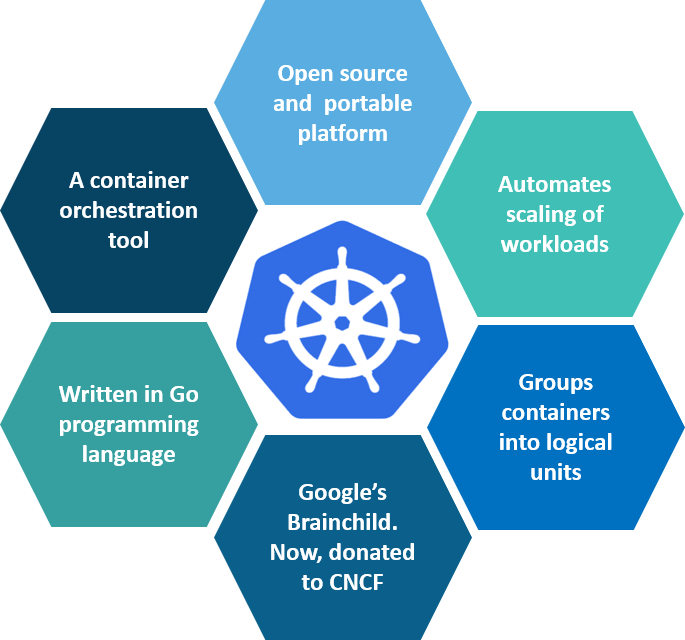
**Basic Kubernetes Interview Questions**

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. | 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?**



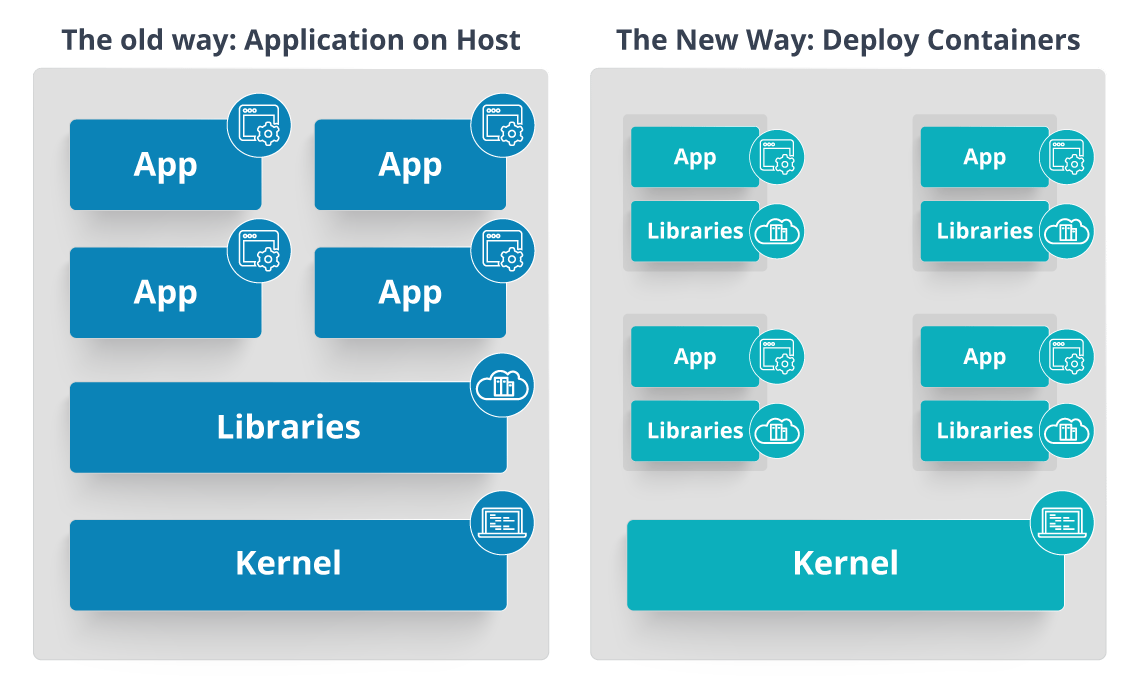
**Fig 1:** What is Kubernetes – Kubernetes Interview Questions

Kubernetes is an open-source container management tool which holds the responsibilities of container deployment, scaling & descaling of containers & load balancing. Being the 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 which 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 which needs Java then that particular application we’ll get access to Java and if there’s another application which needs Python then only that particular application will have access to Python.

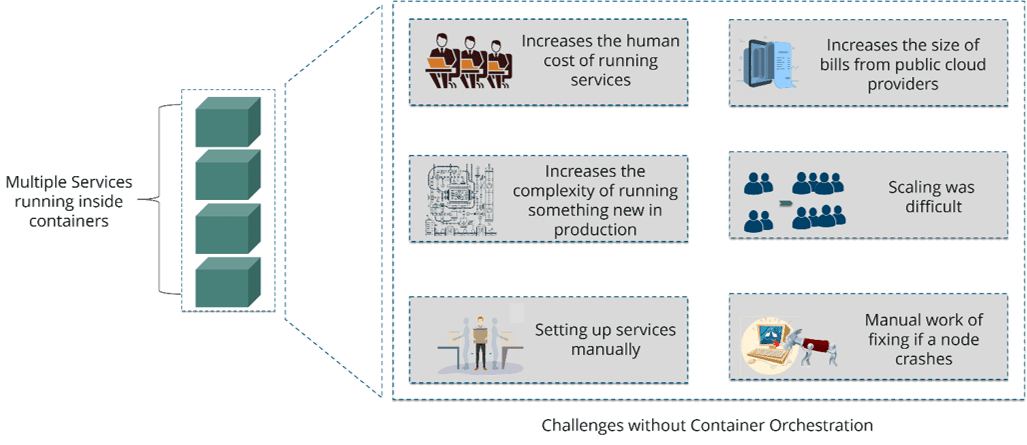
The individual blocks that you can see on the right side of the diagram are basically containerized and these are isolated from other applications. So, the applications have the necessary libraries and binaries isolated from the rest of the system, and cannot be encroached by any other application.

**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.

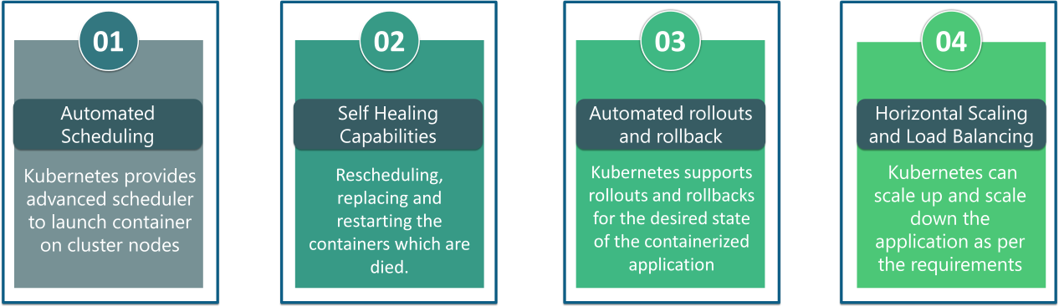


**Fig 3:** Challenges Without Container Orchestration – Kubernetes Interview Questions

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:



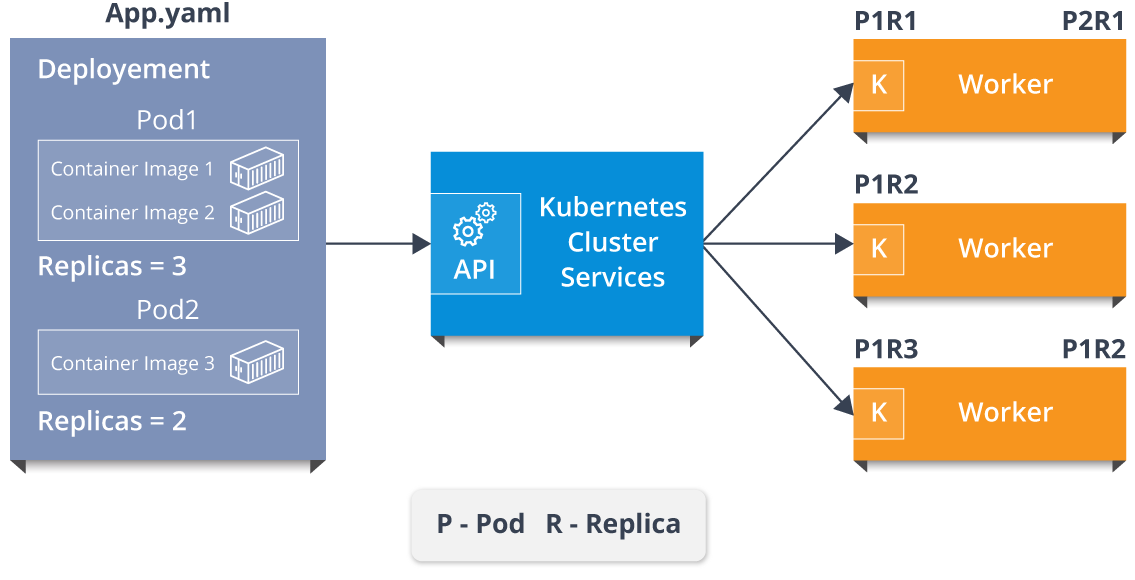
**Fig 4:** Features Of Kubernetes – Kubernetes Interview Questions

**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 the clusters. This Kubernetes based engine supports only those clusters which run within the Google’s public cloud services.

