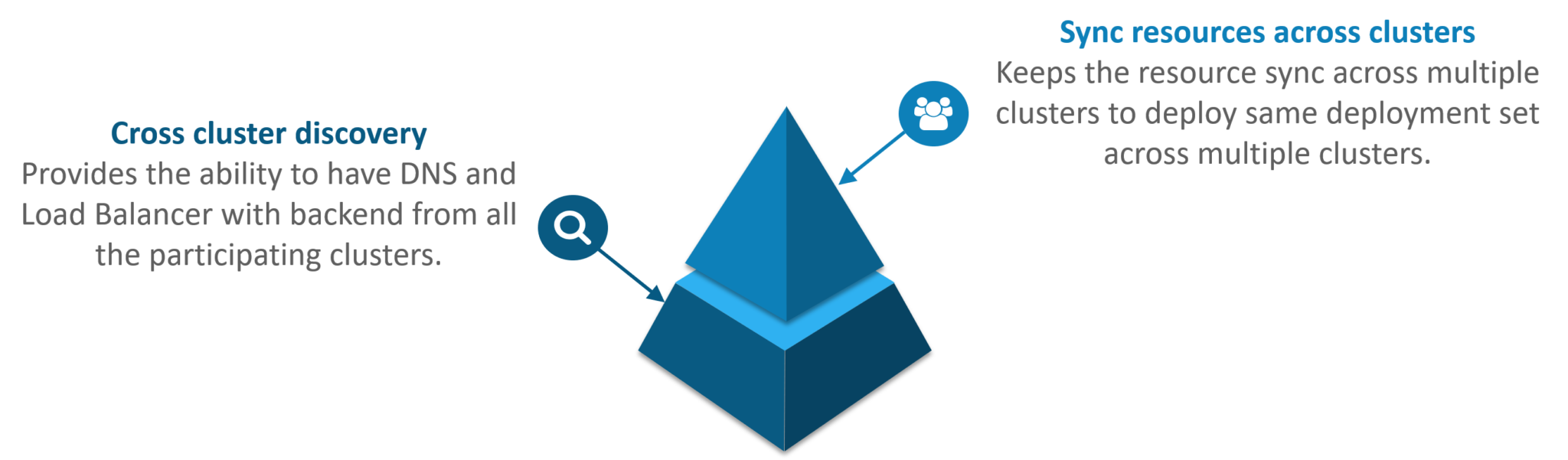


### ****Fig 14:**** Best Security Measures – Kubernetes Interview Questions

### ****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 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.

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

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.

## ****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.

### 1. What is Kubernetes?

This is one of the most basic Kubernetes interview questions yet one of the most important ones! [Kubernetes](https://www.simplilearn.com/tutorials/kubernetes-tutorial/what-is-kubernetes" \o "Kubernetes" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) is an open-source container orchestration tool or system that is used to automate tasks such as the management, monitoring, scaling, and deployment of containerized applications. It is used to easily manage [several containers](https://www.simplilearn.com/top-kubernetes-tools-to-manage-containers-article" \o "several containers" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) (since it can handle grouping of containers), which provides for logical units that can be discovered and managed.

### 2. What are K8s?

K8s is another term for Kubernetes.

### 3. What is orchestration when it comes to software and DevOps?

Orchestration refers to the integration of multiple services that allows them to automate processes or synchronize information in a timely fashion. Say, for example, you have six or seven microservices for an application to run. If you place them in separate containers, this would inevitably create obstacles for communication. Orchestration would help in such a situation by enabling all services in individual containers to work seamlessly to accomplish a single goal.

### 4. How are Kubernetes and Docker related?

This is one of the most frequently asked Kubernetes interview questions, where the interviewer might as well ask you to share your experience working with any of them. [Docker is an open-source](https://www.simplilearn.com/tutorials/docker-tutorial" \o "Docker is an open-source" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) platform used to handle software development. Its main benefit is that it packages the settings and dependencies that the software/application needs to run into a container, which allows for portability and several other advantages. Kubernetes allows for the manual linking and orchestration of several containers, running on multiple hosts that have been created using Docker.

