

where you have most frequently asked Kubernetes interview questions in the interviews. So, have a look at these questions to gain in-depth knowledge of various concepts related to Kubernetes.

Q1) What is Kubernetes?

Answer: Kubernetes is an extensible, portable, and an open-source platform used for managing services and containerized workloads. It is a large and fast-growing ecosystem as its services, tools, and supports that are frequently and widely available.

Q2) Describe the history of Kubrnetes?

Answer: Kubernetes word has been originated from Greek with a meaning pilot or helmsman. It was foremostly designed by Google in 2014. It has become the building block for running the workload productions at scale for Google. Later it has been maintained by Cloud Native Computing Foundation.

Q3) What are the major differences between Kubernetes and Docker Swarm?

Answer:

Features	Kubernetes	Docker Swarm
Applicati on definition	The application is set up using a consolidation of pods and deployments.	In this, the applications are set as micro-services.
Logging and moni toring	Kubernetes supports multiple versions of monitoring and logging.	It is supported for only auditing of third-party applications.
Scalabilit y	Highly scalable	More scalable when compared to Kubernetes
Networki ng	In this the networking model is flat.	The overlay of the network is created by joining a node with the cluster.

Features	Kubernetes	Docker Swarm
Availability	It provides a huge amount of availability among the nodes.	It also provides large and more availability for the services that are depicted in Swarm

Q4) What are the reasons why Kubernetes is more useful by walking back in time?

Answer: Kubernetes mainly contains three important deployments. They are:

Traditional Deployment

Virtualized Deployment

Container Deployment

These three are the most crucial aspects that are useful by going back in time

Traditional Deployment:

Earlier in this era, applications can run on the physical servers by various organizations. This causes allocation issues related to resources, which can be solved by running each and every application on the different servers.

Virtualized Deployment:

The introduction of virtualization was done so, that it allows us to run many numbers of virtual machines on only one server CPU.

Container Deployment:

Container deployment has flexible isolation properties in order to share an operating system among applications.

Q5) Why do we need Kubernetes and what it can do?

Answer: Kubernetes is the container that provides a good way to run and bundle your applications. We need to effectively manage the containers in the production environment that

allows us to run applications. It also provides a framework to run distributed systems resiliently.

Q6) What are the features of Kubernetes?

•

Answer: The features of Kubernetes are as follows:

Storage orchestration

•

Automated rollbacks and rollouts

self-healing

•

Configuration management

Packing of bin automatically

Load balancing and service discovery

Q7) List out the components of Kubernetes?

Answer: There are mainly three components to deliver a functioning Kubernetes cluster. They are:

Addons

Node components

Master components

Q8) How does Kubernetes relate to Docker?

Answer: Kubernetes is a container for the Docker which is more comprehensive than Docker Swarm and is designed to counterpart clusters of the nodes at a level in a well-defined manner. Whereas, Docker is the platform tool for building and running the Docker containers.

Q9) Define Kube Scheduler?

Answer: It is the important aspect of the master node which notices the newly created pods with no assigned node for it, and selects one of the nodes to run on them.

Q10) What are the benefits of Kubernetes?

Answer: The benefits of Kubernetes are as follows:

It provides easy service organizations with pods.

It works on any of the OS as it an open-source modular tool.

It has a huge community among container orchestration tools.

Q11) Define Kubernetes Namespace?

Answer: Namespaces are used in environments where there are multiple users in a respective team or project. It is mainly designed to provide scopes for the names and the assigned names must be unique within the namespace. Moreover, they provide a way to divide cluster resources within the existing namespace itself.

Q12) Mention the namespaces that initially the Kubernetes starts with?

Answer: Initially the Kubernetes starts with three namespaces, and they are:

kube-public: This is created automatically and can be read by all the users and it is the most reserved for cluster usage.

default: It is for the objects who do not contain namespaces.

kube-system: It is for the objects which are created by the Kubernetes system.

Example of the initial namespaces in Kubernetes is given below:

kubectll get namespace

NAME	STATUS	AGE
default	Active	1d

NAME	STATUS	AGE
kube-system	Active	1d
kube-public	Active	1d

Q13) What are Kubernetes pods?

Answer: Pods are defined as the group of the containers that are set up on the same host.

Applications within the pod also have access to shared volumes.

Q14) Define Kubelets?

Answer: It is the node agent that runs on each node. It works based on PodSpec, which JSON object in terms of a pod. The Kubelet logs take a set of PodSpecs that provides various mechanisms and ensures that the PodSpecs are running effectively.

Q15) What is the command Kubectl and its syntax?