**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?**

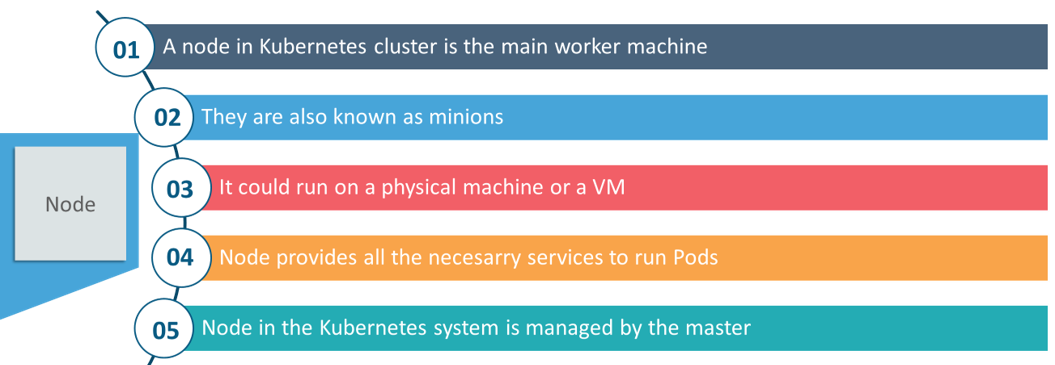
Minikube is a tool that makes it easy to run Kubernetes locally. This runs a single-node Kubernetes cluster inside a virtual machine.

**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?**  


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

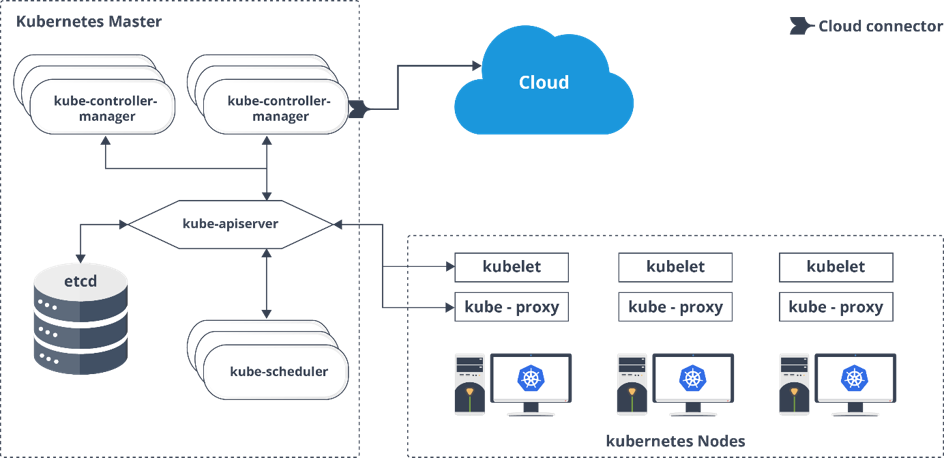
Interested in learning Kubernetes?[Learn Now](https://www.edureka.co/kubernetes-certification)

**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 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.



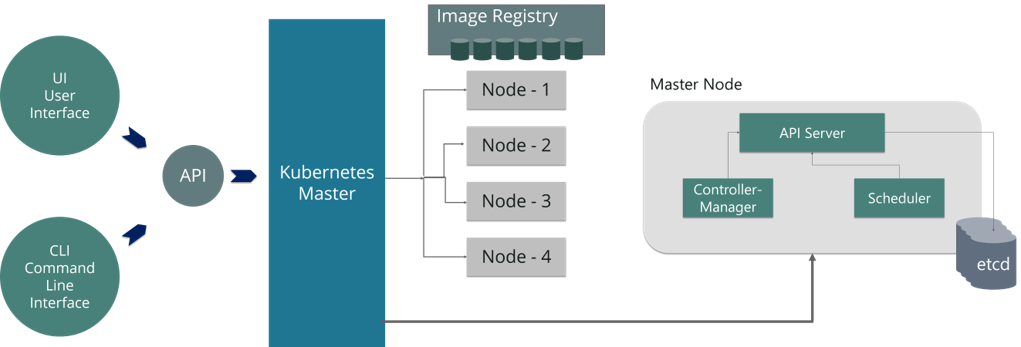
**Fig 7:** Architecture Of Kubernetes – Kubernetes Interview Questions

**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 which 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)](https://www.edureka.co/kubernetes-certification" \t "_blank)

**[Kubernetes Certification Training](https://www.edureka.co/kubernetes-certification" \t "_blank)**

* *[Instructor-led Sessions](https://www.edureka.co/kubernetes-certification" \t "_blank)*
* *[Real-life Case Studies](https://www.edureka.co/kubernetes-certification" \t "_blank)*
* *[Assignments](https://www.edureka.co/kubernetes-certification" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/kubernetes-certification" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/kubernetes-certification" \t "_blank)

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

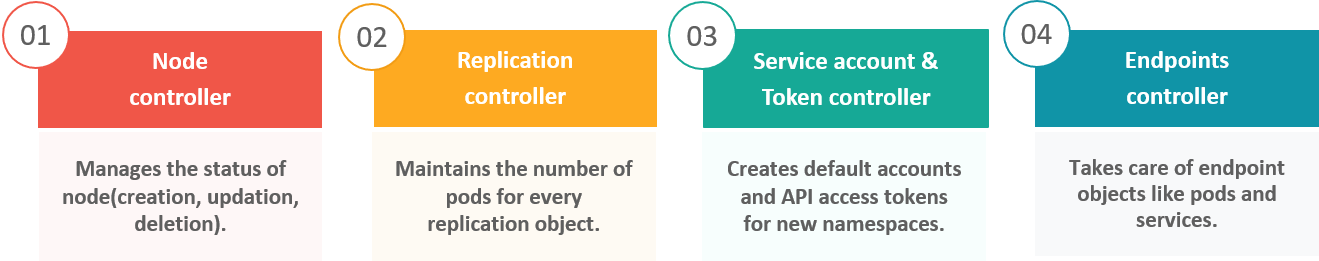
**Q4.  What is the role of kube-apiserver and kube-scheduler?**

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 distribution and management of workload on the worker nodes. So, it selects the most suitable node to run the unscheduled pod based on resource requirement and keeps a track of resource utilization. It makes sure that the workload is not scheduled on nodes which are already full.

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

Multiple controller processes run on the master node but are compiled together to run as a single process which is 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 :  


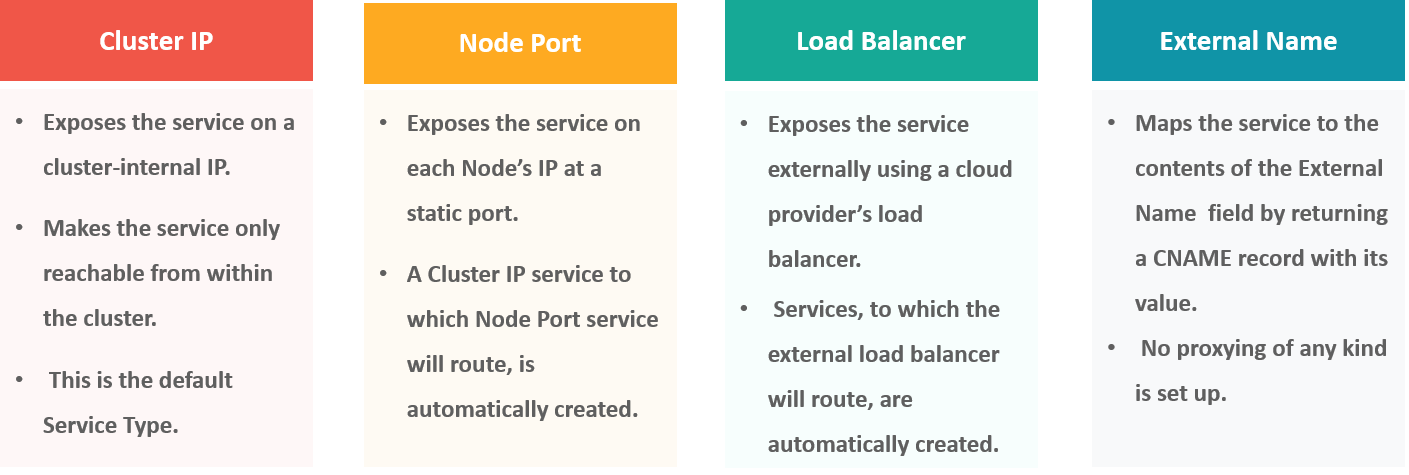
**Fig 9:** Types Of Controllers – Kubernetes Interview Questions

**Q6.  What is ETCD?**

Etcd is written in Go programming language and is a distributed key-value store used for coordinating between 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.

