

Kubernetes interview questions.

#### Question 1

**What is the difference Docker and Kubernetes?**

Docker is a container platform whereas Kubernetes is a container orchestration environment that offers capabilities like Auto healing, Auto Scaling, Clustering and Enterprise level support like Load balancing.

#### Question 2

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

On a broad level, you can divide the Kubernetes components in two parts

1. Control Plane (API SERVER, SCHEDULER, Controller Manager, C-CM, etcd)
2. Data Plane (Kubelet, Kube-proxy, Container Runtime)

#### Question 3

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

Kubernetes is better suited for large organizations as it offers more scalability, networking capabilities like policies and huge third-party ecosystem support.

#### Question 4

**What is the difference between Docker container and a Kubernetes pod?**

A pod in Kubernetes is a runtime specification of a container in docker. A pod provides more declarative way of defining using YAML and you can run more than one container in a pod.

#### Question 5

**What is a namespace in Kubernetes?**

In Kubernetes namespace is a logical isolation of resources, network policies, rbac and everything. For example, there are two projects using same k8s cluster. One project can use ns1 and other project can use ns2 without any overlap and authentication problems.

#### Question 6

**What is the role of kube proxy ?**

Kube-proxy works by maintaining a set of network rules on each node in the cluster, which are updated dynamically as services are added or removed. When a client sends a request to a service, the request is intercepted by kube-proxy on the node where it was received. Kube-proxy then looks up the destination endpoint for the service and routes the request accordingly.

Kube-proxy is an essential component of a Kubernetes cluster, as it ensures that services can communicate with each other.

#### Question 7

**What are the different types of services within Kubernetes?**

There are three different types of services that a user can create.

1. Cluster IP Mode
2. Node Port Mode
3. Load Balancer Mode

#### Question 8

**What is the difference between NodePort and LoadBalancer type service?**

When a service is created a NodePort type, The kube-proxy updates the Iptables with Node IP address and port that is chosen in the service configuration to access the pods.

Whereas if you create a Service as type Load Balancer, the cloud control manager, creates a external load balancer IP using the underlying cloud provider logic in the C-CM users can access services using the external IP.

#### Question 9

**What is the role of Kubelet?**

Kubelet manages the containers that are scheduled to run on that node. It ensures that the containers are running and healthy, and that the resources they need are available.

Kubelet communicates with the Kubernetes API server to get information about the containers that should be running on the node, and then starts and stops the containers as needed to maintain the desired state. It also monitors the containers to ensure that they are running correctly, and restarts them if necessary.

#### Question 10

Day to Day activities on Kubernetes

Absolutely, if you're asked about day-to-day activities on Kubernetes in an interview, here's a concise and structured response you could consider:

"In a typical day as a Kubernetes administrator/engineer, my responsibilities revolve around ensuring the smooth operation of the Kubernetes cluster and the applications running on it. This involves a range of tasks:

**1. Cluster Monitoring and Health:**

- Regularly monitor the cluster's health using tools like Prometheus and Grafana.
- Keep an eye on resource utilization to optimize cluster performance.
- Address any alerts or anomalies promptly to maintain stability.

**2. Application Deployment and Scaling:**

- Deploy new applications or updates using Kubernetes manifests or Helm charts.
- Scale application instances up or down based on traffic patterns and demand.

**3. Debugging and Troubleshooting:**

- Investigate and resolve any issues related to application deployments, networking, or performance.
- Utilize 'kubectl logs' to analyze container logs for debugging purposes.

**4. Configuration Management:**

- Manage application configurations using ConfigMaps and Secrets.
- Ensure seamless configuration updates without disrupting application availability.

**5. Security and Access Control:**

- Implement Role-Based Access Control (RBAC) to manage user permissions.
- Set up Network Policies to control communication between pods.
- Stay vigilant about security patches and updates.

**6. Load Balancing and Networking:**

- Configure Services to expose applications internally or externally.
- Manage Ingress controllers for efficient traffic routing.

#### 7. **Data Persistence and Management:**

- Oversee stateful applications using StatefulSets or specialized Operators.
- Manage data storage through Persistent Volumes and Persistent Volume Claims.

#### 8. **CI/CD Integration:**

- Integrate Kubernetes into the CI/CD pipeline to automate deployments.
- Ensure continuous delivery of updates and improvements.

#### 9. **Backup and Disaster Recovery:**

- Set up robust backup and recovery mechanisms to safeguard critical data.
- Regularly test disaster recovery plans to maintain readiness.

#### 10. **Resource Optimization:**

- Analyze and fine-tune resource requests and limits for optimal utilization.
- Implement resource quotas to prevent resource hogging.

#### 11. **Logging and Monitoring:**

- Establish centralized logging for efficient troubleshooting and auditing.
- Leverage monitoring tools to track application performance and cluster health.

My goal in these tasks is to guarantee the availability, reliability, and performance of the cluster and applications, contributing to the organization's efficient use of Kubernetes technology."

Remember, tailoring your response to match your actual experiences and the specific job description will make your answer even more impactful.



