### 5. What are the main differences between the Docker Swarm and Kubernetes?

Docker Swarm is Docker’s native, open-source container orchestration platform that is used to cluster and schedule Docker containers. Swarm differs from Kubernetes in the following ways:

* [Docker Swarm](https://www.simplilearn.com/tutorials/docker-tutorial/docker-swarm" \o "Docker Swarm" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) is more convenient to set up but doesn’t have a robust cluster, while Kubernetes is more complicated to set up but the benefit of having the assurance of a robust cluster
* Docker Swarm can’t do auto-scaling (as can Kubernetes); however, Docker scaling is five times faster than Kubernetes
* Docker Swarm doesn’t have a GUI; Kubernetes has a GUI in the form of a dashboard
* Docker Swarm does automatic load balancing of traffic between containers in a cluster, while Kubernetes requires manual intervention for load balancing such traffic
* Docker requires third-party tools like ELK stack for logging and monitoring, while Kubernetes has integrated tools for the same
* Docker Swarm can share storage volumes with any container easily, while Kubernetes can only share storage volumes with containers in the same pod
* Docker can deploy rolling updates but can’t deploy automatic rollbacks; Kubernetes can deploy rolling updates as well as automatic rollbacks

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

Deploying Applications consist of an architecture that has an operating system. The operating system will have a kernel that holds various libraries installed on the operating system needed for an application.

Whereas container host refers to the system that runs the containerized processes. This kind is isolated from the other applications; therefore, the applications must have the necessary libraries. The binaries are separated from the rest of the system and cannot infringe any other application.

### 7. What are the features of Kubernetes?

* [Kubernetes](https://www.simplilearn.com/tutorials/kubernetes-tutorial/getting-started-with-kubernetes" \o "Kubernetes" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) places control for the user where the server will host the container. It will control how to launch. So, Kubernetes automates various manual processes.
* Kubernetes manages various clusters at the same time.
* It provides various additional services like management of containers, security, networking, and storage.
* Kubernetes self-monitors the health of nodes and containers.
* With Kubernetes, users can scale resources not only vertically but also horizontally that too easily and quickly.

### 8. What are the main components of Kubernetes architecture?

There are two primary components of [Kubernetes Architecture](https://www.simplilearn.com/tutorials/kubernetes-tutorial/kubernetes-architecture" \o "Kubernetes Architecture" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank): the master node and the worker node. Each of these components has individual components in them.

### 9. Explain the working of the master node in Kubernetes?

The master node dignifies the node that controls and manages the set of worker nodes. This kind resembles a cluster in Kubernetes. The nodes are responsible for the cluster management and the API used to configure and manage the resources within the collection. The master nodes of Kubernetes can run with Kubernetes itself, the asset of dedicated pods.

### 10. What is the role of Kube-apiserver?

This kind validates and provides configuration data for the API objects. It includes pods, services, replication controllers. Also, it provides REST operations and also the frontend of the cluster. This frontend cluster state is shared through which all other component interacts.

### 11. What is a node in Kubernetes?

A node is the smallest fundamental unit of computing hardware. It represents a single machine in a cluster, which could be a physical machine in a data center or a virtual machine from a cloud provider. Each machine can substitute any other machine in a Kubernetes cluster. The master in Kubernetes controls the nodes that have containers.

### 12. What does the node status contain?

The main components of a node status are Address, Condition, Capacity, and Info.

### 13. What process runs on Kubernetes Master Node?

The Kube-api server process runs on the master node and serves to scale the deployment of more instances.

### 14. What is a pod in Kubernetes?

In this Kubernetes interview question, try giving a thorough answer instead of a one-liner. [Pods are high-level structures](https://www.simplilearn.com/how-get-started-with-kubernetes-pod-security-policy-article" \o "Pods are high-level structures" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) that wrap one or more containers. This is because containers are not run directly in Kubernetes. Containers in the same pod share a local network and the same resources, allowing them to easily communicate with other containers in the same pod as if they were on the same machine while at the same time maintaining a degree of isolation.