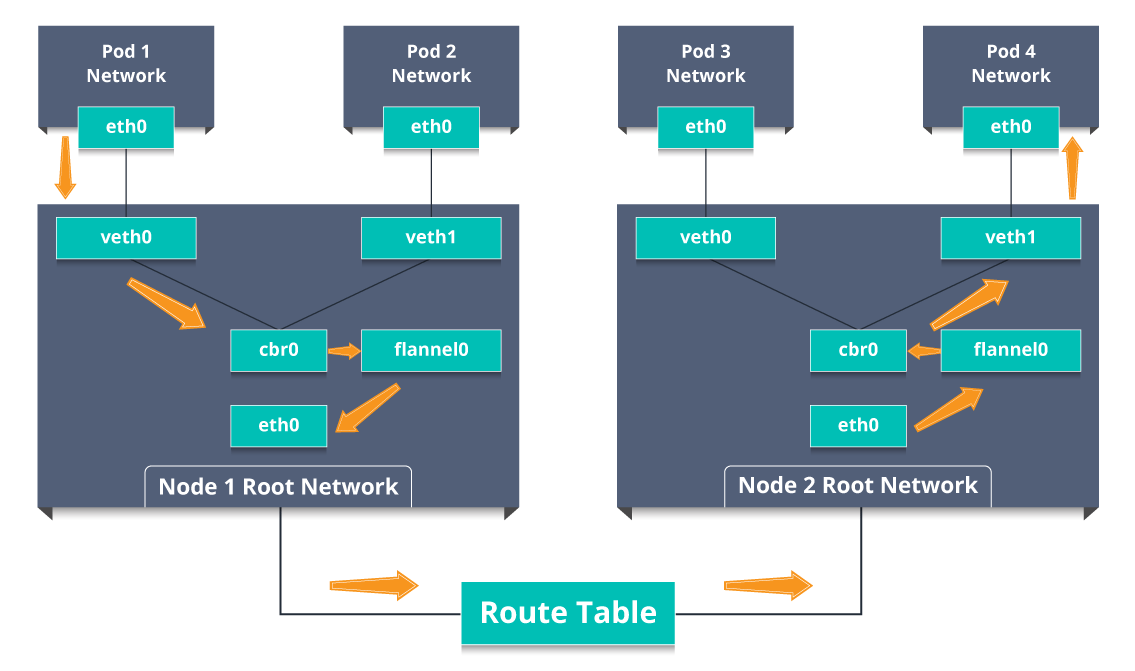
**Q9. What is Ingress network, and how does it work?**

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.



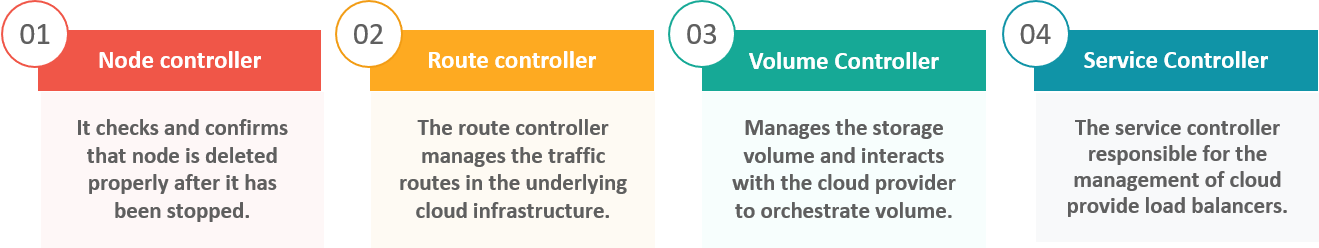
**Fig 11:** Working Of Ingress Network – Kubernetes Interview Questions

* 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:

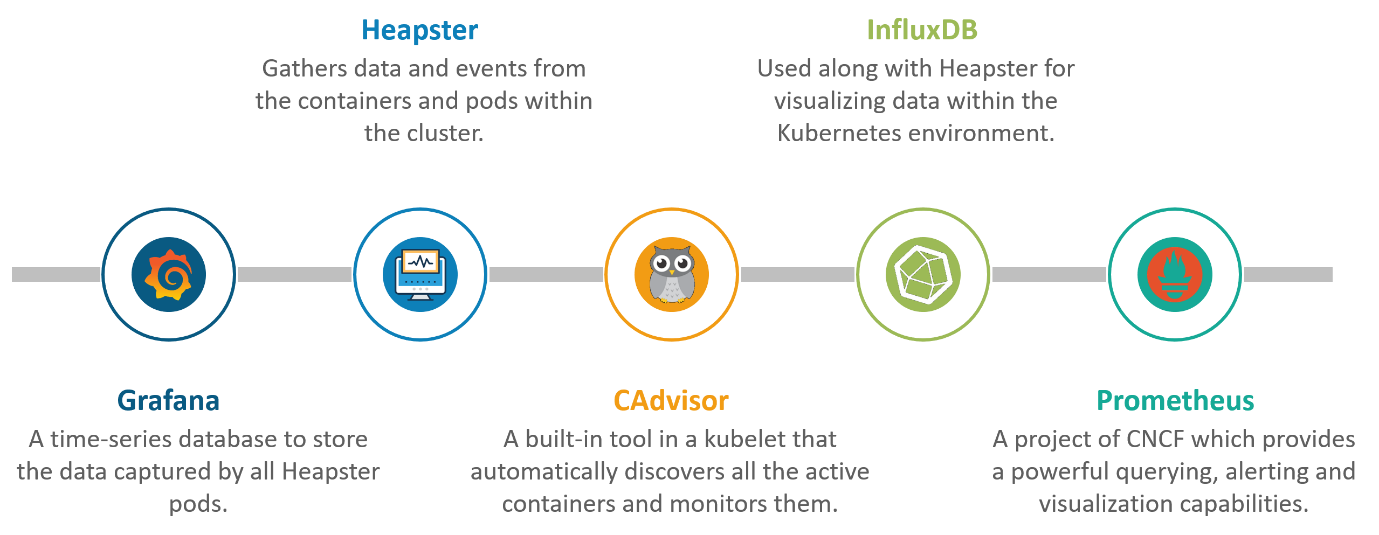


**Fig 12:** Types Of Cloud Controller Manager – Kubernetes Interview Questions

**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 replication controller?**

Replica Set and Replication Controller do almost the same thing. Both of them 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 use Set-Based selectors while replication controllers use Equity-Based selectors.

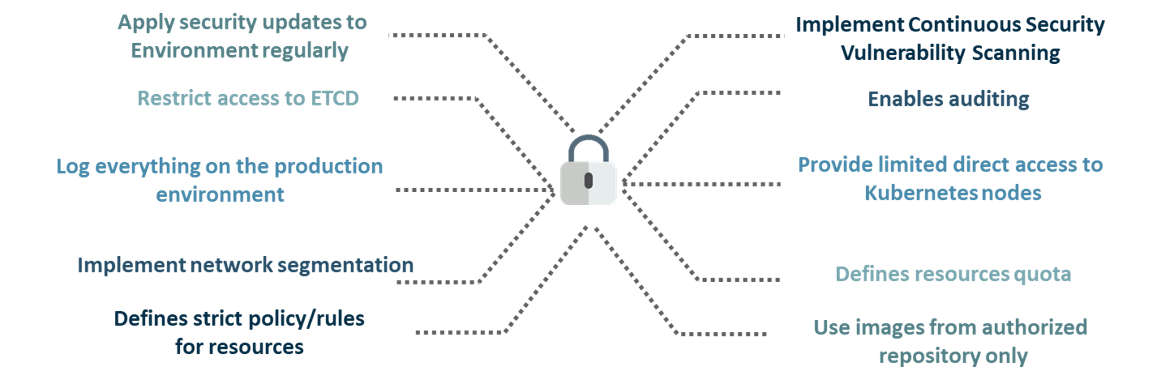
* **Equity-Based Selectors:**This type of selector allows filtering by label key and values. So, in layman terms, the equity-based selector will only look for the pods which will have the exact same phrase as that of the label.  
  **Example**: Suppose your label key says app=nginx, then, with this selector,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