Answer: It is defined as a CLI (command-line interface) for performing and running commands against Kubernetes clusters.

The syntax for Kubectl is

kubectl **[command]** **[TYPE]** **[NAME]** **[flags]**

Q16) How does the Kubernetes Cluster work?

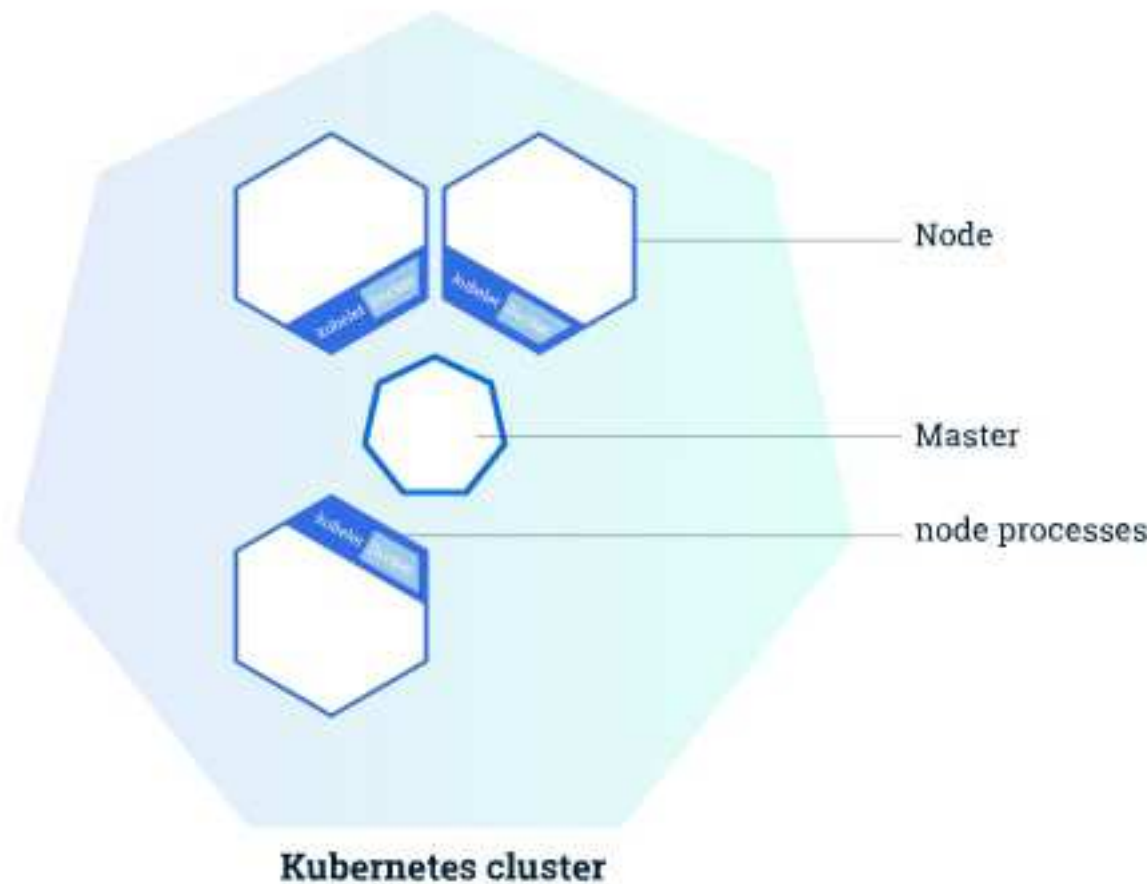
Answer:

•

The master is the one who is responsible for managing clusters. Kubernetes automates the scheduling and distribution of application containers across the cluster in a more effective manner. In Kubernetes, Minikubes is used to create clusters. The Kubernetes cluster consists of mainly two important sources, and they are:

The master coordinates the cluster

Nodes are the workers who run applications



Q17) What do you understand by the term Kube-proxy?

Answer: This is a network-proxy that runs on each and every node and also reflects as defined in the Kubernetes API. This proxy can also perform stream forwarding across a set of backends. It is one of the optional add-ons that provide the DNS cluster for the cluster APIs.

The syntax to configure Proxy is:

kube-proxy [flags]

Q18) Describe in brief the working of the master node in Kubernetes?

Answer: Kubernetes master is mainly designed to control the nodes and the nodes mainly consist of a crucial part called containers. Now, here comes the pods these pods are made up of a group of containers based upon the requirements and configurations. Every container which we utilize is present inside a pod so, if the set-up for the pod is made then the can be deployed using CUI (Command Line Interface). Scheduling of the pods is done based on the node and relevant

requirements. The connection between the node and the master components in the Kubernetes is made using the Kube-apiserver.

