1. What is Kubernetes?

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications.

2. What is a container in Kubernetes?

A container is a lightweight, standalone software package that contains everything needed to run an application, including code, runtime, libraries, and settings.

3. What is a Kubernetes Pod?

A Pod is the smallest deployable unit in Kubernetes, representing a single instance of a running process or application.

4. Explain the purpose of a Kubernetes ReplicaSet.

A ReplicaSet ensures that a specified number of replicas (identical instances) of a Pod are running at all times to maintain high availability.

5. How does Kubernetes achieve container orchestration?

Kubernetes automates the deployment, scaling, and management of containers across clusters of hosts.

6. What is a Kubernetes Deployment?

A Deployment is a higher-level resource that manages ReplicaSets and provides declarative updates to applications.

7. How do you define Kubernetes resources like Pods or Deployments?

Resources are defined using YAML files, which describe the desired state of the resource.

8. What is a Kubernetes Service?

A Service provides a stable network endpoint to connect to a set of Pods, allowing for load balancing and discovery.

9. What is a Kubernetes Namespace?

A Namespace provides a way to logically divide a Kubernetes cluster into virtual clusters to isolate resources and manage access.

10. How do you scale a Kubernetes Deployment?

- Use the kubectl scale command or modify the replicas field in the Deployment's YAML file.

11. What is a Kubernetes ConfigMap?

- A ConfigMap is a Kubernetes resource that stores configuration data in key-value pairs and can be consumed by Pods.

12. What is a Kubernetes Secret?

- A Secret is a Kubernetes resource used to store sensitive data, such as passwords or tokens, in an encrypted format.

13. How do you update a Kubernetes Deployment?

- Update the Deployment's YAML file with the desired changes and use kubectl apply to apply the changes.

14. Explain the concept of Kubernetes Nodes.

- Nodes are individual machines (virtual or physical) in a Kubernetes cluster where containers are deployed and managed.

15. What is a Kubernetes Master Node?

- The Master Node controls the Kubernetes cluster, managing its overall state, scheduling, and orchestration.

16. How does Kubernetes handle container networking?

- Kubernetes assigns a unique IP address to each Pod and provides service discovery and load balancing.

17. What is a Kubernetes Ingress Controller?

- An Ingress Controller manages external access to services in a Kubernetes cluster, typically handling HTTP traffic.

18. How do you troubleshoot a misbehaving Pod in Kubernetes?

- Use kubectl logs to view container logs, and kubectl describe pod to gather more information about the Pod's state.

19. What is a Kubernetes StatefulSet?

- A StatefulSet manages stateful applications by providing stable network identities and persistent storage for each Pod.

20. How can you mount a volume in a Kubernetes Pod?

- Define a volume in the Pod's YAML and then reference it in the container's volume mounts.

21. What is Kubernetes Helm?

- Helm is a package manager for Kubernetes that simplifies the deployment and management of applications using predefined charts.

22. How does Kubernetes handle rolling updates?

- Kubernetes performs rolling updates by gradually replacing old Pods with new ones, ensuring minimal downtime.

23. What is Kubernetes' Horizontal Pod Autoscaler (HPA)?

- The HPA automatically adjusts the number of replicas in a Deployment based on CPU utilization or custom metrics.

24. What is Kubernetes' Vertical Pod Autoscaler (VPA)?

- The VPA adjusts the resource requests and limits for Pods based on actual usage to optimize resource allocation.

25. How do you expose a Kubernetes Deployment outside the cluster?

- Use a Service with a Type of LoadBalancer or create an Ingress resource.

26. What is Kubernetes' Cluster Autoscaler?

- The Cluster Autoscaler dynamically adjusts the size of a Kubernetes cluster by adding or removing nodes as needed.

27. How do you upgrade Kubernetes itself?

- Upgrade Kubernetes by upgrading the cluster components, like the kubelet and control plane, following the official documentation.

28. What is a Kubernetes DaemonSet?

- A DaemonSet ensures that a copy of a specific Pod runs on all or selected nodes in the cluster.

29. How do you roll back a failed Kubernetes Deployment update?

- Use kubectl rollout undo deployment/deployment-name to roll back to the previous version.

30. What is Kubernetes' Job resource used for?

- A Job creates one or more Pods and ensures they run to completion, often used for batch processing or one-time tasks.

31. How does Kubernetes handle scheduling and placement of Pods?

- Kubernetes uses a scheduler to place Pods on appropriate nodes based on resource requirements, node availability, and other factors.

32. Explain the concept of Kubernetes' Taints and Tolerations.

- Taints are applied to nodes to repel Pods. Tolerations are set on Pods to allow them to tolerate specific taints.

33. What is a Kubernetes Namespace used for?

- A Namespace provides a way to partition and isolate resources within a cluster, aiding in multi-tenancy and resource management.

34. What is Kubernetes' ResourceQuota?

- A ResourceQuota limits the resource consumption of objects in a Namespace, preventing resource contention.

35. How do you upgrade a Kubernetes Deployment without downtime?

- Use rolling updates to gradually replace old Pods with new ones, maintaining application availability.

36. What is Kubernetes' Horizontal Pod Autoscaler (HPA)?

- The HPA automatically adjusts the number of replicas in a Deployment based on CPU utilization or custom metrics.

37. What is Kubernetes' Vertical Pod Autoscaler (VPA)?

- The VPA adjusts the resource requests and limits for Pods based on actual usage to optimize resource allocation.

38. How do you expose a Kubernetes Deployment outside the cluster?

- Use a Service with a Type of LoadBalancer or create an Ingress resource.

39. What is Kubernetes' Cluster Autoscaler?

- The Cluster Autoscaler dynamically adjusts the size of a Kubernetes cluster by adding or removing nodes as needed.

40. How do you upgrade Kubernetes itself?

- Upgrade Kubernetes by upgrading the cluster components, like the kubelet and control plane, following the official documentation.

41. What is a Kubernetes DaemonSet?

- A DaemonSet ensures that a copy of a specific Pod runs on all or selected nodes in the cluster.

42. How do you roll back a failed Kubernetes Deployment update?

- Use kubectl rollout undo deployment/deployment-name to roll back to the previous version.

43. What is Kubernetes' Job resource used for?

- A Job creates one or more Pods and ensures they run to completion, often used for batch processing or one-time tasks.

44. How does Kubernetes handle scheduling and placement of Pods?

- Kubernetes uses a scheduler to place Pods on appropriate nodes based on resource requirements, node availability, and other factors.

45. Explain the concept of Kubernetes' Taints and Tolerations.

- Taints are applied to nodes to repel Pods. Tolerations are set on Pods to allow them to tolerate specific taints.

46. What is a Kubernetes Namespace used for?

- A Namespace provides a way to partition and isolate resources within a cluster, aiding in multi-tenancy and resource management.

47. What is Kubernetes' ResourceQuota?

- A ResourceQuota limits the resource consumption of objects in a Namespace, preventing resource contention.

48. How do you upgrade a Kubernetes Deployment without downtime?

- Use rolling updates to gradually replace old Pods with new ones, maintaining application availability.

49. What is Kubernetes' Horizontal Pod Autoscaler (HPA)?

- The HPA automatically adjusts the number of replicas in a Deployment based on CPU utilization or custom metrics.

50. What is Kubernetes' Vertical Pod Autoscaler (VPA)?

- The VPA adjusts the resource requests and limits for Pods based on actual usage to optimize resource allocation.