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

Headless Service is similar to that of a ‘Normal’ services but does not have a Cluster IP. This service enables you to directly reach the pods without the need of accessing it 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:

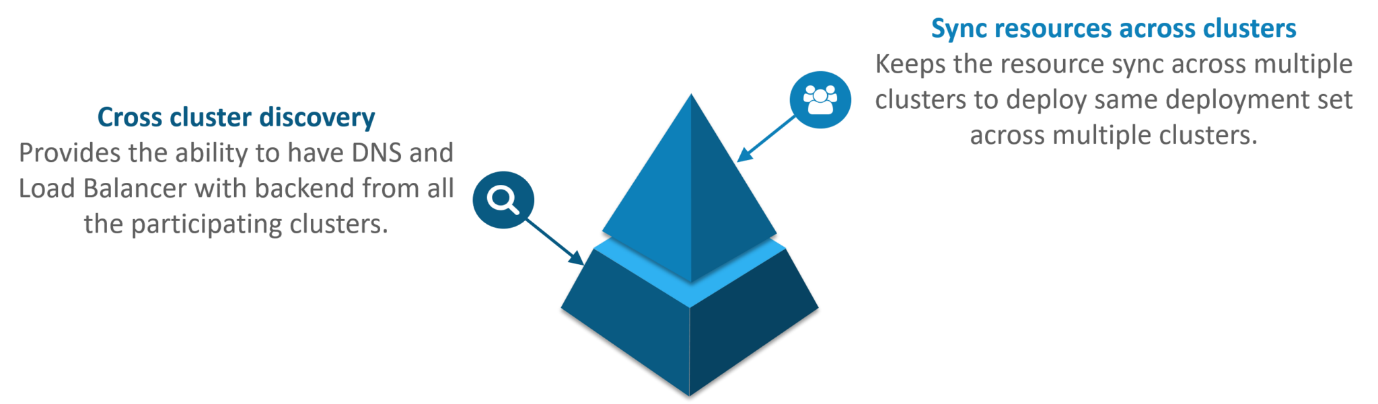


**Fig 14:** Best Security Measures – Kubernetes Interview Questions

**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.

[[](https://www.edureka.co/kubernetes-certification)](https://www.edureka.co/kubernetes-certification" \t "_blank)

**[Kubernetes Certification Training](https://www.edureka.co/kubernetes-certification" \t "_blank)**

[Watch The Course Preview](https://www.edureka.co/kubernetes-certification" \t "_blank)

**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.

*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)

**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 Interview Questions**

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

[[](https://www.edureka.co/kubernetes-certification)](https://www.edureka.co/kubernetes-certification" \t "_blank)

**[Kubernetes Certification Training](https://www.edureka.co/kubernetes-certification" \t "_blank)**

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/kubernetes-certification" \t "_blank)

**Q1. What are minions in 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?**

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

**1. What is Kubernetes?**

First let us compare Kubernetes with Docker Swarm

|  |  |  |
| --- | --- | --- |
| **Comparison** | **Kubernetes** | **Docker Swarm** |
| Controller | Master | Manager |
| Slave | Nodes | Worker |
| Deployment Unit | Pod | Task |
| Load balancing | Service | Ingress |

Kubernetes is a container orchestration tool that is used for automating the managing, monitoring, scaling and deploying of containerized applications. It creates groups of containers that can be logical discovered and managed for easy operations on containers.

**2. What are the benefits of Kubernetes?**

With container orchestration tool Kubernetes, it becomes extremely easy to handle the containers. You can respond to customer demands by deploying the applications faster and in a more predictable manner.

Here we will list some of the benefits of Kubernetes:

* Automatic scheduling
* Automated rollback
* Horizontal scaling
* Auto healing capabilities.

Become DevOps Certified in 16 hrs.

**GET CERTIFIED**

**3. What is the difference between Kubernetes and Docker Swarm?**

Docker Swarm is the default container orchestration tool that comes with Docker. Docker Swarm can only orchestrate simple Docker containers. Kubernetes on the other hand helps to manage much more complex software application containers. Kubernetes offers support for larger demand production environment.

**4. What is orchestration in software?**

Application orchestration in software process means you can integrate two or more applications. You will be able to automate arrangement, coordination and management of computer software. The goal of any orchestration process is to streamline and optimize frequent repeatable processes.

**5. What is a Kubernetes namespace?**

The Kubernetes namespace is used in the environment wherein you have multiple users spread in geographically vast areas and working on multiple projects. What the namespace does is divide the cluster resources between multiple users.

**6. What is a Pod in Kubernetes?**

You can think of Kubernetes Pod as a group of containers that are run on the same host. So if you regularly deploy single containers then your container and Pod will be one and the same.

**7. What is a node in Kubernetes?**

A node in Kubernetes is a worker machine which is also known as a minion. This node could be a physical machine or a virtual machine. For each of the node there is a service to run pods and it is managed by master components. The node services could include kubelet, kube-proxy and so on.

**8. What is a Heapster?**

The Heapster lets you do the container cluster monitoring.It lets you do cluster-wide monitoring and event data aggregation. It has native support for Kubernetes.

**9. What is a container cluster?**

A container cluster lets you place and manage the containers in a dynamic setup. It can be considered as a set of nodes or Compute Engine instances. The API server of the Kubernetes does not run on cluster nodes but instead the Container Engine hosts the API server.

**10. What is the Kubelet?**

You can think of Kubelet as the lowest level component in a Kubernetes. The Kubelet is responsible for making the individual machines to run. The sole purpose of a Kubelet is given a set of containers, it has to ensure that they are all running.

[**Interested in Learning DevOps? Click Here**](https://intellipaat.com/devops-certification-training/#course-content)

**11. What is Minikube?**

The Minikube makes it easy for the local running of Kubernetes. Within a Virtual Machine, the Minikube runs a single-node Kubernetes cluster.

**12. What is Kubectl?**

The Kubectl is a Kubernetes command line tool that is used for deploying and managing applications on Kubernetes. The Kubectl is especially useful for inspecting the cluster resources, for creating, updating and deleting the components.

**13. What is the Gke?**

The Gke is a Google Kubernetes Engine which is used for managing and orchestrating systems for Docker containers. The Gke also lets you orchestrate container clusters within the Google Public Cloud as well.

**14. What is Kube proxy?**

The Kube proxy runs on each of the node. It can do simple tasks like TCP, UDP Forwarding and so on. It shows the services in the Kubernetes API on each node.

**15. What are the Kubernetes Master components?**

The components of the Kubernetes Master include the API server, the controller manager, Scheduler, and the etc components. The Kubernetes Master components are responsible for running and managing the Kubernetes cluster.

**16. What is the use of kube-controller-manager?**

It is the Kubernetes controller manager. The kube-controller-manager is a daemon that embeds the core control loops which regulates system state and it is a non-terminating loop.

**17. What is load-balancing on Kubernetes?**

The process of load-balancing will let you expose the services. There are two types of load-balancing when it comes to Kubernetes.

Internal load balancing: This is used for balancing the loads automatically and allocating the pods with the required configuration.

External load balancing: This directs the traffic from external loads to the backend pods.

**18. what does a Kube-scheduler do?**

The Kube-scheduler has to job of assigning the nodes to the newly created pods.

**19. Where is the Kubernetes cluster data stored?**

The primary data store of Kubernetes is etcd which is responsible for all Kubernetes cluster data store.

**20. How to set a static IP for Kubernetes load balancer?**

Kubernetes Master assigns a new IP address.

You can set a static IP for Kubernetes load balancer by changing the DNS records every time the

### Kubernetes Interview Questions And Answers

|  |  |
| --- | --- |
| **Kubernetes Interview Questions** | |
| Kubernetes is an | Open source software |
| Kubernetes is a | System for automating deployment, scaling and management of containerized applications |
| Kubernetes was | Originally designed by Google and now maintained by the Cloud Native Computing Foundation. |
| Kubernetes can | It aims to provide a platform for automating deployment, scaling, and operations of application containers. |
| Kubernetes developed by | Google |
| Kubernetes License | Apache License 2.0 |
| Kubernetes has written in | Go Programming |

**Kubernetes Interview Question # 1) What is the Kubernetes?**

**A) Kubernetes** is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery.

**Kubernetes Interview Question # 2) What is Kubernetes and how to use it?**

**A) Kubernetes** is an open-source platform designed to automate deploying, scaling, and operating **application** containers. With **Kubernetes**, you are able to quickly and efficiently respond to customer demand: Deploy your applications quickly and predictably.