### 15. What is the job of the kube-scheduler?

The kube-scheduler assigns nodes to newly created pods.

### 16. What is a cluster of containers in Kubernetes?

A cluster of containers is a set of machine elements that are nodes. Clusters initiate specific routes so that the containers running on the nodes can communicate with each other. In Kubernetes, the container engine (not the server of the Kubernetes API) provides hosting for the API server.

### 17. What is the Google Container Engine?

The Google Container Engine is an open-source management platform tailor-made for[Docker containers](https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker-container" \o "Docker containers" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank) and clusters to provide support for the clusters that run in Google public cloud services.

### 18. What are Daemon sets?

A Daemon set is a set of pods that runs only once on a host. They are used for host layer attributes like a network or for monitoring a network, which you may not need to run on a host more than once.

### 19. What is ‘Heapster’ in Kubernetes?

In this Kubernetes interview question, the interviewer would expect a thorough explanation. You can explain what it is and also it has been useful to you (if you have used it in your work so far!). A Heapster is a performance monitoring and metrics collection system for data collected by the Kublet. This aggregator is natively supported and runs like any other pod within a Kubernetes cluster, which allows it to discover and query usage data from all nodes within the cluster.

### 20. What is Minikube?

With the help of Minikube, users can Kubernetes locally. This process lets the user run a single-node Kubernetes cluster on your personal computer, including Windows, macOS, and Linus PCs. With this, users can try out Kubernetes also for daily development work.

### 21. What is a Namespace in Kubernetes?

Namespaces are used for dividing cluster resources between multiple users. They are meant for environments where there are many users spread across projects or teams and provide a scope of resources.

### 22. Name the initial namespaces from which Kubernetes starts?

* Default
* Kube – system
* Kube – public

### 23. What is the Kubernetes controller manager?

The controller manager is a daemon that is used for embedding core control loops, garbage collection, and Namespace creation. It enables the running of multiple processes on the master node even though they are compiled to run as a single process.

### 24. What are the types of controller managers?

The primary controller managers that can run on the master node are the endpoints controller, service accounts controller, namespace controller, node controller, token controller, and replication controller.

### 25. What is etcd?

Kubernetes uses etcd as a distributed key-value store for all of its data, including metadata and configuration data, and allows nodes in Kubernetes clusters to read and write data. Although etcd was purposely built for CoreOS, it also works on a variety of operating systems (e.g., Linux, BSB, and OS X) because it is open-source. Etcd represents the state of a cluster at a specific moment in time and is a canonical hub for state management and cluster coordination of a Kubernetes cluster.

### 26. What are the different services within Kubernetes?

Different types of Kubernetes services include:

* Cluster IP service
* Node Port service
* External Name Creation service and
* Load Balancer service

### 27. What is ClusterIP?

The ClusterIP is the default Kubernetes service that provides a service inside a cluster (with no external access) that other apps inside your cluster can access.

### 28. What is NodePort?

The NodePort service is the most fundamental way to get external traffic directly to your service. It opens a specific port on all Nodes and forwards any traffic sent to this port to the service.

### 29. What is the LoadBalancer in Kubernetes?

The LoadBalancer service is used to expose services to the internet. A Network load balancer, for example, creates a single IP address that forwards all traffic to your service.

### 30. What is the Ingress network, and how does it work?

 An ingress is an object that allows users to access your Kubernetes services from outside the Kubernetes cluster. Users can configure the access by creating rules that define which inbound connections reach which services.

How does it work- This is an API object that provides the routing rules to manage the external users' access to the services in the Kubernetes cluster through HTTPS/ HTTP. With this, users can easily set up the rules for routing traffic without creating a bunch of load balancers or exposing each service to the nodes.

### 31. What do you understand by Cloud controller manager?

You must have heard about Public, Private and [hybrid clouds](https://www.simplilearn.com/what-is-hybrid-cloud-article" \o "hybrid clouds" \t "https://www.simplilearn.com/tutorials/kubernetes-tutorial/_blank). With the help of cloud infrastructure technologies, you can run Kubernetes on them. In the context of Cloud Controller Manager, it is the control panel component that embeds the cloud-specific control logic. This process lets you link the cluster into the cloud provider's API and separates the elements that interact with the cloud platform from components that only interact with your cluster.