Q19) What is the function of Kube-apiserver?

Answer: This API server of Kubernetes is mainly used to configure and validate API objects that include replication controllers, services, pods, and many more. Kube-apiserver services the p y R E S T operations and provides the frontend to the cluster s shared region. interaction takes place between the components.

The representation for Kube-apiserver is provided as follows:

kube-apiserver [flags]

Q20) What is the role of a Kube-scheduler?

Answer: It is defined as a workload-specific, policy rich, and topology-aware function which majorly impacts on availability, capability, and performance. The duty of scheduler is to collect individual and collective resource requirements, data locality, hardware/software policy constraints, inter-workload interference, and many more into its account. API shows or displays the necessary workload requirements.

The representation for the Kube-scheduler is:

kube-scheduler [flags]

Q21) Describe a few words about Kuberntes Controller Manager?

Answer: Kube-controller-manager is a divinity that embeds the crucial core control loops shipped with the Kubernetes. In most of the robotic and automation applications, control loops are the non-terminating loops that regulate the state of the particular system. In Kubernetes, the controller itself is the control loop that watches the shared state of the cluster using the apiserver. Examples of the controllers that ships today with Kubernetes are namespaces, replications, and many more.

The representation for the Kube-controller-manager is given as:

kube-controller-manager [flags] •

Q22) What do you mean by the term etcd?

Answer: Kubernetes uses etcd to store all its data. The reason behind it is that Kubernetes is a distributed system so as to store distributed data it uses etcd. Etcd is a distributed, most reliable key-value for storing the most critical data.

Q23) Define the term Minikube in Kubernetes?

Answer: To easily learn Kubernetes locally minikube tool is used. This runs on the single-node Kubernetes cluster inside a virtual machine.

Q 24) What is Kubernetes load balancing?

Answer: The process of load balancing lets you show or display the services. There are two types of load balancing in Kubernetes, and they are:

Internal load balancing

External load balancing

Internal load balancing:

This balancing is used to balance the loads automatically and allocates the pods within the necessary configuration.

External load balancing:

It transfers or drags the entire traffic from the external loads to backend pods.

Q25) List out the components that interact with the node interface of Kubernetes?

Answer: The following are the components that interact with the node interface of Kubernetes, and they are:

Node Controller

Kubelet

Kubectl

Q26) Name the process which runs on Kubernetes Master Node?

Answer: The process that runs on Kubernetes Master Node is called the Kube-apiserver process.

Q27) What are Kubernetes Minions?

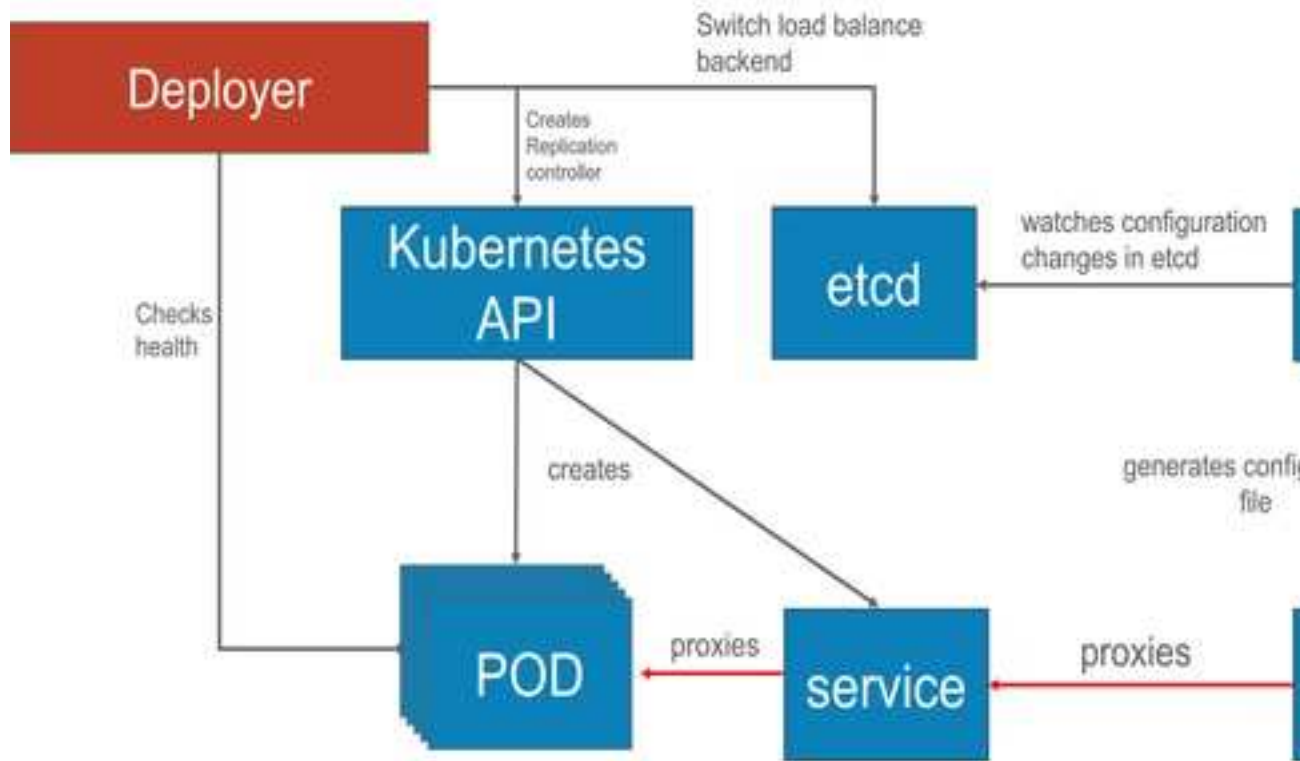
Answer: Node in the Kubernetes is called as minions previously, it is a work machine in the Kubernetes. Each and every node in the Kuberntes contains the services to run the pods.

