

Kubernetes Fundamentals

Kubernetes Resources

Question 1

Which of the following is a built-in Kubernetes resource for scheduling pods on nodes?

- (A) Deployment
- (B) Ingress
- (C) Service
- (D) DaemonSet

Answer

(D) DaemonSet

Question 2

What Kubernetes resource is used to expose an application running on a set of pods as a network service?

- (A) ReplicaSet
- (B) Job
- (C) Service
- (D) CronJob

Answer

(C) Service

Kubernetes Architecture

Question 1

What is the primary component of a Kubernetes control plane that manages API requests?

- (A) Kube-scheduler
- (B) Kube-controller-manager
- (C) Etcd
- (D) Kube-apiserver

Answer

(D) Kube-apiserver

Question 2

Which component of a Kubernetes node is responsible for managing network rules and IP addresses?

- (A) Kubelet
- (B) Kubectl
- (C) Kube-proxy
- (D) Container runtime

Answer

(C) Kube-proxy

Kubernetes API**Question 1**

Which component of the Kubernetes control plane exposes the RESTful API that serves as the primary entry point for managing the cluster?

- (A) kube-scheduler
- (B) kube-controller-manager
- (C) kube-apiserver
- (D) etcd

Answer

(C) kube-apiserver

Question 2

How does the Kubernetes API ensure that different clients can interact with the cluster autonomously and in a consistent manner?

- (A) By using YAML configuration files exclusively
- (B) Through a single and uniform endpoint known as the kube-apiserver
- (C) By integrating with third-party authentication services only
- (D) By implementing a versioned set of API objects and operations

Answer

(B) Through a single and uniform endpoint known as the kube-apiserver **(D)** By implementing a versioned set of API objects and operations

Containers

Question 1

What is the primary advantage of using containers in application deployment?

- (A) They allow the hardware to operate at a higher efficiency.
- (B) They isolate applications and their dependencies.
- (C) They increase the size of the application bundle.
- (D) They convert applications into virtual machines.

Answer

(B) They isolate applications and their dependencies.

Question 2

Which of the following statements about containers is true?

- (A) Containers always require a specific operating system to run.
- (B) Containers can run consistently across different computing environments.
- (C) Containers are the same as virtual machines.
- (D) Containers only support monolithic applications.

Answer

(B) Containers can run consistently across different computing environments.

Scheduling

Question 1

Which of the following is responsible for determining which nodes can run a pod in Kubernetes?

- (A) Kubelet.
- (B) Kube-scheduler.
- (C) Pod controller.
- (D) Node selector.

Answer

(B) Kube-scheduler.

Question 2

What role does a node's taint play in Kubernetes scheduling?

- (A) It causes pods with matching tolerations to be allocated to the node.
- (B) It prevents any pods from being scheduled on the node.
- (C) It enhances the node's performance for specific applications.
- (D) It limits the amount of resources available to a pod.

Answer

(A) It causes pods with matching tolerations to be allocated to the node.

Container Orchestration

Container Orchestration Fundamentals

Question 1

What is the primary function of a container orchestration platform like Kubernetes?

- (A) It monitors network traffic and protects against attacks.
- (B) It manages the lifecycle of containers, including their deployment, scaling, and operation.
- (C) It provides a runtime environment for compiling code.
- (D) It offers a platform for cloud storage solutions.

Answer

(B) It manages the lifecycle of containers, including their deployment, scaling, and operation.

Question 2

Which component in Kubernetes is responsible for maintaining the desired state of applications?

- (A) Controller Manager
- (B) Kubelet
- (C) Etcd
- (D) Docker Daemon

Answer

(A) Controller Manager

Runtime

Question 1

What is the primary role of a container runtime in a Kubernetes environment?

- (A) To build container images from scratch.
- (B) To run containers on a node.
- (C) To schedule pods across the cluster.
- (D) To monitor the health of nodes.

Answer

(B) To run containers on a node.

