### **1. What is Kubernetes?**

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

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

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

Master

The master node is the most vital component of Kubernetes architecture. It is the entry point of all administrative tasks. There is always one node to check for fault tolerance.

The master node has various components, such as:

* ETCD
* Controller Manager
* Scheduler
* API Server
* Kubectl

#### 1. ETCD

* This component stores the configuration details and essential values
* It communicates with all other components to receive the commands to perform an action.
* Manages network rules and post-forwarding activity

#### 2. Controller Manager

* A daemon (server) that runs in a continuous loop and is responsible for gathering information and sending it to the API Server
* The key controllers are the replication controllers, endpoint controller, namespace controllers, and service account controllers
* The controller manager runs controllers to administer nodes and endpoints

#### 3. Scheduler

* The scheduler assigns the tasks to the slave nodes
* It is responsible for distributing the workload and stores resource usage information on every node
* Tracks how the working load is used on clusters and places the workload on available resources.

#### 4. API Server

* Kubernetes uses the API server to perform all operations on the cluster
* It is a central management entity that receives all REST requests for modifications, serving as a frontend to the cluster

#### 5. Kubectl

* Kubectl controls the Kubernetes cluster manager

Syntax - kubectl [flags]

* Pod: A group of containers
* Labels: Used to identify pods
* Kubelet: Container agents responsible for maintaining pod sets
* Etcd: A metadata service
* CAdvisor: Monitors resource performance and usage
* Replication controller: Manages pod replication
* Scheduler: Schedules pods in worker nodes

### 

### **3. What is a node in Kubernetes?**

A node is the smallest fundamental unit of computing hardware. It represents a single machine in a cluster, which could be a physical machine in a data center or a virtual machine from a cloud provider.

**4. What process runs on Kubernetes Master Node?**

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

### **5. What is a pod in Kubernetes?**

Pods are high-level structures that wrap one or more containers. This is because containers are not run directly in Kubernetes. Containers in the same pod share a local network and the same resources, allowing them to easily communicate with other containers in the same pod as if they were on the same machine while at the same time maintaining a degree of isolation.

### **6. What is the job of the kube-scheduler?**

The kube-scheduler assigns nodes to newly created pods.

### **7. What is a cluster of containers in Kubernetes?**

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

### **8. What is a Namespace in Kubernetes?**

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

### **9. Name the initial namespaces from which Kubernetes starts?**

* Default
* Kube – system
* Kube – public

### **10. What is the Kubernetes controller manager?**

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

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

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

**12. What is orchestration when it comes to software and DevOps?**

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

**13. How are Kubernetes and Docker related?**

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

**14. What are the main differences between the Docker Swarm and Kubernetes?**

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

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

**15. What is the difference between deploying applications on hosts and containers?**

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

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

**16. What are the main components of Kubernetes architecture?**

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

**17. Explain the working of the master node in Kubernetes?**

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

**18. What is the role of Kube-apiserver?**

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

**19. What is the job of the kube-scheduler?**

The kube-scheduler assigns nodes to newly created pods.

**20. What is a Namespace in Kubernetes?**

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

**21. What is the Kubernetes controller manager?**

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

**22. What are the different services within Kubernetes?**

Different types of Kubernetes services include:

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