**Kubernetes Interview Question # 3) What is the meaning of Kubernetes?**

**A) Kubernetes** (commonly referred to as “K8s”) is an open-source system for automating deployment, scaling and management of containerized applications that was originally designed by Google and donated to the Cloud Native Computing Foundation.

### Docker Kubernetes Interview Questions For Experienced

**Kubernetes Interview Question # 4) What is a docker?**

**A) Docker container** is an open source software development platform. Its main benefit is to package applications in “**containers**,” allowing them to be portable among any system running the Linux operating system (OS).

**Kubernetes Interview Question # 5) What is orchestration in software?**

A) Application **Orchestration**. Application or service **orchestration** is the process of integrating two or more applications and/or services together to automate a process, or synchronize data in real-time. Often, point-to-point integration may be used as the path of least resistance.

**Kubernetes Questions # 6) What is a cluster in Kubernetes?**

A) These master and node machines run the **Kubernetes cluster** orchestration system. A container **cluster** is the foundation of Container Engine: the **Kubernetes**objects that represent your containerized applications all run on top of a **cluster**.

**Interview Questions on Kubernetes # 7) What is a swarm in Docker?**

**A) Docker Swarm** is a clustering and scheduling tool for **Docker** containers. With **Swarm**, IT administrators and developers can establish and manage a cluster of**Docker** nodes as a single virtual system.

**Kubernetes Openshift Interview Question # 8) What is Openshift?**

**A) OpenShift** Online is Red Hat’s public cloud application development and hosting platform that automates the provisioning, management and scaling of applications so that you can focus on writing the code for your business, startup, or big idea.

### Advanced Kubernetes Interview Questions

**Docker and Kubernetes Interview Question # 9) What is a namespace in Kubernetes?**

**A) Namespaces** are intended for use in environments with many users spread across multiple teams, or projects. **Namespaces** are a way to divide cluster resources between multiple uses (via resource quota). In future versions of **Kubernetes**, objects in the same **namespace** will have the same access control policies by default.

**Kubernetes Interview Question # 10) What is a node in Kubernetes?**

A) A **node** is a worker machine in **Kubernetes**, previously known as a minion. A **node**may be a VM or physical machine, depending on the cluster. Each **node** has the services necessary to run pods and is managed by the master components. The services on a **node** include [Docker](https://codingcompiler.com/docker-interview-questions-answers/), kubelet and kube-proxy.

**Kubernetes Interview Question # 11) What is Docker and what does it do?**

**A) Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

**Kubernetes Interview Question # 12) What is a**Heapster**?**

**A) Heapster** is a cluster-wide aggregator of monitoring and event data. It supports Kubernetes natively and works on all Kubernetes setups, including our Deis Workflow setup.

**Kubernetes Interview Question # 13) Why do we use Docker?**

**A) Docker** provides this same capability without the overhead of a virtual machine. It lets you put your environment and configuration into code and deploy it. The same **Docker** configuration can also be used in a variety of environments. This decouples infrastructure requirements from the application environment.

**Kubernetes Interview Question # 14) What is a docker in**cloud**?**

A) A node is an individual Linux host used to deploy and run your applications. **Docker Cloud** does not provide hosting services, so all of your applications, services, and containers run on your own hosts. Your hosts can come from several different sources, including physical servers, virtual machines or cloud providers.

**Kubernetes Interview Question # 15) What is a cluster of containers?**

A) A **container cluster** is a set of Compute Engine instances called nodes. It also creates routes for the nodes, so that containers running on the nodes can communicate with each other. The **Kubernetes API** server does not run on your cluster nodes. Instead, Container Engine hosts the API server.

### Real-Time Kubernetes Scenario Based Interview Questions

**Kubernetes Interview Questions # 16) What is the**Kubelet**?**

A) Kubelets run pods. The unit of execution that **Kubernetes** works with is the pod. A pod is a collection of containers that share some resources: they have a single IP, and can share volumes.

**Kubernetes Interview Questions # 17) What is Minikube?**

A) Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a VM on your laptop for users looking to try out Kubernetes or develop with it day-to-day.

**Kubernetes Interview Questions # 18) What is**Kubectl**?**

A) kubectl is a command line interface for running commands against **Kubernetes clusters**. This overview covers **kubectl** syntax, describes the command operations, and provides common examples. For details about each command, including all the supported flags and subcommands, see the kubectl reference documentation.

**Kubernetes Interview Questions # 19) What is the**Gke**?**

A) **Google Container Engine (GKE)** is a management and orchestration system for Docker container and container clusters that run within Google’s public cloud services. Google Container Engine is based on Kubernetes, Google’s open source container management system.

**Kubernetes Interview Questions # 20) What is k8s?**

A) Kubernetes, also sometimes called **K8S** (K – eight characters – S), is an open source orchestration framework for containerized applications that was born from the Google data centers.

**Kubernetes Interview Questions # 21) What is KUBE proxy?**

A) Synopsis. The **Kubernetes** network proxy runs on each node. Service cluster ips and ports are currently found through Docker-links-compatible environment variables specifying ports opened by the service proxy. There is an optional addon that provides cluster DNS for these cluster IPs.

**Kubernetes Interview Questions # 22) Which process runs on Kubernetes master node?**

A) Kube-apiserver process runs on Kubernetes master node.

**Kubernetes Interview Questions # 23) Which process runs on Kubernetes non-master node?**

A) Kube-proxy process runs on Kubernetes non-master node.

**Kubernetes Interview Questions # 24) Which process validates and configures data for the**api**objects like pods, services?**

A) kube-apiserver process validates and configures data for the api objects.

**Kubernetes Interview Questions # 25) What is the use of kube-controller-manager?**

A) kube-controller-manager embeds the core control loop which is a non-terminating loop that regulates the state of the system.

**Kubernetes Interview Questions # 26) Kubernetes objects made up of what?**

A) Kubernetes objects are made up of Pod, Service and Volume.

**Kubernetes Interview Questions # 27) What are Kubernetes controllers?**

A) Kubernetes controllers are Replicaset, Deployment controller.

**Kubernetes Interview Questions # 28) Where Kubernetes cluster data is stored?**

A) etcd is responsible for storing Kubernetes cluster data.

**Kubernetes Interview Questions # 29) What is the role of**kube**-scheduler?**

A) kube-scheduler is responsible for assigning a node to newly created pods.

**Kubernetes Interview Questions # 30) Which container runtimes supported by Kubernetes?**

A) Kubernetes supports docker and rkt container runtimes.

**Kubernetes Interview Questions # 31) What are the components interact with Kubernetes node interface?**

A) Kubectl, Kubelet, and Node Controller components interacts with Kubernetes node interface.

[1. What are pods in Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled1)

A Kubernetes pod is a group of containers that are being deployed in the same host. Pods have the capacity to operate one level higher than the individual containers. This is because pods have the group of containers that work together to produce an artefact or to process a set of work.

[2. What are nodes in kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled2)

A node is a type of work machine in Kubernetes that was previously known as a minion. A node can be a type of virtual machine or the physical machine. It always depends upon the clusters. Each of the nodes provides the services that are necessary to run pods, and it is also managed by the master components.

[3. What is heapster in Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled3)

Heapster is a type of cluster-wide aggregator that helps in the process of monitoring and event data. Heapster helps to enable the container cluster monitoring and performance analysis for Kubernetes.

[4. What is Kubernetes Load Balancing?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled4)

Load Balancing is one of the most common and the standard ways of exposing the services.

There are two types of load balancing in Kubernetes and they are:

1. Internal load balancer – This type of balancer automatically balances loads and allocates the pods with the required configuration.
2. External Load Balancer – This type of balancer directs the traffic from the external loads to backend pods.

[5. What does the nodes status contains?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled5)

The followings are the main components that the node status:

* Address
* Condition
* Capacity
* Info

[6. What are the initial namespaces from which the Kubernetes starts?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled6)

The followings are the three initial namespaces from which the Kubernetes starts:

* Default
* Kube – system
* Kube – public

[7. What are the main components of the Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled7)

The following are the main components of the Kubernetes:

* API server
* Scheduler
* Controller manager
* .etcd
* .Addons

[8. What do you mean by Kubelet?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled8)

Kubelet is a type of primary node agents that especially runs on each node. Kubelet only works on the descriptions that the containers provide to the Podspec. Kubelet also makes sure that the container described in Podspec is healthy and running.

[9. What are namespaces in Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled9)

Kubernetes is especially intended for the use of the environments with many other users that are being spread across multiple teams or projects. Namespaces are the way to divide the cluster resources between the multiple users.