Q28) What is heapster?

Answer: Heapster is a metrics collection and performance management system for the Kubernetes versions. It allows us for a collection of workloads, pods, and many more.

Q29) Explain Kubernetes architecture with a neat diagram?

Answer:



Q30) What is the future scope for Kubernetes?

Answer: Kubernetes will become one of the most used operating system (OS) for the cloud in the future. The future of Kubernetes mostly lies in virtual machines (VM) than n containers.

Q31) What do you mean by Kubernetes?

Kubernetes is a technology developed by Google Research and Development Team, is available open source. Kubernetes allows Docker containers to establish multiple services at a time.

Simply, it is a container management tool designed to work in different types of platforms. This tool is widely used to automate, scale, monitor, manage and deploy container based applications.

Q32) State the important features of Kubernetes.

The features of Kubernetes are stated as follows:

- It provides better infrastructure for Container based applications

- It provides uninterrupted development services, deployment services and integration services.

- It schedules operations automatically.

It restores a database or program to a previous defined state automatically.

It grades operations according to a scale automatically.

It corrects errors automatically.

Q33) What are the differences between Docker Swarm and Kubernetes?

Docker Swarm	Kubernetes
Installing and Configuring the clusters is very easy but the clusters were not resilient	Installing and Configuring the clusters is very difficult but the clusters were resilient (strong)
Docker Swarm does not contain GUI	Kubernetes dashboard is present in a GUI
Scales good with five times more faster than Kubernetes	Scales faster but slower than Docker swarm
Docker swarm does not scale automatically	Kubernetes scales automatically
Possess automatic load balancing feature through which it balances in case of any traffic between docker containers in cluster	Does not contain automatic load balancing feature and therefore require manual help in case of any traffic between containers and the pods.

Q34) What are the components of Kubernetes Master machine? Explain

The following are the key components of Kubernetes Master machine:

ETCD: ETCD is used to store the configuration data of every node present in the cluster. It can store good amount of key values which can be shared with several nodes in the cluster. Because of its sensitivity, Kubernetes API Server can only access ETCD. But, it contain a shared key value store which can be accessed by everyone.

API Server: Kubernetes itself is an API server controls and manages the operations in cluster through API Server. This server provides an interface to access various system libraries and tools to communicate with it.

Process Planner(Scheduler): Scheduling is the major component of Kubernetes Master machine. Scheduler shares the workload. Scheduler is responsible to monitor the amount of workload distributed and used in the cluster nodes. It also keeps the workload after monitoring on the available resources to receive the workload.

Control Manager: This component is responsible to administer the current position of cluster. It is equivalent to a daemon process continuously runs in a unending loop which collects and sends the collected data to the API server. It handles and controls various controllers.

Q35) Explain the node components of Kubernetes.

The following are the major components of a server node to exchange information with Kubernetes.

Docker: Every node contain Docker to run the containers smoothly and effectively. Docker is the basic component of every node in a cluster.