This also enables the cloud providers to release the features at a different pace compared to the main Kubernetes project. It is structured using a plugin mechanism and allows various cloud providers to integrate their platforms with Kubernetes.

### 32. What is Container resource monitoring?

This refers to the activity that collects the metrics and tracks the health of containerized applications and microservices environments. It helps to improve health and performance and also makes sure that they operate smoothly.

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

A replication controller is referred to as RC in short. It is a wrapper on a pod. This provides additional functionality to the pods, which offers replicas.

It monitors the pods and automatically restarts them if they fail. If the node fails, this controller will respawn all the pods of that node on another node. If the pods die, they won't be spawned again unless wrapped around a replica set.

Replica Set, on the other hand, is referred to as rs in short. It is told as the next-generation replication controller. This kind of support has some selector types and supports the equality-based and the set-based selectors.

It allows filtering by label values and keys. To match the object, they have to satisfy all the specified label constraints.

### 34. What is a headless service?

A headless service is used to interface with service discovery mechanisms without being tied to a ClusterIP, therefore allowing you to directly reach pods without having to access them through a proxy. It is useful when neither load balancing nor a single Service IP is required.

### 35. What are federated clusters?

The aggregation of multiple clusters that treat them as a single logical cluster refers to cluster federation. In this, multiple clusters may be managed as a single cluster. They stay with the assistance of federated groups. Also, users can create various clusters within the data center or cloud and use the federation to control or manage them in one place.

You can perform cluster federation by doing the following:

Cross cluster that provides the ability to have DNS and Load Balancer with backend from the participating clusters.

Users can sync resources across different clusters in order to deploy the same deployment set across the various clusters.

### 36. What is Kubelet?

The kubelet is a service agent that controls and maintains a set of pods by watching for pod specs through the Kubernetes API server. It preserves the pod lifecycle by ensuring that a given set of containers are all running as they should. The kubelet runs on each node and enables the communication between the master and slave nodes.

### 37. What is Kubectl?

Kubectl is a CLI (command-line interface) that is used to run commands against Kubernetes clusters. As such, it controls the Kubernetes cluster manager through different create and manage commands on the Kubernetes component

### 38. Give examples of recommended security measures for Kubernetes.

Examples of standard Kubernetes security measures include defining resource quotas, support for auditing, restriction of etcd access, regular security updates to the environment, network segmentation, definition of strict resource policies, continuous scanning for security vulnerabilities, and using images from authorized repositories.

### 39. What is Kube-proxy?

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

### 40. How can you get a static IP for a Kubernetes load balancer?

A static IP for the Kubernetes load balancer can be achieved by changing DNS records since the Kubernetes Master can assign a new static IP address.