[10. What are the different types of services being provided by Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled10)

The followings are the different types of services being provided by the Kubernetes:

* Cluster IP
* Node Port
* Load Balancer
* External name

[11. What is GKE in Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled11)

Firstly GKE stands for Google Kubernetes Engine. GKE is a management and an orchestration system that is used for Docker container and all the container clusters that basically run within the Google’s public cloud services. Google Kubernetes engine is based on Kubernetes.

[12. What are the uses of Google Kubernetes Engine?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled12)

The followings are the uses of the Google Kubernetes Engine:

* Create or resize Docker container clusters
* Creates container pods, replication controller, jobs, services or load balancer
* Resize application controllers
* Update and upgrade container clusters
* Debug container clusters.

[13. What do you understand by Kubernetes?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled13)

Kubernetes is basically a type of an open – source container. Kubernetes has the potential to hold the container deployment, scaling and descaling of the container and load balancing. Kubernetes was being developed in the year of 2014. It is also used to manage the Linux containers across the privates, hybrid and cloud environments.

[14. What are the features of Minikube?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled14)

The followings are the main features of the Minikube:

* DNS
* Nodeports
* Configure maps and secrets
* Dashboards
* Enabling CNI
* Ingress
* Container runtime: Docker, rkt, CRI – O and containerd

[15. What are the difference between Kubernetes and Docker Swarm?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/" \l "collapseUnfiled15)

The followings are the main difference between Kubernetes and docker and they are:

* The installation structure of the Kubernetes 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 do the process of the auto scaling but the Docker swarm cannot do the process of the auto scaling.
* Kubernetes is highly scalable and also scales fast. But the Docker swarm scales are 5x faster than Kubernetes and is also highly scalable.

Pros of the Kubernetes

The following are the pros of the Kuberenetes:

* Kubernetes is open source and free
* It is highly scalable
* Kubernetes makes it a lot easier to establish effective CI/CD Pipelines
* It is less tightly bound to particular sets of tools
* It provides scheduler, rolling upgrades, health checks and autoscaling
* Kubernetes runs best in any of the operating systems
* Kubernetes provides more concepts and is more powerful than Docker swarm
* It has a flat networking space
* It also has a customizable functionality
* It is comprehensive and detailed documented

Cons of the Kubernetes

Some of the cons of the Kubernetes are as follows:

* Kubernetes is not simple to manage the services
* It is really difficult to install and configure
* It takes time to get up and run
* No placements are still available yet

Main features of the Kubernetes

The following are the main features of the Kubernetes:

1. Kubernetes automates various manual processes. They will automatically control the server for you and also help you host the container. It also controls how to Kubernetes will be launched.
2. Help to interact with certain groups of the container. Kubernetes helps to manage the numerous clusters at the same time.
3. Kubernetes also helps to check constantly the health of nodes and containers.
4. Kubernetes will allow the scaling resources horizontally and vertically, which is easy and quick.
5. Kubernetes also knows where to place the containers. The Kubernetes does this by calculating the best location for the containers.
6. Kubernetes provides additional services to the containers.
7. Kubernetes also offers security, networking and storage services to the container.

### [1. What is minikube?](https://www.onlineinterviewquestions.com/kubernetes-interview-questions/page/2/" \l "collapseUnfiled1)

1. Minikube is a type of tool that makes the Kubernetes easy to run locally. Minikube basically runs on the single nodes Kubernetes cluster that is inside the virtual machine on your laptop. This is also used by the developers who are trying to develop by using Kubernetes day to day.

### 1. Define what Kubernetes is. What is its purpose?

It refers to a system that does automating and scaling as well as the management of containerized applications. It handles the grouping of containers, which provides application for the logical units for management as well as discovery.

### 2. Could you please provide a definition of a Docker?

Docker refers to an open-source platform, which handles software development. The main advantage with it is it packages the settings into particular apparatus and then allows their portability within the systems running a Linux operating system.

### 3. Give a description of Orchestration when it comes to software?

The service orchestration alludes to integration of multiple services for them to allow the automation of the processes or synchronizing the information on a timely basis. As such, the point-to-point integration may be used as one such path for the least amount of resistance.

### 4. How would you say that Kubernetes is related to Docker?

Docker is responsible for the management of the lifecycle of containers and these are manually linked and orchestrated with Kubernetes.

### 5. What are the scenarios in which a Java Developer is going to use Docker?

Following scenarios are those a java developer can use docker for:

• Running UAT’s with the use of Docker.   
• Sharing development workspace, with pre-configured development environment.  
• Continuous integration is a popular use case for Docker. The teams looking build and deploy their applications may use Docker combined with ecosystem tools such as Jenkins to drive applications from development, testing, staging and into production without the need to change any code.

### 6. What are Daemon sets?

The Daemon set is a set of pods that is run only once on a host. It can be used for host layer attributes like a network, host monitoring or the storage plugin or things, which you would not want to run more than once on a host.

### 7. What is the “Master”?

Master refers to a central point of control, which gives a unified view of a cluster. There is a single master node, which controls different minions. Master servers then work together to accept user requests and determine the best means of scheduling the workload containers, authenticate clients and nodes as well as adjust on the cluster wide networking and managing the scaling and health checking of responsibilities.

### 8. Could you please describe Kubernetes minions.

The minion is an element that transforms itself into a worker element, which is a node. The node could refer to a physical machine or VM, which is dependent on the particular cluster. Each node has the services, which are required to run the pods and they are monitored through the main components. The services here may include that of the container runtime such as Kubelet.

### 9. What are labels and annotations when it comes to Kubernetes?

A label in Kubernetes is a meaningful type of tag word, which is attached to the Kubernetes objects in order to make them as part of a group. The Labels may be used for working on different instances for the purposes of management or even routing purposes. For one, the controller-based objects may use the labels to mark the pods they would operate on. The microservices use labels to understand the structure of the backend pods they route the requests toward. The labels are some of the key value pairs. Each unit may have more than one label but each unit may only have one entry for each of the keys. The key is most commonly utilized as an identifier or unique ID. However, at the same time may classify the objects using other criteria according to public access, application versions and the developmental stages.

The annotations attach arbitrary key value information to the Kubernetes object. The levels, however, ought to be utilized for meaningful information in order to match a pod with selection criteria, so the annotations have less structured data. The annotations are a means for adding more metadata to the object, which is not helpful for the selection purposes.

### 10. What are the node server components for Kubernetes?

In Kubernetes the server do their work through running containers, which are known as nodes. The execution of tasks and reporting the status to the master would be the main objective of the Node server.

The main process of the Kubernetes node, which does some of the significant container operations:

• The Kubelet is the node-daemon, which communicates with Kubernetes master for all the machines, which are a part of a Kubernetes cluster.  
• It regularly accesses the controller in order to check and report on the status of the cluster.  
• It merges the available CPU, memory and disk for a node into the large Kubernetes cluster. It also communicates the state of the containers back up to the API server for control loops in order to observe the current state of the containers.

### 11. Can you please tell me what the difference is between deploying applications on dedicated hosts versus on Kubernetes containers?

This kind of architecture is generally going to have an operating system associated with it. And that OS will have a kernel that will have various libraries installed on it for the application to work. In this framework, it is possible to have N number (meaning any number) of applications and all of them are going to share the available libraries. Each are present and accessed within the operating system, though, while deploying the applications in containers, the architecture can sometimes be a bit different. This architecture approach will have a kernel and that may be the only thing, which will be common between all of the applications. If there is a particular one, which requires Java, then that one will get access to it. The individual blocks, which can be seen on the right side of the diagram, are containerized and these are isolated from the other applications. That means the applications would have the required libraries and binaries, which are isolated from the system and these, cannot be encroached by other applications.

### 12. Describe the meaning of a cluster in Kubernetes?

The master and the node machinery refer to the ones run by the Kubernetes cluster orchestration framework. A container cluster would be the foundation of the Container Engine. The objects that represent the applications, which are containerized, are then run atop the cluster.

### 13. What is a Swarm in Docker?

The docker Swarm is a clustering and scheduling tool for the Docker containers. When it comes to Swarm, the IT administrators and developers would establish and manage a cluster of Docker nodes as part of the single virtual system.

### 14. What is OpenShift?

OpenShift online is Red Hat’s public cloud application development and hosting platform which provides automation for management, provision and the scaling of application so it is possible to focus on writing the code for the business or big idea.

### 15. What does the nodes status contain?

Some of the following things would be the main components of the node status.

• Address  
• Condition  
• Capacity  
• Info

### 16. What are Pods in Kubernetes?

A Kubernetes pod is a particular group of containers, which are deployed, in the same host. The Pods have the capacity to operate on a level, which is higher as compared to the individual containers. That is because the pods have the group of containers, which work together to produce an artifact or to process a particular set of work.

