1st What is Kubernetes?

- Kubernetes is an open-source platform for automating deployment, scaling, and management of containerized applications.

2ndWhat is a pod in Kubernetes?

- A pod is the smallest and simplest unit in the Kubernetes object model. It represents a single instance of a running process in your application.

3rdWhat is a node in Kubernetes?

- A node is a worker machine in Kubernetes, which hosts the pods and runs the containers.

4th What is a cluster in Kubernetes?

- A cluster is a set of nodes that run containerized applications.

5th What is a replica set in Kubernetes?

- A replica set is a controller that ensures a specified number of replicas of a pod are running at any given time.

6th What is a deployment in Kubernetes?

- A deployment is a higher-level abstraction that manages replica sets and provides declarative updates to pods.

7th What is a service in Kubernetes?

- A service is an abstraction that defines a logical set of pods and a policy to access them.

8th What is a label in Kubernetes?

- A label is a key-value pair that is attached to objects, such as pods, to describe and categorize them.

9th What is a selector in Kubernetes?

- A selector is a mechanism to filter a set of objects based on their labels.

10thWhat is a namespace in Kubernetes?

- A namespace is a virtual cluster within a physical cluster, used to isolate resources and manage access to them.

11thWhat is a config map in Kubernetes?

- A config map is a way to store configuration data for your application that can be consumed as environment variables or command-line arguments.

12thWhat is a secret in Kubernetes?

- A secret is a way to store sensitive information, such as passwords or API keys, that can be consumed by pods.

13thWhat is an ingress in Kubernetes?

- An ingress is a way to expose HTTP and HTTPS routes from outside the cluster to services within the cluster.

14thWhat is a volume in Kubernetes?

- A volume is a persistent data store that can be attached to a pod, allowing data to persist even if the pod is deleted or recreated.

15thWhat is a stateful set in Kubernetes?

- A stateful set is a way to manage stateful applications, such as databases, in a predictable manner.

16thWhat is rolling update in Kubernetes?

- Rolling update is a mechanism to update the pods in a deployment one-by-one, without any disruption to the application.

17thWhat is a daemon set in Kubernetes?

- A daemon set is a way to ensure that a specific pod runs on all or a set of nodes in a cluster.

18thWhat is a job in Kubernetes?

- A job is a way to run a batch process in Kubernetes, ensuring that a specified number of pods complete successfully.

19thWhat is a cron job in Kubernetes?

- A cron job is a way to run a job on a schedule in Kubernetes, similar to a traditional Unix cron.

20thWhat is a scaling in Kubernetes?

- Scaling is the process of increasing or decreasing the number of replicas in a deployment, replica set, or stateful set.

21stHow does Kubernetes handle failures in the cluster?

- Kubernetes monitors the health of the cluster and automatically replaces failed pods or nodes, ensuring that the desired state of the cluster is maintained.

22ndWhat is horizontal pod autoscaling in Kubernetes?

- Horizontal pod autoscaling is a mechanism to automatically scale the number of replicas in a deployment based on CPU or memory utilization.

23rdWhat is vertical pod autoscaling in Kubernetes?

- Vertical pod autoscaling is a mechanism to automatically increase the CPU or memory resources available to a pod based on utilization.

24thWhat is an admission controller in Kubernetes?

- An admission controller is a plugin that implements admission control policies in Kubernetes, such as enforcing resource limits or security policies.

25thWhat is a custom resource in Kubernetes?

- A custom resource is a way to extend the Kubernetes API to store custom data in the cluster.

26thWhat is an operator in Kubernetes?

- An operator is a custom controller that extends the Kubernetes API to manage complex, stateful applications.

27thWhat is a kustomization in Kubernetes?

- A kustomization is a way to manage and reuse common configuration across multiple Kubernetes manifests.

28thWhat is a Helm chart in Kubernetes?

- A Helm chart is a package that contains definitions for creating and deploying complex applications in Kubernetes.

29thWhat is kubectl in Kubernetes?