Question 2

Which of the following container runtime options is supported by Kubernetes natively?

- (A) Docker Engine
- (B) Kubernetes CRI (Container Runtime Interface) implementations
- (C) Hyper-V
- (D) VMware vSphere

Answer

(A) Docker Engine **(B)** Kubernetes CRI (Container Runtime Interface) implementations

Security

Question 1

Which of the following are best practices for securing a Kubernetes cluster?

- (A) Use Role-Based Access Control (RBAC).
- (B) Grant admin privileges to all users.
- (C) Disable network policies.
- (D) Regularly audit the cluster.

Answer

(A) Use Role-Based Access Control (RBAC). **(D)** Regularly audit the cluster.

Question 2

What is the purpose of Pod Security Policies in Kubernetes?

- (A) To manage network traffic between pods.
- (B) To enforce rules on how pods are allowed to execute.
- (C) To scale pods automatically.
- (D) To manage storage resources for pods.

Answer

(B) To enforce rules on how pods are allowed to execute.

Networking

Question 1

Which networking model does Kubernetes primarily use to enable communication between Pods?

- (A) Network Address Translation (NAT)
- (B) Virtual Private Network (VPN)
- (C) Flat networking model
- (D) Peer-to-peer network

Answer

(C) Flat networking model

Question 2

In Kubernetes networking, which of the following is responsible for exposing services to external traffic?

- (A) DaemonSet
- (B) ConfigMap
- (C) Ingress
- (D) PersistentVolume

Answer

(C) Ingress

Service Mesh

Question 1

Which of the following are key features provided by a service mesh?

- (A) Load balancing and service discovery.
- (B) Communication security (e.g., mTLS).
- (C) Container image storage.
- (D) CI/CD pipeline automation.

Answer

(A) Load balancing and service discovery. **(B)** Communication security (e.g., mTLS).

Question 2

How do service meshes benefit microservices architectures?

- (A) They provide network-level abstractions for service-to-service communication.
- (B) They manage centralized logging collection for applications.
- (C) They eliminate the need for Docker containers.
- (D) They provide automatic scaling of server resources.

Answer

(A) They provide network-level abstractions for service-to-service communication.

Storage

Question 1

Which of the following Kubernetes resources is used to manage persistent data storage?

- (A) Node
- (B) PersistentVolume
- (C) ServiceAccount
- (D) ConfigMap

Answer

(B) PersistentVolume

Question 2

In Kubernetes, which of the following is true about a PersistentVolumeClaim (PVC)?

- (A) It automatically deletes the persistent volume when it's released.
- (B) It is a request for storage by a user.
- (C) It requires manual attachment to pods by cluster administrators.
- (D) It describes the specifics of how storage can be accessed from within the container.

Answer

(B) It is a request for storage by a user.

Cloud Native Architecture

Autoscaling

Question 1

What is the purpose of the Horizontal Pod Autoscaler in Kubernetes?

- (A) To increase the number of nodes in a cluster based on CPU utilization.
- (B) To scale the number of pods in a deployment based on observed metrics.
- (C) To upgrade the Kubernetes API server automatically.
- (D) To provide persistent storage for applications.

Answer

(B) To scale the number of pods in a deployment based on observed metrics.

Question 2

Which component is responsible for monitoring resource utilization and scaling decisions in Kubernetes Autoscaling?

- (A) Kube-scheduler
- (B) Controller Manager
- (C) Metrics Server
- (D) Kubelet

Answer

(C) Metrics Server

Serverless

Question 1

Which of the following is a key characteristic of serverless computing?

- (A) Requires manual scaling of resources.
- (B) Charges based on consumed compute time.
- (C) Necessitates extensive server management.
- (D) Requires the user to explicitly start and stop server instances.

Answer

(B) Charges based on consumed compute time.

Question 2

In the context of serverless architecture, what does FaaS stand for?

- (A) Function as a Service.
- (B) File as a Service.
- (C) Fragmentation as a Service.
- (D) Framework as a Service.