### 17. Give a description of what a Namespace is when it comes to Kubernetes?

The Namespace can be used in different environments with different users, who operate across a number of projects or even teams. They refer to the process of dividing the cluster resources between the different utilizations. For the future iterations, the objects within a similar Namespace would have a similar access control policy.

### 18. Provide a definition of a node within Kubernetes?

The node is the worker element within the framework and it was previously known as a minion. It could be a virtual or physical type of machine but this is dependent on the cluster. The nodes have the services needed for running the pods and they are managed via the main components. The services of each of the node may include such things as the Kubelet, Docker and Kube Proxy.

### 19. What are the reasons for the utilization of Docker?

It provides a similar capability without considerations of the overhead, which would be there with a virtual machine. Docker makes it possible to place the environment and configuration into code before its deployment. A similar Docker configuration could be used also for the different types of environments. That would then decouple the needs of the infrastructure from that of the environment.

### 20. What is a Docker in Cloud?

The node alludes to an individual Linux hosting platform utilized for the deployment and running of the applications. The cloud does not provide hosting services so the applications, containers and services may run on the particular hosts. The hosts do hail from different origins like the virtual machines, physical servers or the cloud providers.

### 21. Can you please discuss the cluster of containers in Kubernetes.

A container cluster concerns a set of machine elements that are nodes. The clusters initiate particular routes so the containers running on the nodes are able to communicate with each other. The server of the Kubernetes API does not function on these nodes. The container engine is the one which provides hosting for the API server.

### 22. Can you please tell me what a Container Orchestration is?

If you consider an engineering scenario where there is a need for 5 to 6 microservices for the application for a moment. These micro-services are then placed within individual containers but they are not going to be able to communicate without the container orchestration. Therefore as orchestration means the combination of all of the instruments playing together in harmony in music, container orchestration would mean all of the services within the individual containers working together in order to fulfill the requirements of a single particular server.

### 23. What is the significance of Container Orchestration?

Consider there are 5 to 6 micro-services for a single application performing different tasks and all of them are living within containers. In order to make sure these containers communicate with one other properly, there is a need for what is called container orchestration, which is built right into Kubernetes.

### 24. What are some of the different attributes of Kubernetes?

• Automated scheduling: Kubernetes allows advanced scheduler to launch container on the particular cluster nodes.  
• Automated rollouts and rollback: Kubernetes supports the different rollouts and rollbacks for the desired state of the containerized application.  
• Self-healing features: rescheduling, replacing and restarting the containers, which are dead.  
• Horizontal scaling and load balancing: Kubernetes may scale up and down the application according to the requirements.

### 25. What are the means by which Kubernetes simplifies containerized Deployment?

If you can think about a typical application, it would essentially be a cluster of containers running across different hosts. These containers would need to communicate with each other. Because Kubernetes is a cloud agnostic and may run on private as well as, public providers it has to be ones choice to simplify the containerized deployment.

### 26. Can you please tell me what some of the main advantages of Kubernetes is?

With container orchestration tool Kubernetes, it has become easy for one to handle the containers. You may respond to different customer demands through deploying the applications in a faster manner and in a way, which is predictable. So there is:

• Automated rollback  
• Automated scheduling   
• Horizontal scaling  
• Auto healing capabilities

### 27. What is the difference between Docker Swarm and Kubernetes?

• The installation structure of Kubernetes is complex though if it were installed then the cluster would be robust. However, the Docker Swarm installation process happens to be simple though the cluster is not robust per se.   
• Kubernetes is incredibly scalable. It was essentially built for large scale. However, the Docker swarm scales are five times faster than Kubernetes and is very scalable.  
• Kubernetes may also do the process of the auto scaling though the Docker swarm cannot do the process of the auto scaling.

### 28. Can you please tell me what a ‘Heapster’ is with Kubernetes?

A Heapster refers to a cluster wide aggregator of data that is provided by Kubelet running on each one of the nodes. This container management tool is supported natively on Kubernetes clusters and runs as a pod in the same way as any other pod within the cluster would run. Because of this it discovers all nodes within the cluster and queries the usage information from the Kubernetes nodes.

### 29. Can you please tell me what a Google Container Engine is?

A Google Container Engine refers to an open source management platform for the Docker containers and the clusters; it was built by Google for engineer operations. The Kubernetes based engine supports the clusters that run within the Google’s public cloud services, which is often required for large-scale applications.

### 30. What positive things can you say about clusters within Kubernetes?

The fundamental thesis behind Kubernetes is it is possible to enforce the desired state management. As a result it is possible to feed the cluster services a particular configuration. This is going to then go to the cluster services to go out and run the configuration within the [configured] infrastructure. As such, the deployment file is going to have all of the configurations, which required nourishment within the cluster services. The file will also require feeding to the API and so it would mean the cluster services retained the means for scheduling the pods in the appropriate setting and making sure the appropriate pods are running. That way the worker nodes, Kubelet and the API make up the Kubernet cluster.

### 31. Describe Minikube?

This is a tool, which makes it more efficient to run the Kubernetes system from a local point. It can run be described as running a single node of Kubernetes cluster within a virtual machine or VM.

### 32. What is Kubectl?

Kubectl refers to the platform which an engineer can pass commands to the cluster. By doing so, it provides the CLI with the means to run commands against the Kubernetes cluster via different create and manage commands on the Kubernetes component.

### 33. What is Kubelet?

The Kublet refers to an agent service that runs on each node and allows the slave to communicate with the master node. Therefore, Kubelet works on the the containers that are provided to it within the PodSpec and makes sure the containers prescribed within the Podspec are health and running adequately.

### 34. What are K8s?

This is another term for Kubernetes, whereby (K-eight characters-S), would be the open source orchestration framework for the containerized applications. It just another name for what we call Kubernetes.

### 35. Can you please tell me what Kube Proxy is?

The Kubernetes network proxy is a command or agent can be run on each one of the Kubernetes nodes. The service cluster type IPs and ports are then found through some Docker links for compatible environmental elements that specify the ports which are opened by a service type proxy. This is an option, which allows cluster DNS for the IPS cluster.

### 36. What is the process that runs on Kubernetes Master Node?

Kube-apiserver process runs on Kubernetes master node.

### 37. Can you discuss how the master node works in Kubernetes.

Kubernetes master controls the nodes and the containers are within the nodes. These individual containers are stored within pods and inside each pod, based according to the configuration and requirements. Because of this, if the Kubernetes pods have to be deployed, then they may either be accessed using a user interface or command line tool. These pods would be scheduled to run on the nodes and based on the source requirements, the pods are allocated to see these nodes. The job of the kube APIserver is to make certain there is absolute communication between the Kubernetes node and its master components.

### 38. What is the role of the Kube apiserver and the Kube scheduler?

The Kube apiserver follows the scale out architecture plan and is the front end which comes to the master node control panel. That would expose all the APIs of the Kubernetes Master Node components. It is responsible for the establishment of communication between the Kubernetes node and the Kubernetes master components. The Kube scheduler is at its core, responsible for the distribution and management of the workload on the worker nodes. It selects the most suitable nodes to run the unscheduled pod depending on the resource needs and keeps track of the overall resource utilization. It makes certain the workload is not scheduled on the nodes that may already be full.

### 39. Give a short description of the process validating the configured information for API objects like the Pods.

Kube apiserver process verifies and modifies the data for the API services.

### 40. Can you please tell me the general use of the Kube [controller manager](https://kubernetes.io/docs/reference/command-line-tools-reference/kube-controller-manager/).

Very simply put, the Kube APIerver procedures verify and modify the data for the API objects.

### 41. Kubernetes Objects are made up of?

Kubernetes objects are made of Pod, Service and volume.

### 42. Determine the Kubernetes controllers.

The Kubernetes controllers include Deployment controller and Replicaset.

### 43. What is ECTD?

ECTD is written is a tool that is written in the Go Programming language (created by Google) and is a distributed key value store. Because of this function, the ECTD stores of the configuration data of the Kubernetes cluster which shows the state of the cluster at any time.

### 44. Describe the different types of services within Kubernetes.

• Cluster IP: this function exposes the services on a clusters internal IP address. It is also the default service type and makes the service only reachable from inside of the cluster.  
• Node Port: it is a Cluster IP service to which Node Port service is going to route and is automatically created. It also exposes the service on each Node IP at a static port.   
• External Name: this service maps the contents of the External Name field through returning a CNAME record with that particular value. There is no proxying of any sort, which is set up.  
• Load Balancer: this one exposes the services from an external perspective with the use of a [cloud provider’s](https://www.zdnet.com/article/top-cloud-providers-2018-how-aws-microsoft-google-ibm-oracle-alibaba-stack-up/) load balancer. The services to which the external load balancer are going to route are automatically created.