### **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

### **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?**

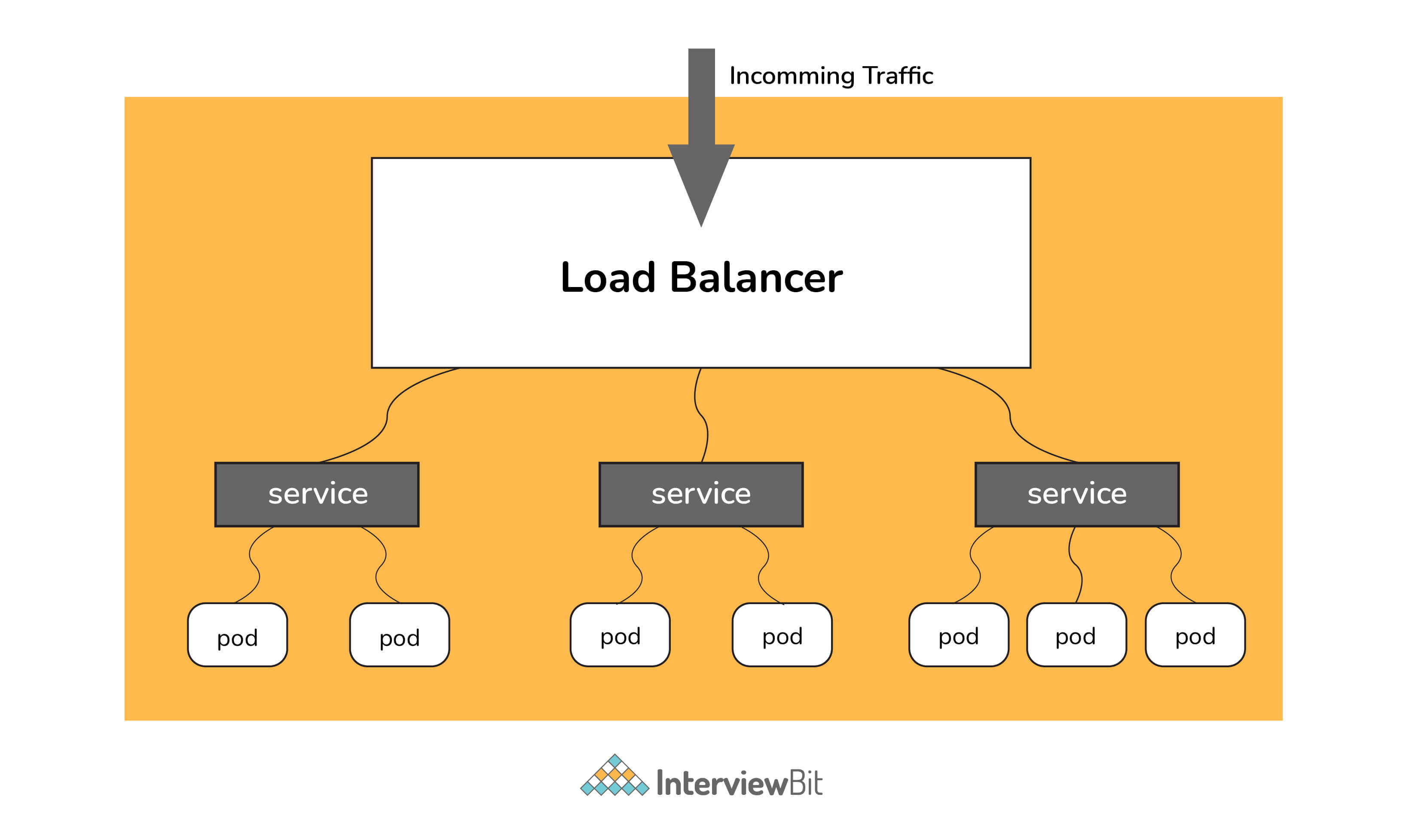
 init containers will set a stage for you before running the actual POD.

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

### **Q1. What is Kubernetes?**

Kubernetes is an open-source container orchestration system for deploying, scaling and managing automated applications. It offers an excellent community and works with all cloud providers. Hence, it is a multi-container management solution.

### **Q2. What is a container?**

Containers are a technology for collecting the compiled code for an application when it is required at run-time. Each container allows you to run repeatable, standard dependencies and the same behavior whenever the container runs. It divides the application from the underlying host infrastructure to make the deployment much easier in cloud or OS platforms.

### **Q3. What are the nodes that run inside the Kubernetes?**

A node is a worker machine or VM depending on the cluster. Each node contains services to run the pods and the pods are managed by the master components.

### **Q4. What are the services that a node gives and its responsibilities?**

The services that include in a node is as follows:

* Container run-time
* Kubelet
* Kube-proxy

The Container run-time is responsible to start and manage the containers. The [kubelet](https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/" \l ":~:text=The kubelet is the primary,object that describes a pod." \t "https://mindmajix.com/_blank) is responsible for running the state of each node and receives commands from the master to work on it and it is also responsible for the metric collection of pods. The Kube-proxy is a component that manages the subnets and makes services available for all other components.

