What is Kubernetes? / What do you understand by Kubernetes?

Kubernetes is an open-source container-orchestration tool or system used to automate tasks such as the management, monitoring, scaling, and deploying containerized applications.

---

2) What is the use of Kubernetes?

Kubernetes is mainly used to easily manage several containers (since it can handle the grouping of containers), which provides logical units that can be discovered and managed.

---

3) Who was the inventor of Kubernetes?

Kubernetes was initially designed and developed by Google and is now maintained by the Cloud Native Computing Foundation.

4) What are K8s?

K8s is nothing but just another term for Kubernetes.

---

5) What was the main motive behind the development of Kubernetes?

The main motive behind the development of Kubernetes is to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts."

---

6) What do you understand by the term orchestration when it comes to software and DevOps?

The term orchestration specifies integrating multiple services that allow them to automate processes or synchronize information in a specific time sequence.

For example, suppose we have six or seven microservices for an application to run, then if you place them in separate containers, this would inevitably create obstacles for communication. Using orchestration, we can do it quickly to enable all services in individual containers to work seamlessly to accomplish a single goal.

7) What is the relation between Docker and Kubernetes?

Docker is an open-source platform used to handle software development. It is mainly used to package the settings and dependencies that the software/application needs to run into a container, which allows for portability and several other advantages. On the other hand, Kubernetes is used to allow the manual linking and orchestration of several containers, running on multiple hosts that have been created using Docker.

8) What are the key differences between the Docker Swarm and Kubernetes?

Docker Swarm is an open-source container orchestration platform used to cluster and schedule Docker containers. It is a native of Docker. Following is the list of key differences between the Docker Swarm and Kubernetes:

Docker Swarm	Kubernetes
Docker Swarm is easy and convenient to set up, but it doesn't have a robust cluster.	Kubernetes is more complicated than Docker Swarm to set up, but it assures a robust cluster.
Docker Swarm can't do auto-scaling as the Kubernetes can do, but Docker's scaling is five times faster than Kubernetes.	Kubernetes can do auto-scaling, but scaling is slower than Docker Swarm.
Docker Swarm doesn't provide a GUI.	Kubernetes provides a GUI in the form of a dashboard.
Docker Swarm provides an automatic load balancing feature of traffic between containers in a cluster.	It requires manual intervention in Kubernetes for load balancing such traffic.
Docker requires third-party tools such as the ELK stack for logging and monitoring.	Kubernetes provides such integrated tools for logging and monitoring purposes.
In Docker Swarm, we can easily share storage volumes with any container.	In Kubernetes, we can only share storage volumes with containers in the same pod.
We can deploy rolling updates in Docker Swarm but can't deploy automatic rollbacks.	In Kubernetes, we can deploy rolling updates as well as automatic rollbacks.

9) What do you understand by a node in Kubernetes?

In Kubernetes, a node is the smallest unit of hardware. It is used to define a single machine in a cluster that can act as a virtual machine from a cloud provider or physical machine in the data center. Every machine of the Kubernetes cluster can act as a substitute for other machines.

---

10) What is the use of a Kubernetes Kube-scheduler?

A Kube-scheduler is the default scheduler for Kubernetes. It is used to assign nodes to newly created pods.

11) What do you understand by daemon sets in Kubernetes?

Daemon sets are sets of pods that run on a host and are used for host layers attributes like monitoring network or simple network.

---

12) What is Heapster in Kubernetes?

A Heapster is a metrics collection and performance monitoring system for data collected by the Kublet.

---

13) What are the main reasons behind using Kubernetes?

Kubernetes is mainly used because of the following reasons:

- We can use Kubernetes easily on on-premises bare metal, OpenStack, public clouds Google, Azure, AWS, etc.
- Using Kubernetes, we can avoid vendor lock issues as it can use any vendor-specific APIs or services except where Kubernetes provides an abstraction, e.g., load balancer and storage.
- It enables applications that need to be released and updated without any downtime.

---

14) What does the node status contain in Kubernetes?

In Kubernetes, the main components of a node status are Address, Condition, Capacity, and Info.

---

15) What is the Kubernetes Network Policy?

In Kubernetes, the Network Policy specifies how the same namespace's pods would communicate with each other and the network endpoint.