### 45. Can you please tell me what the Cloud Controller manager is?

The Cloud Controller manager is there for persistent storage, abstracting the clouds code base from the main Kubernetes code base and the overall management of the communication with underlying cloud services. It may be split into different containers depending on the cloud platform that is being run by the engineers. Then it allows the cloud vendors and Kubernetes code to be built upon without an element of dependency. Therefore, the cloud vendor can develop their code and then connect with the Kubernetes cloud controller manager while running Kubernetes. There are different forms of cloud controller manager and they include the following:

• Node controller: this one checks and confirms the node is deleted in a proper manner after having been stopped.  
• Volume controller: this one manages the storage volume and interacts with the cloud provider in order to orchestrate the volume.  
• Service controller: the service controller is there for the management of the cloud providing the load balancers.  
• Route controller: the route controller manages the traffic routes within the underlying cloud infrastructure.

### 46. What is ingress Network and what are the ways in which it works?

The Ingress Network is a collection of engineering rules which act as an entry point to the Kubernetes cluster. This allows for inbound connections that can be configured to provide the services externally through URLs (like POST endpoints), which are reachable through virtual hosting or through a load balancer. Ingress, therefore is an API object, which handles the external access to the services within a cluster usually through the means of HTTP and would be the best way of exposing service. The working of the ingress network can be illustrated through the following example. The two nodes have the POD and root network namespacing which is utilizing a Linux bridge. There should be a new virtual Ethernet device described as flannel0 (a network plugin) added to the root network. You might consider now that would like the packet to go from the first to the fourth type of pod.

• It is then passed to CBR0, and that is what brings the ARP request for allotting the destination and it is found none would have the destination IP address.  
• At this time the bridge may sends the packet to flannel0 as the node’s route table is configured with flannel0.  
• That mean the flannel daemon communicates with the API server of Kubernetes to know all of the pod IPs and their particular respective nodes for mapping for the pods IPs to the node IPs.   
• The job of the network plugin is to wrap the packet within a UDP that has extra headers. This subsequently changes the sources and destination IPs to their respective nodes. Then it sends the packet through eth0.   
• Because the route table is knowledgeable about the way to route traffic between the different nodes, it gives the packet to the destination node2.  
• The packet will get to eth0 of node2, and then goes back to flannel0 for de-capsulation. It would still emit back in the root network namespace. Additionally, the packet is forwarded to the Linux bridge to make an ARP request ascertaining the IP, which belonging to veth1. • The packet would cross the root network and reach the particular destination, which is Pod4.

### 47. Can you please tell me what some of the disadvantages of Kubernetes is?

• It is hard to install and configure   
• It takes time to start running and gain traction   
• There are no placements available as yet   
• It is not simple to manage the services

### 48. Can you please tell me what a headless service is?

A headless service is almost the same as a ‘Normal’ service, but does not necessarily have a Cluster IP associated with it. This allows you to directly reach the pods without having to access it through the proxy.

### 49. Can you please tell me what federated clusters are?

Multiple clusters may be managed as a single cluster or with the assistance of federated clusters. It is possible to create multiple clusters within the data center or cloud and then use the federation for controlling or managing them in one place. The federated clusters are able to achieve this by doing some of the following:

• Syncing resources across the different clusters: this keeps the resource sync across the different clusters in order to deploy the same deployment set across the different clusters.  
• Cross Cluster, discover: this provides the ability to have DNS and Load Balancer with backend from the participating clusters.

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

The replica set and replication controller generally do the same things for the engineering operators. Both of them have the job of making certain that a specified number of pod replicas are running at the same time. The difference is there with the use of selectors to replicate the pods. The replica Set utilizes Set Based selectors though the replication controllers utilize Equity Based selectors.

• Equity Based Selectors: this type of selector allows for the filtering of label keys and values. That means the equity-based selector is only going to search for the pods that are going to have the same exact phrase as compared to the label. An example in this case is considering the label key claims app=nginx.  
• Selector based selectors: this selector allows for the filtering of keys according to a set of values. That means the selector-based selector is going to search for the pods whose label has been mentioned within the set. For example, if the label key says app within (nginx, Apache or NPS). With this selector, if the application has any of the nginx, Apache or NPS settings then the selector is going to return it as true.

### 51. Can you tell me how you get a static IP for a Kubernetes load balancer?

The Kubernetes Master is allowed to assign a new IP static address. It is possible to get a static IP for Kubernetes load balancer through changing the DNS records.

**Q: What is Kubernetes?  
A:** Kubernetes is Google's open source system for managing Linux containers across private, public and hybrid cloud environments.  
It is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.   
It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available. It contains tools for orchestration, service discovery and load balancing that can be used with Docker and Rocket containers. As needs change, a developer can move container workloads in Kubernetes to another cloud provider without changing the code.   
It helps automates the deployment, scaling, maintenance, scheduling and operation of multiple application containers across clusters of nodes.   
[Understanding need for Kubernetes](https://www.javainuse.com/devOps/kubernetes)  
  
**Q: What are the advantages of Kubernetes?  
A:** The advantages of using Kubernetes are as follows-

* **Automated Scheduling-** Kubernetes provides advanced scheduler to launch container on cluster nodes. Kubernetes role is to automate the distribution (scheduling) of application containers across a cluster in an efficient way.
* **Auto Healing Capabilities** - Kubernetes auto-healing mechanisms, such as auto-restarting, re-scheduling, and replicating containers
* **Automated Rollback** - Sometimes you may want to rollback a Deployment; for example, when the Deployment is not stable, such as crash looping. By default, all of the Deployments rollout history is kept in the system so that you can rollback anytime you want.
* **Horizontal Scaling** - Autoscaling is one of the key features in Kubernetes cluster. It is a feature in which the cluster is capable of increasing the number of nodes as the demand for service response increases and decrease the number of nodes as the requirement decreases.

**Q: What is Docker?  
A:** Consider a scenario - You have just joined a new organization as a developer. You will now have to setup the project with the assistance of a fellow developer. He suggests you follow certain steps for setting up the required environment and then start the project deployable like a WAR. You do the same, but keep getting some or other issues regarding environment configuration. May be even your fellow developer has forgot some configuration property he might have set. Well you are stuck in such a situation. This is known as Dependency Hell. Other similar scenario of this dependency hell are - The application is running on my dev machine but not in production. Dont know what issue is. There is also other scenarios like Matrix of Hell. But this is mostly related to DEVOPS people. **Docker to the rescue.**  
**Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.**[Docker Tutorials](https://www.javainuse.com/devOps/docker)   
  
  
  
**Q: Whats is difference between Kubernetes and Docker Swarm?  
A:** Docker already has its own orchestration manager named Docker Swarm. Docker Swarm is much easier to use and does not require to learn any new tool or technology since its part of Docker. Kubernetes helps manage more complex container deployments while Docker Swarm offers a simple approach to get started with. Kubernetes helps support higher demands production environments and is used by many large organizations.  
  
**Q: How to deploy Spring Boot WAR to Docker?  
A:**[Deploying Spring Based WAR Application to Docker](https://www.javainuse.com/devOps/docker/docker-war)  
  
**Q: How to deploy Spring Boot JAR to Docker?  
A:**[Deploying Spring Based JAR Application to Docker](https://www.javainuse.com/devOps/docker/docker-jar)  
  
**Q: How to deploy multiple microservices to Docker?  
A:**[Deploying Multiple Spring Boot Microservices using Docker Networking](https://www.javainuse.com/devOps/docker/docker-networking)  
  
**Q: What are the scenarios in which a Java Developer will use Docker?  
A:**Following scenarios a java developer can use docker-

* Sharing development workspace, with preconfigured development environment.
* Continuous integration is one of the most popular use cases for Docker. Teams looking build and deploy their applications quickly use Docker, combined with ecosystem tools like Jenkins, to drive apps from dev, testing staging and into production without having to change any code.
* Running UAT's using Docker

**Q: What are namespaces in Kubernetes?  
A:** Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces. Kubernetes allows use of multiple namespaces.  
  
**Q: What is a node in Kubernetes?  
A:** A node is a worker machine in Kubernetes, previously known as a minion. A node may be a VM or physical machine, depending on the cluster. Each node has the services necessary to run pods and is managed by the master components. The services on a node include the container runtime, kubelet and kube-proxy.   
  
**Q: What is a pod in Kubernetes?  
A:** A Kubernetes pod is a group of containers that are deployed together on the same host. If you frequently deploy single containers, you can generally replace the word "pod" with "container" and accurately understand the concept.   