- kubectl is the command-line interface for managing a Kubernetes cluster.

30thWhat is kubeadm in Kubernetes?

- kubeadm performs the actions necessary to get a minimum viable cluster up and running. By design, it cares only about bootstrapping, not about provisioning machines. Likewise, installing various nice-to-have addons, like the Kubernetes Dashboard, monitoring solutions, and cloud-specific addons, is not in scope.

31stWhat is Minikube in Kubernetes?

- Minikube is a tool that makes it easy to run a single-node Kubernetes cluster locally, for testing and development purposes.

32ndWhat is a namespace in Kubernetes?

- A namespace is a way to partition resources in a Kubernetes cluster, allowing different parts of an application to be isolated from each other.

33rdWhat is a label in Kubernetes?

- A label is a key-value pair that can be attached to objects in a Kubernetes cluster, used for organization, filtering, and selection.

34thWhat is an annotation in Kubernetes?

- An annotation is a key-value pair that can be attached to objects in a Kubernetes cluster, used to store arbitrary non-identifying metadata.

35thWhat is a selector in Kubernetes?

- A selector is a way to filter objects in a Kubernetes cluster based on the values of their labels.

36thWhat is a resource quota in Kubernetes?

- A resource quota is a way to limit the amount of resources (such as CPU, memory, or storage) that can be consumed by objects in a namespace.

37thWhat is a network policy in Kubernetes?

- A network policy is a way to define access control rules for network traffic in a Kubernetes cluster, allowing or denying communication between pods.

38thWhat is a config map in Kubernetes?

- A config map is a way to store configuration data for use by pods in a Kubernetes cluster, without embedding it in the pod specification.

39thWhat is a secret in Kubernetes?

- A secret is a way to store sensitive data, such as passwords or SSL certificates, for use by pods in a Kubernetes cluster, without exposing it in the pod specification.

40thWhat is a persistent volume in Kubernetes?

- A persistent volume is a way to provide storage to pods in a Kubernetes cluster that persists beyond the lifetime of a pod.

41stWhat is a persistent volume claim in Kubernetes?

- A persistent volume claim is a request for storage by a pod in a Kubernetes cluster, matching with a persistent volume that satisfies the claim's requirements.

42ndWhat is a stateful set in Kubernetes?

- A stateful set is a way to deploy stateful applications in a Kubernetes cluster, providing stable network identities and persistent storage to pods.

43rdWhat is a deployment in Kubernetes?

- A deployment is a way to manage and update replicas of a pod in a Kubernetes cluster, ensuring that the desired state is maintained.

44thWhat is a replica set in Kubernetes?

- A replica set is a way to ensure that a specified number of replicas of a pod are running in a Kubernetes cluster, automatically replacing failed pods.

45thWhat is a pod in Kubernetes?

- A pod is the smallest and simplest unit in the Kubernetes object model, representing a single instance of a running process in a cluster.

46thWhat is a service in Kubernetes?

- A service is a way to expose pods to the network in a Kubernetes cluster, providing stable network identities, load balancing, and network access.

SITUATION BASED QUESTIONS.

Q.You have a deployment in your cluster that has been running for a while and now you need to update the image being used. How would you go about doing this update?
Answer: To update the image being used in a deployment, you would edit the deployment YAML file to use the new image tag or version. Then, you would use the kubectl apply command to apply the changes to the cluster, which would trigger a rolling update of the pods in the deployment, replacing the old pods with new pods using the updated image.

Q.You have a deployment in your cluster that is running low on resources, how would you go about increasing the resources being used by the deployment's pods?
Answer: To increase the resources being used by a deployment's pods, you would edit the deployment YAML file to specify the desired resource requests and limits for the pods. Then, you would use the kubectl apply command to apply the changes to the cluster, which would trigger a rolling update of the pods in the deployment, replacing the old pods with new pods using the updated resource requirements.