Proxy service of Kubernetes: Proxy service is responsible to establish communication to the host. Every node communicates with the host through proxy. Proxy service helps nodes to transmit data to the containers upon its request and is also responsible for load balancing. It is also responsible to control pods present in node, data volumes, creation of new containers, secrets etc.,

Service of Kubelet: Kubelet service helps every node to share information from the control pane and vice versa. Kubelet is responsible to read the details of node configuration and the write values which were present in the ETCD store. This service administers the port forwarding, protocols of network etc.,

Q36) What do you mean by Kubernetes Name space?

Namespaces are given to provide an identity to the user to differentiate them from the other users. Namespace assigned to a user must be unique. Through namespaces, cluster resources can be separated and shared within the assigned namespace itself.

Q37) State the functions of Kubernetes name space.

The primary functions of Kubernetes namespace are stated below:

Namespaces assist information exchange between pod to pod through the same namespace.

They are considered as virtual clusters which will be present on the same cluster.

Namespaces are used to deliver logical segregation of team and their corresponding environments.

Q38) How do you create a Namespace?

To create a name space, the following command should be written:

```
kubectl create -f namespace.yml
```

Q39) Write commands to control the Namespace.

To control the name space, we have to create a name space initially:

```
kubectl create -f namespace.yml
```

Then, we have to check the available namespaces from the list:

```
kubectl get namespace
```

To get a specific name space we require, use the following command:

```
kubectl get namespace<xyz>
```

To describe the services offered by the namespace, use the command:

```
kubectl describe namespace<xyz>
```

If you want to delete a namespace from the list, use the following command:

```
kubectl delete namespace<xyz>
```

Note: xyz is given for example. You can give any name in the namespace region.

Q40) Explain how will you setup Kubernetes.

Virtual Data Center is the basic setup before installing Kubernetes. Virtual Data center is actually believed to be set of machines which can interact with each of them through a network. If the user does not have any existing infrastructure for cloud, he can go for setting up Virtual Data Center in the PROFITBRICKS. Once completing this setup, the user has to setup and configure the master and node. For an instance, we can consider the setup in Linux Ubuntu. Same setup can be followed in other Linux machines.

Installation of Docker is the basic setup to run Kubernetes. But, there are some prerequisites needed before installing Kubernetes. We shall install Docker initially to start with. Following steps should be followed to install Docker.

User has to provide login credentials and login as a root user

Install the apt package and update it if necessary. If update is needed, use the commands:

```
sudo apt-get update
```

```
sudo apt-get install apt-transport-https ca-certificates.
```

Q41) Once update is installed, add new key for GPG using the command:

```
sudo apt-key adv
```

This key will be extracted from the Docker list

Further, update the image of the API package using the command:

```
sudo apt-get update
```

Install Docker Engine. Check whether the kernel version you are using is the right one.

After installing Docker Engine, install etcd. Now, install Kubernetes on the machines.

Q42) What do you know about Kubelets?

Kubelet is nothing but a node agent which runs on every node on a cluster. Kubelet works on the basis of the specifications in Pods. These Kubelets were responsible to check whether the PodSpecs are working perfectly. For this purpose, Kubelets create logs and monitor the PodSpecs.

Q43) What do you know about Pods in Kubernetes?

Pods are actually contain a class of containers which are installed and run on the same host. Containers were present on pods and therefore configuring the pods as per the specifications is important. As per the requirement of the nodes in a cluster, scheduling of pods can be established.

Q44) What are the types of Kubernetes pods? How do you create them?

Kubernetes contain two kinds of Pods. They are:

Single Container Pod: User has to give Kubectl run command where he defined the image in Docker registry to create a single container pod. The following command is used to create a single container pod:

```
kubectl run <abcd> --image<xyz1234> --
```

```
--name <name of the pod>
```

```
--image <image name on the registry>
```

Multicontainer pods: To create multicontainer pods, we need to create a yaml file including the details of the containers. User has to define the complete specifications of the containers such as its name, image, port details, image pull policy, database name, etc.,

Q45) What is the use of the API server in Kubernetes?

The API server is responsible to provide a front end to the clusters that are shared. Through this interface, the master and node communicate with one another. The primary function of API server is to substantiate and configure the API objects which includes pods, associated services, controllers etc.,

Q46) What do you mean by Kubernetes images?

There is no specific support to Kubernetes images as on date and Docker images actually support Kubernetes. To create an infrastructure for Containers, Docker images are the primary elements to form it. Every container present inside a pod will contain a Docker image running on it.

Q47) Tell me about the functions of Kubernetes Jobs.