### **Q5. What is a master node in Kubernetes?**

A master node is a node that controls and manages the set of worker nodes and resembles a cluster in Kubernetes.

|  |
| --- |
| If you would like to become a ****Certified Kubernetes Administrator****, then visit Mindmajix - A Global online training platform: “**[Kubernetes Training](https://mindmajix.com/kubernetes-training" \o "Kubernetes Training" \t "https://mindmajix.com/_blank)**".This course will help you to achieve excellence in this domain. |

### **Q6. What are the main components of the master node?**

The main components of the master node that help to manage worker nodes are as follows:

* Kube-server: It acts as a frontend of the cluster and communicates with the cluster through the API server.
* Kube controller: It implements governance across the cluster and runs the set of controllers for the running cluster.
* Kube scheduler: It schedules the activities of the nodes and holds the node resource to determine proper action for triggering events.

### **Q7. What is a pod and what does it do?**

A pod is a group of containers that are deployed together on the same host. It is the basic execution unit of the Kubernetes application that can create or deploy the Kubernetes unit of object models.

[Kubernetes pods](https://kubernetes.io/docs/concepts/workloads/pods/" \t "https://mindmajix.com/_blank) can be used in two ways. they are as follows:

1. Pods that can run in a single container
2. Pods that can run with multiple containers when it is required to work together

### **Q8. What are the different types of multiple-container pods?**

There are three different types of multi-container pods. They are as follows:

* ****Sidecar****: The Sidecar pattern is a single node pattern made of two containers of the application. It contains the core logic of the application and it sends the logic files to the bucket.
* ****Adapter****: It is used to standardize and normalize the output application or monitor data for aggregation. It performs restructuring, reformatting and can write the correct formatted output for the application.
* ****Ambassador****: It is a proxy pattern that allows connecting other containers with a port on the localhost.

### **Q9. What is the Namespace? How many namespaces are there in Kubernetes?**

A namespace is used to work with multiple teams or projects spread across. It is used to divide the cluster resources for multiple users.

### **Q10. Mention different kinds of Namespaces in Kubernetes?**

The namespaces are of three kinds. They are:

1. ****Default****: The default namespace that when the cluster comes out of the box with no other namespaces
2. ****Kube-system:**** The namespace for objects created by Kubernetes.
3. ****Kune-public:**** The namespace that can create automatically and is visible and readable publicly throughout the whole cluster. The public aspect of this namespace is only convenient and reserved for cluster usage.

### **Q11. How are Kubernetes related to docker?**

[Docker](https://mindmajix.com/what-is-docker-how-docker-works" \o "What is Docker?" \t "https://mindmajix.com/_blank) provides the lifecycle management of a container and the docker image builds the run-time of a container. The containers run on multiple hosts through a link and are orchestrated using Kubernetes. Dockers build these containers and help to communicate with multiple hosts through Kubernetes

### **Q12. Mention the difference between Kubernetes and a docker?**

|  |  |  |
| --- | --- | --- |
| ****Features**** | ****Kubernetes**** | ****Docker**** |
| Installation and cluster configuration | The installation process is very complicated but once it has been done, the Cluster is robust. | The installation is very simple, but it does not have a robust cluster. |
| Auto-scaling | It can do Auto-scaling | It cannot do Auto-scaling |
| Data volumes | It can store data only with other containers on the same pod | It can store data on any other containers |
| Logging and monitoring | It is an in-built tool for logging and monitoring | It is a third-party tool, uses ELA stack for logging and monitoring |

### **Q13. Why do we need Container orchestration in Kubernetes?**

Container orchestration is used to communicate with several micro-services that are placed inside a single container of an application to perform various tasks.

The use of container orchestration is as follows:

* It controls and automates various tasks such as deployment, scaling, etc.,
* Reduces the complexity of running time
* Scaling becomes easy
* It is used to deploy and manage complex containerized applications
* Reduces manual setting up services

### **Q14. What are the tools of container orchestration?**

There are many Container orchestration tools that provide a framework for managing microservices and containers at scale. The popular most tools for container orchestration are as follows:

* Kubernetes
* Docker swarm
* Apache Mesos

### **Q15. What are the major operations of Kubelet as a node service component in Kubernetes?**

The major operations that the Kubelet do as follows:

* The Kubelet is a node that communicates with master components to work on all the parts of the Kubernetes cluster.
* It merges the available CPU, memory, and disk of a node into a large Kubernetes cluster.
* It provides access to the controller to check and report the status of the cluster.
* It is responsible for the collection of metric pods

### **Q16. Mention the list of objects of Kubernetes?**

The following is the list of objects used to define the workloads.