Q.You need to add a new environment variable to the containers in a deployment, how would you go about doing this?
Answer: To add a new environment variable to the containers in a deployment, you would edit the deployment YAML file to add the new environment variable to the list of environment variables for the containers. Then, you would use the kubectl apply command to apply the changes to the cluster, which would trigger a rolling update of the pods in the deployment, replacing the old pods with new pods using the updated environment variables.

Q.You have a service in your cluster that is not responding as expected, how would you go about troubleshooting the issue?
Answer: To troubleshoot a service issue in a cluster, you would first check the logs of the pods backing the service to see if there are any error messages or stack traces. You could use the kubectl logs command to view the logs. If the issue is related to network connectivity, you could check the status of the service and its endpoints using the kubectl get svc and kubectl describe svc commands, respectively. If necessary, you could also check the network policy and routing rules in the cluster to ensure that the service and its pods are accessible.

Q.You have a deployment that is running slowly and you need to debug why. What steps would you take to diagnose the issue?
Answer: To diagnose a slow deployment, you would first check the resource utilization of the pods in the deployment, such as CPU and memory usage, to see if they are being over-utilized. You could use the kubectl top command to view the resource utilization of pods. If the pods are over-utilized, you may need to scale the deployment or increase the resources allocated to the pods. If the pods are not over-utilized, you could look at the logs of the pods to see if there are any error messages or performance issues. You could also check the network connectivity between the pods and other components in the cluster.

Q.You have a deployment that is failing to start due to a configuration error. What steps would you take to diagnose and fix the issue?
Answer: To diagnose a failing deployment, you would first check the logs of the pods to see if there are any error messages or stack traces indicating the cause of the failure. You could use the kubectl logs command to view the logs. If there is a configuration error, you would need to edit the deployment YAML file to fix the error and apply the changes to the cluster using the kubectl apply command. If the logs do not provide enough information to diagnose the issue, you could also check the status of the pods using the kubectl get pods command and the events related to the pods using the kubectl describe pods command.

Q.You have a deployment that is crashing frequently and you need to find the cause. What steps would you take to diagnose the issue?
Answer: To diagnose a frequently crashing deployment, you would first check the logs of the pods to see if there are any error messages or stack traces indicating the cause of the crashes. You could use the kubectl logs command to view the logs. If the logs do not provide enough information, you could also check the status of the pods using the kubectl get pods command and the events related to the pods using the kubectl describe pods command. If the issue is related to resource utilization, you could check the resource utilization of the pods using the kubectl top command. If necessary, you could also check the network policy and routing rules in the cluster to ensure that the pods are accessible.

Q.You have a deployment that is working as expected, but you want to make a change to it. How would you go about making the change while minimizing downtime?
Answer: To make a change to a deployment while minimizing downtime, you would typically perform a rolling update, which updates the pods in the deployment one-by-one instead of all at once. To perform a rolling update, you would edit the deployment YAML file to reflect the desired change, then use the kubectl apply command to apply the changes to the cluster. Kubernetes would then automatically perform a rolling update, replacing the old pods with new pods with the desired change. You could also use other strategies such as blue-green deployment or canary deployment, but these methods are more complex and may require additional components such as service load balancers.

Q.You have a deployment that is working as expected, but you need to scale it up to handle increased traffic. How would you go about scaling the deployment?
Answer: To scale a deployment, you would simply update the replica count in the deployment YAML file to the desired number of replicas, then use the kubectl apply command to apply the changes to the cluster. Kubernetes would then automatically create or terminate pods as necessary to match the desired replica count. You could also use the kubectl scale command to change the replica count of a deployment directly, without editing the YAML file.

Q.You have a deployment that is not responding and you need to restart it. How would you go about restarting the deployment?
Answer: To restart a deployment, you would first use the kubectl get pods command to view the pods in the deployment, then use the kubectl delete pod command to delete one or more of the pods. Kubernetes would then automatically recreate the deleted pods, effectively restarting the deployment. You could also perform a rolling update of the deployment, as described in question 8, which would also result in the pods being restarted. However, this method is generally more time-consuming and may result in more downtime than simply deleting and recreating the pods.