KUBERNETES INTRVIEW QUESTIONS:

1. **What is Kubernetes?**

Kubernetes is an open-source platform designed to automate deploying, scaling, and operating application containers. Originally developed by Google, it has become the standard for managing containerized workloads and services.

2. **Explain the concept of Container Orchestration?**

Container orchestration is a tool that developers may use anywhere there are containers to automate the life cycle management of the containers. It provides a automatic deployment, scaling, and management of containerized applications so that the developers do not have any worry about that the underlying infrastructure.

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

In Kubernetes, a Pod is the smallest deployable unit that you can create and manage. It represents a single instance of a containerized application. A Pod can contain one or more containers that are tightly coupled and share resources, such as networking and storage.

4. **How does Kubernetes handle container scaling?**

To automatically scale the workload to match demand, a Horizontal Pod Autoscaling in Kubernetes updates a workload resource (such a deployment or stateful set). Horizontal scaling indicates that more pods are added in response to an increase in load.

5. **What is Kubelet?**

Kubelet plays a critical role in the Kubernetes cluster, acting as the primary interface between the Kubernetes master and individual nodes, and ensuring that containerized workloads are effectively managed and maintained.

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

The idea of the Service is to group a set of Pod endpoints into a single resource. We can configure various ways to access the grouping. By default, we can get a stable cluster IP address that the clients inside the cluster can use to contact Pods in Service.

7. **What is the role of the kube-proxy in Kubernetes and how does it facilitate communication between Pods?**

Kube-proxy is another essential component of Kubernetes responsible for facilitating communication between Pods within a cluster and from external clients to Pods. Its primary role is to provide network proxy and load balancing services for traffic that flows into or out of a Kubernetes node.

8. Describe the role of etcd in Kubernetes?

Etcd is the cluster brain that maintains records of all cluster information, which includes the desired state, the current state, resource configurations, and runtime data.

9. What is a Namespace in Kubernetes?

In Kubernetes, a Namespace is a way to logically divide and isolate resources within a cluster. It provides a scope for Kubernetes objects, allowing multiple virtual clusters to coexist within the same physical cluster. Namespaces provide a level of isolation by segregating resources such as Pods, Services, Deployments, and other objects within a cluster

10.what is meta data in Kubernetes manifest file?

In Kubernetes YAML files, metadata is a section used to provide descriptive information about Kubernetes objects. It includes details such as the name, namespace, labels, annotations, and other identifying information for the object.

11.Write simple example for manifest file?

```
    apiVersion: v1
kind: Pod
metadata:
  name: mypod
  namespace: mynamespace
  labels:
    app: myapp
    environment: production
  annotations:
    description: "This is a sample pod for demonstration purposes"
```

12.Explain the use of Labels and Selectors in Kubernetes?

Labels are key-value pairs attached to Kubernetes objects (such as Pods, Services, Deployments, etc.) that provide additional metadata. They are used to identify and categorize objects based on arbitrary attributes.

Selectors are used to identify and select Kubernetes objects based on their labels. They allow you to define criteria for selecting resources that match specific label combinations. Here's how selectors are typically used:

13.Describe the role of a Proxy in Kubernetes?

In Kubernetes, a proxy, specifically kube-proxy, plays a crucial role in networking and service discovery within the cluster. Kube-proxy acts as a network proxy and load balancer for services running in a Kubernetes cluster. It is responsible for forwarding requests to the appropriate backend Pods based on service definitions.

14.What is a Persistent Volume (PV) in Kubernetes?

A Persistent Volume (PV) in Kubernetes is an object that allows pods to access storage from a defined device. This device is usually described via a Kubernetes StorageClass. When a PVC is created individually, it is generated and designated to the specified storage device.

15.What advantages does Kubernetes have?

- Container Orchestration
- Automated Load Balancing
- Auto Scaling
- Rolling Update & Rollbacks
- Service Discovery and Load Balancing
- Storage Orchestration
- Self-Healing
- Secrets and Configuration Management
- Role-Based Access Control (RBAC)
- Pods and Multi-Container Support
- Monitoring and Logging

16.Describe how a Horizontal Pod Autoscaler (HPA) works?

With a dedicated instance for each workflow, each configured Horizontal Pod Autoscaler works as part of a control loop, automatically changing the workloads' shape to maintain the desired state by periodically contrasting its metrics to the user-configured target thresholds.

17.Define daemon sets?

Daemon sets are a set of pods that runs on a host. They are used for host layers attributes like monitoring network or simple network.

18.Define Kubernetes controller manager?

The controller manager is a daemon used for garbage collection, core control loops, and namespace creation. It enables the running of more than one process on the master node.

19.Mention the types of controller managers

Types of controller managers are: 1) endpoints controller, 2) service accounts controller, 3) node controller, 4) namespace controller, 5) replication controller, 6) token controller.

20.Explain Kubernetes Architecture

- Master Node: The master node is the first and most vital component which is responsible for the management of Kubernetes cluster. It is the entry point for all kinds of administrative tasks. There may be more than one master node in the cluster to check for fault tolerance.
- API Server: The API server acts as an entry point for all the REST commands used for controlling the cluster.
- Scheduler: The scheduler schedules the tasks to the slave node. It stores the resource usage information for every slave node. It is responsible for distributing the workload.
- Etcd: etcd components, store configuration detail, and wright values. It communicates with the most component to receive commands and work. It also manages network rules and port forwarding activity.
- Worker/Slave nodes: Worker nodes are another essential component that contains all the required services to manage the networking between the containers, communicate with the master node, which allows you to assign resources to the scheduled containers.
- Kubelet: It gets the configuration of a Pod from the API server and ensures that the described containers are up and running.
- Docker Container: Docker container runs on each of the worker nodes, which runs the configured pods.
- Pods: A pod is a combination of single or multiple containers that logically run together on nodes.

21.What is the difference between Kubernetes and Docker Swarm?

Kubernetes	Docker Swarm
Kubernetes Provides an auto-scaling feature.	Docker Swarm does not provide an auto-scaling
Manually configure your load balancing settings.	Does auto load balancing
Installation is complicated & time-consuming.	Installation is easy & fast.
GUI is available.	GUI not available.