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

### **Q17. What is the difference between the pod and the container?**

Pods are the collection of containers used as the unit of replication in Kubernetes. Containers are the set of codes to compile in a pod of the application. Containers can communicate with other containers in the same pod.

### **Q18. Explain Stateful sets in Kubernetes?**

Ans: Stateful set is a workload API object used to manage the stateful application. It is used to manage deployments and scale the sets of pods. The state information and other resilient data of stateful pods were stored and maintained in the disk storage that connects with the stateful set.

### **Q19. How to determine the status of deployment?**

To determine the status of the deployment, use the command below:

kubectl rollout status

If the output runs, then the deployment is successfully completed.

### **Q20. Explain Replication controllers?**

Replication controllers act as supervisors for all long-running pods. It ensures that the specified number of pods are running at the run-time and also ensures that a pod or a set of pods are homogeneous in nature. It maintains the desired number of pods if the number of pods it will terminate the extra pod. And if there is a failed pod, the controller will automatically replace the failed pod.

Visit here to learn [Kubernetes Online Course in Hyderabad](https://mindmajix.com/kubernetes-training-hyderabad" \o "Kubernetes Online Course" \t "https://mindmajix.com/_blank)

### **Q21. What are the features of Kubernetes?**

The features of the Kubernetes are as follows:

* It provides an automated and advanced scheduler to launch the containers on the cluster
* Replacing, rescheduling, and restarting the containers that are failed while compilation
* It supports rollouts and rollback for the desired state of the containerized application
* It can scale up and scale down as per the requirements.

### **Q22. What is kubectl?**

Kubectl is the command-line tool used to control the Kubernetes clusters. It provides the CLI to run the command against clusters to create and manage the Kubernetes components.

### **Q23. What is the Google container engine?**

The Google Container Engine (GKE) is the open-source management for the Docker containers and the clusters. This Kubernetes-based container engine supports only the clusters that run within the Google public cloud service.

### **Q24. What are the different types of services in Kubernetes?**

The different types of services that support Kubernetes are as follows:

* ****Cluster IP:**** It exposes the services on cluster internal IP and makes the services reachable within the cluster only.
* ****Node port:**** It exposes the services on each node’s IP at the static port.
* ****Load balancer:**** It provides services externally using a cloud provider’s load balancer. It creates the service to route the external load balancer automatically.
* ****External name:**** It navigates the service to the contents of the external field by returning the CNAME record by its value.

### **Q25. Mention the various container resource monitoring tools?**

The various container monitoring tools are as follows:

* Grafana
* Heapster
* CAdvisor
* InfluxDB
* Prometheus

### **Q26. What is Heapster?**

Heapster is a performance monitoring and metric collection system. It provides cluster-wide data aggregation by running with a kubelet on each node. It allows for the collection of metrics, pods, workloads, containers, and other signals that are generated by the clusters.

### **Q27. Explain Daemon sets?**

A daemon set ensures that all the eligible nodes run a copy of the pod runs only once in a host. It was created and scheduled by the daemon controller. It is a process that runs in the background and does not produce any visible output.

### **Q28. What are the uses of Daemon sets?**

The uses of Daemon sets are as follows:

* It runs cluster storage such as ceph, glusterd on each node.
* It runs the logs collection of daemons on every node such as fluentd or filebeat.
* It runs node monitoring on every node.

### **Q29. Explain the Replica set?**

A Replica set is used to maintain a stable set of replica pods. It is used to specify the available number of identical pods. It was also considered as a replacement for the replication controller sometimes.

### **Q30. What is ETCD in Kubernetes?**

ETCD is the distributed key-value store. It stores and replicates the configuring data of the Kubernetes cluster.