The important function of Kubernetes job is to form a single or multiple pods and to monitor, log how well they are running. Jobs reflect the running of pods and they assure how many pods finished successfully. A job is said to be complete if the specified number of pods successfully run and complete.

Q48) What do you know about Labels in Kubernetes?

Keys will contain some values. Labels contain pair of key values connected to pods, associated services and the replication controllers. Generally, labels were added to some object during creation. During run time, they can be modified.

Q49) What do you know about Selectors and what are the types of selectors in Kubernetes API?

Since multiple objects have the possibility of same labels, selectors are used in Kubernetes. Label Selectors are unique and users use it to choose a set of objects. Till date, Kubernetes API allows two kinds of Label selectors. They are:

Selectors based on Set: This kind of selector permits to filter the keys as per the set of values.

Selectors based on Equality: This kind of selector permit filter as per key and by value. If there is any matching object found, it should meet the expectations of the specified labels.

Q50) What do you know about Minions? Explain.

Minion is nothing but a node present in the Kubernetes cluster on a working machine. Minions can either be a virtual machine, a physical one or a cloud sample. Every node present in a cluster should meet the configuration specifications to run a pod on the node. Two prime services such

as kubelet and proxy services along with Docker were needed to establish interface and communication with the nodes which runs the Docker containers present in the pod which were created on the node. Minions were not actually formed by Kubernetes but could be formed by a cluster manager present in virtual or physical machines or by a service provider for a cloud.

Q51) What do you mean by Node Controller?

Node controller are the group of services which were running in the Kubernetes Master. Node controllers are responsible to observe the activities of the nodes present on a cluster. They do this as per the identity of metadata name assigned to a node. Node controller checks for the validity of a node. If the node is found valid, it assigns a fresh created pod to the valid node. If the node is invalid, node controller will wait till the node becomes valid so as to assign a pod.

Q52) Tell me about Google container Engine.

Google container Engine is available open source and is a Kubernetes based Engine which supports for clusters which can run within the public cloud services of Google. This engine services as a platform for Docker containers and clusters.

Q53)What do you mean byIngress network?

Ingress network provides set of rules to enter into the Kubernetes cluster. This network is responsible to provide the incoming connections further This allows inbound connections, further configured according to the required specifications so as to offer give services through URLs which are available externally,through load balance traffic, or by providing virtual hosting which is name based. Therefore, Ingress network can be defines as an API object that controls and administers external access to the services present in a cluster, through HTTP.

Q54) What do you know about Kubernetes Service?

Kubernetes service is defined as analytical pairs of pods. As per the information present on top of the pod, it will contain a DNS name and one IP address through which pods can be accessed. Kubernetes service is very useful to regulate and administer load balancing as per specific requirements. Kubernetes service also supports pods in scaling them too easily.

Q55) What are the types of Kubernetes services?

The following are the types of Kubernetes services:

Node port: Node port helps to fetch the details of a static port of the node deployed currently. With the assistance of Cluster IP, Node port routing can be established automatically. User can access this node port service away from the cluster through the following command:

NodeIP:nodePort.

ClusterIP: Cluster IP is responsible to fetch the information present in a Kubernetes cluster. It also aids in limiting the service within a cluster.

Load balancing: Load balancing is an important service available in Kubernetes to automatically balance the load in case of traffic. The above two services such as NodePort and ClusterIP were automatically created with which they help the external load balancer to do routing.

Q56) What are the functions of Replication controller?

The following are the main functions of the replication controller:

It is responsible to control and administer the lifecycle of the pod.

It is responsible to monitor and verify whether the allowed number of pod replicas were running

It helps the user to check the running status of the pod

Replication controller lets the user to alter a particular pod. The user can drag its position to the top or to the bottom.

Q57) What do you know about Replica set?

Replica set is considered as a substitute to the replication controller. The prime function of replica set is to assure the number of pod replicas running. There are two types of Label selectors supported by Kubernetes API. They are: Equality based selectors and Set based selectors. The primary difference between the replication controller and replica set is that, replication controller supports equality based selector alone whereas the replica set allows both the types of selectors.

Q58) How do you update, delete and rollback in a Deployment strategy?

Update: Through this feature, user could be able to update the existing deployment during runtime and before its completion. Through update, the ongoing deployment will end and a fresh deployment will be created.

Delete: Through this feature, the user could be able to cancel or pause the ongoing deployment by deleting the deployment before its completion. Creating similar deployment will resume the deployment.

Rollback: User can restore a database or program to a previously defined state. This process is called as Rollback. User could be able to rollback the ongoing deployment through this feature.

Q 59) What do you mean by Recreate and Rolling Up Deployment strategy?

With the aid of Deployment strategies, user could be able to replace the existing replication controller to a new replication controller. Recreate is used to kill all the running (existing) replication controllers and creates newer replication controllers. Recreate helps the user in faster deployment whereas it increases the downtime, if in case the new pods old pods.

Rolling update also helps the user to replace the existing replica controller to newer ones. But, the deployment time is slow and in fact, we could say, there is no deployment at all. Here, some old pods and some new pods were readily available to the user to process any time.

Q60) Write a command to create and fetch the deployment:

To create: `kubectl create -f Deployment.yaml` record

To fetch: `kubectl get deployments`

Q61) Write a command to check the status of deployment and to update a deployment.

To check the status: `kubectl rollout status deployment/Deployment`

To update a deployment: `kubect setimage deployment/Deployment tomcat = tomcat:6.0`

Q62) What do you mean by volumes? What are the differences between Docker volumes and Kubernetes Volumes?

Volumes can be considered as directories through which the containers in a pod can be accessed. The differences between Kubernetes volumes and Docker volumes are:

Kubernetes Volumes	Docker Volumes
Volumes are not limited to any particular container	Volumes are limited to a particular pod in a container
It supports all or any of the container deployed in a pod of kubernetes	Does not support all container deployed in Docker
Supports many types of storage on the pod and also supports multiple of storage at the same time	No such support in Docker

Q63) List the Kubernetes volume you are aware of.

The following are some of the Kubernetes volume which are widely used:

NFS: Network File System lets an ongoing NFS to let you mount on your pod. Though you remove the pod from the node, NFS volume will not be erased but only the volume is unmounted.

Flocker: Flocker is available open source and is used to control and administer data volumes. It is a manager for data volume for a clustered container. Through Flocker volume, user can create a Flocker dataset and mount the same to the pod. If in case, there is no such dataset available in Flocker, the user has to create the same through Flocker API.

EmptyDIR: Once a pod is assigned to a node, EmptyDIR is created. This volume stay active till the pod is alive and running on that particular node. EmptyDIR volume does not contain anything in the initial state and is empty; the user can read or write files from this volume. The data present in the volume gets erased once the pod is removed from that particular node.

AWS Elastic Block Store: This volume mounts Amazon Web Services Elastic Block Store on to

your pod. Though you remove the pod from the node, data in the volume remains.

GCE Persistent Disk: This volume mounts Google Compute Engine Persistent Disk on to your pod. Similar to AWS Elastic Block store, the data in the volume remains even after removing the pod from the node.

Host path: Host path mounts a directory or file from the file system of the host on to your pod.

RBD: Rados Block Device volume lets a Rados Block device to be mounted on to your pod. Similar to AWS Elastic Block store and GCE Persistent Disk Volumes, even after removing the pod from the node, the data in the volume remain.

Q64) What do you mean by Persistent Volume?

Persistent Volume is a network storage unit controlled by the administrator. PV is a strategy used to control an individual pod present in a cluster.

Q65) What do you mean by Persistent Volume Claim?

Persistent Volume Claim is actually the storage provided to the pods in Kubernetes after the request from Kubernetes. User is not expected to have knowledge in the provisioning and the claims has to be created where the pod is created and in the same namespace.

Q66) Define Secrets in Kubernetes.

As the name implies, secrets are sensitive information and in this context, they are login credentials of the user. Secrets are objects in Kubernetes which stores sensitive information namely the user name and the passwords after encrypting them.

Q67) How do you create secrets in Kubernetes?

Secrets can be created in various ways in Kubernetes. Some of them are

Through Text (txt) files

Through Yaml File

To create secrets from these files, user has to create username and password using `kubect` command. The secret file has to be saved in the corresponding file format.

Q68) Explain the Network Policy in Kubernetes.

Network policy contains a set of protocol to achieve information transfer between the pods and defines how those pods present in the same name space transfers information with one another. It also defines data transfer with the network endpoint. User has to enable the network policy in the API server while configuring it in run time. Through the resources available in the network policy, select pods using labels and set the rules to permit the data traffic to a particular pod.

Q69) What will happen while adding new API to Kubernetes?

If you add a fresh API to Kubernetes, the same will provide extra features to Kubernetes. So, adding a new API will improve the functioning ability of Kubernetes. But, this will increase the cost and maintenance of the entire system. So, there is a need to maintain the cost and complexity of the system. This can be achieved by defining some sets for the new API.

Q70) How do you make changes in the API?