### **Q31. Explain the Ingress controller?**

An ingress controller is a pod that acts as an inbound traffic handler. It is responsible for reading the ingress resource information and processing the data accordingly.

### **Q32. What is the based selector that is used in the replication controller?**

The Replication controller uses the Equity-Based selector that allows filtering by labels key and values. It only looks for the pods which have the same values as that of the label.

### **Q33. Explain the Load balancer in Kubernetes?**

The load balancer is a way of distributing the loads, which is easy to implement at the dispatch level. Each load balancer sits between the client devices and the backend servers. It receives and distributes the incoming requests to all available servers.

### **Q34. Explain the two different types of load balancers?**

The two different [load balancers](https://mindmajix.com/kubernetes-load-balancer" \o "What is load balancer via Kubernetes?" \t "https://mindmajix.com/_blank) are one is an internal load balancer that balances the load and allocates the pods automatically with the required configuration. And the other is the External load balancer that directs the traffic from external loads to the backend pods.

### **Q35. What is Minikube?**

Minikube is a type of tool that helps to run the Kubernetes locally. It runs on a single-node Kubernetes cluster inside a Virtual machine (VM).

### **Q36. What are the uses of Google Kubernetes Engine?**

The uses of Google Kubernetes Engine are as follows:

* It creates the Docker container cluster
* It resizes the application controllers
* It creates the containers pods, load balancer, services, replication controller
* It updates and upgrades the container cluster
* It helps to debug the container cluster

### **Q37. Explain Prometheus in Kubernetes?**

Prometheus is an open-source toolkit that is used for metric-based monitoring and alerting the application. It provides a data model and a query language and can provide details and actions of metrics. It supports the instrumental application of language for many languages. The Prometheus operator provides easy monitoring for deployments and k8s services, besides Alertmanager and Grafana.

### **Q38. What is the role of clusters in Kubernetes?**

Kubernetes allows the required state management by cluster services of a specified configuration. These cluster services run the configurations in the infrastructure. The following are the steps that are involved in this process as follows:

* The deployment file contains all the configuration that is fed into the cluster
* These deployments are fed into the API server
* The cluster services will schedule the pods in the environment
* It also ensures the right number of pods were running

### **Q39. What is the Cluster IP?**

The cluster Ip is a default Kubernetes service that provides a link between the pods or map container port and the host ports. It provides the services within the cluster and gives access to other apps which are inside the same cluster.

### **Q40. What are the types of controller managers?**

The Different types of controller managers that can run on the master node are as follows:

* Endpoints controller
* Namespace controller
* Service account controller
* Replication controller
* Node controller
* Token controller

### **Q41. What is the Kubernetes architecture?**

The [Kubernetes architecture](https://mindmajix.com/kubernetes-architecture" \o "Kubernetes architecture" \t "https://mindmajix.com/_blank) provides a flexible, coupled mechanism for the service. It consists of one master node and multiple containers. The master node is responsible for managing the clusters, API, and scheduling the pods. Each node runs on the container runtime such as Docker, rkt along with the node that communicates with the master.

### **Q42. What are the main components of Kubernetes architecture?**

The two main components of the Kubernetes architecture are as follows:

* Master node
* Worker node

Each node contains the individual components in it

### **Q43. Define Kube-api server?**

The Kube-API is the frontend of the master node that exposes all the components in the API server. It provides communication between the Kubernetes nodes and the master components.

### **Q44. What are the advantages of Kubernetes?**

The advantages of the Kubernetes are as follows:

* Kubernetes is open-source and free
* It is highly scalable and runs in any operating system
* It provides more concept and is more powerful than Docker swarm
* It provides scheduler, auto-scaling, rolling upgrades, and health checks
* It has a flat network space and customized functionalities
* It is easy to make effective CI/CD pipelines
* It can improve productivity

### **Q45. What are the disadvantages of Kubernetes?**

The disadvantages of the Kubernetes are as follows:

* The installation process and configuration is highly difficult
* It is not easy to manage the services
* It takes a lot of time to run and compile
* It is more expensive than the other alternatives
* It can be overkill for simple applications