Changes in the API server has to be done by the team members of Kubernetes. They are responsible to add a new API without affecting the functions in the existing system.

Q71) What are the API versions available? Explain.

Kubernetes supports several versions of API in order to provide support to multiple structures. Versioning is available at Alpha level, Beta level and Stable level. All these version features are in multiple standards.

Alpha level versions have alpha values. This version is prone to errors but the user can drop for support to rectify errors at any time. But, this version is limited to test in short time alone.

Beta level versions contain beta values. Scripts present in this version will be firm since because they are completely tested. User can look for support any time in case of any errors. This version is not recommended to use in commercial applications.

Stable level versions get many updates often. User has to get the recent version. Generally the

•
The version name will be vX, where v refers to the version and x refers to the

Q72) Explain Kubectl command.

Kubectl commands provide an interface to establish communication between pods. They are also used to control and administer the pods present in the Kubernetes cluster. To communicate with the Kubernetes cluster, user has to declare kubectl command locally. These commands are also used to communicate and control the cluster and the Kubernetes objects.

Q73) What are the kubectl commands you are aware of?

•
kubectl apply

•
kubectl annotate

•
kubectl attach

•
kubectl api-versions

kubectl autoscale

kubectl config

kubectl cluster-info

kubectl cluster-info dump

kubectl set cluster

kubectl get clusters

kubectl set-credentials

Q74) Using create command along with kubectl, what are the things possible?

User can create several things using the create command with kubectl. They are:

Creating name space

Creating deployment

Creating secrets

Creating secret generic

Creating secret docker registry

Creating quota

Creating service account

Creating node port

Creating load balancer

Creating Cluster IP

Q75) What is kubectl drain?

kubectl drain command is used to drain a specific node during maintenance. Once this command is given, the node goes for maintenance and is made unavailable to any user. This is done to avoid assigning this node to a new container. The node will be made available once it completes maintenance.

Q76) How do you create an application in Kubernetes?

Creating an application in Kubernetes requires creating an application in Docker, since Docker is essential for Kubernetes to perform its operation smoothly. User can do any of the following two things to install Docker: can download or do the installation using Docker file. Since Docker is available open source, the existing image from Docker hub can be downloaded and the same has to be stored in a local Docker registry.

To create a new application using Docker file, user has to create a Docker file initially. Once creating an image, the same can be transferred to the container after testing it completely.

Q77) What do you mean by application deployment in Kubernetes?

Deployment is the process of transferring images to the container and assigning the images to pods present in Kubernetes cluster. Application deployment automatically sets up the application cluster thereby setting the pod, replication controller, replica set and the deployment of service. Cluster set up is organized properly so as to ensure proper communication between the pods.

This set up also sets up a load balancer to divert traffic between pods. Pods exchange information between one another through objects in Kubernetes.

Q78) Define Autoscaling in Kubernetes.

One of the important feature of Kubernetes is Autoscaling. Autoscaling can be defined as scaling the nodes according to the demand for service response. Through this feature, cluster increases the number of nodes as per the service response demand and decreases the nodes in case of the decrease in service response requirement. This feature is supported currently in Google Container Engine and Google Cloud Engine and AWS is expected to provide this feature at the earliest.

Q79) How will you do monitoring in Kubernetes?

To manage larger clusters, monitoring is needed. Monitoring is yet another important support in Kubernetes. To do monitoring, we have several tools. Monitoring through Prometheus is a famous and widely used tool. This tool not monitors, but also comes with an alert system. It is available as open source. Prometheus is developed at Sound Cloud. This method has the capability to handle multi-dimensional data more accurately than other methods. Prometheus needs some more components to do monitoring. They are

Prometheus node exporter

Grafana

Ranch-eye

Influx DB

Prometheus exporter

Q80) What is Kubernetes Log?

Kubernetes container logs are much similar to Docker container logs. But, Kubernetes allows users to view logs of deployed pods i.e running pods. Through the following functions in Kubernetes, we can get even specific information as well.

Container name of Kubernetes

Pod name of Kubernetes

Name space of Kubernetes

Kubernetes UID and

Docker image name

Q81) What do you know about Sematext Docker Agent?

Sematext Docker agent is more famous among the recent day developers. It is a log collection agent with metrics and events. Sematext Docker agent runs as a small container in each Docker host and gathers metrics, events and logs for all the containers and cluster nodes. If core services are deployed in Docker containers, it observes every container inclusive of a container for Kubernetes core services.

[Share on Facebook](#)

[Share on Twitter](#)

[Share on LinkedIn](#)

[Share on Pinterest](#)