Answer

(A) Function as a Service.

Community and Governance

Question 1

Which foundation is responsible for the governance of the Kubernetes project?

- (A) Apache Software Foundation
- (B) Eclipse Foundation
- (C) Cloud Native Computing Foundation (CNCF)
- (D) Free Software Foundation

Answer

(C) Cloud Native Computing Foundation (CNCF)

Question 2

What role does the Kubernetes Steering Committee play in the project?

- (A) They oversee the technical direction of Kubernetes development.
- (B) They manage end-user support forums.
- (C) They coordinate the release cycles and schedules.
- (D) They handle the project's code review and merge requests.

Answer

(A) They oversee the technical direction of Kubernetes development.

Roles and Personas

Question 1

Which role in a cloud-native environment is primarily responsible for managing the architecture and ensuring the infrastructure supports application scalability and reliability?

- (A) Developer
- (B) DevOps Engineer
- (C) System Administrator
- (D) Security Specialist

Answer

(B) DevOps Engineer

Question 2

Who among the following is mainly concerned with the security aspects of cloud-native applications, focusing on vulnerability assessments and compliance?

- (A) Developer
- (B) Cloud Architect
- (C) Security Specialist
- (D) Site Reliability Engineer

Answer

(C) Security Specialist

Open Standards

Question 1

Which of the following best describes the role of open standards in cloud-native architecture?

- (A) They are proprietary protocols used by specific vendors.
- (B) They ensure interoperability between different systems and platforms.
- (C) They provide a way to lock-in customers to a specific technology stack.
- (D) They are closed specifications available only to members of certain organizations.

Answer

(B) They ensure interoperability between different systems and platforms.

Question 2

Why are open standards important in the development and deployment of cloud-native applications?

- (A) They limit the portability and scalability of applications.
- (B) They facilitate innovation and collaboration within the community.
- (C) They mandate the use of a specific programming language.
- (D) They decrease the security of applications by making them publicly available.

Answer

(B) They facilitate innovation and collaboration within the community.

Cloud Native Observability

Telemetry & Observability

Question 1

Which of the following tools is commonly used for collecting Kubernetes metrics and visualizing them?

- (A) Docker
- (B) Grafana
- (C) GitHub
- (D) Ansible

Answer

(B) Grafana

Question 2

What is the primary purpose of Prometheus in a Kubernetes environment?

- (A) Container deployment
- (B) Source code management

- (C) Monitoring and alerting
- (D) Network security

Answer

(C) Monitoring and alerting

Prometheus

Question 1

Which of the following best describes Prometheus in the context of cloud-native observability?

- (A) A programming language for building scalable cloud applications.
- (B) A distributed tracing tool.
- (C) A monitoring and alerting toolkit.
- (D) A tool for container orchestration.

Answer

(C) A monitoring and alerting toolkit.

Question 2

Which component of Prometheus is responsible for retrieving metrics data from configured targets?

- (A) Alertmanager
- (B) Prometheus Server
- (C) Pushgateway
- (D) Grafana

Answer

(B) Prometheus Server

Cost Management

Question 1

Which tool is often used for monitoring and managing the cost efficiency of cloud-native applications deployed on Kubernetes?

- (A) Prometheus.
- (B) Kubecost.

- (C) Jenkins.
- (D) GitLab.

Answer

(B) Kubecost.

Question 2

What approach can help optimize cloud-native resources in Kubernetes to improve cost management?

- (A) Over-provisioning resources.
- (B) Utilizing resource quotas.
- (C) Disabling autoscaling.
- (D) Running workloads in the development cluster.

Answer

(B) Utilizing resource quotas.

Cloud Native Application Delivery

Application Delivery Fundamentals

Question 1

Which of the following tools is primarily used for managing containerized applications across multiple hosts in production environments?

- (A) Jenkins
- (B) Kubernetes
- (C) Docker
- (D) Ansible

Answer

(B) Kubernetes

Question 2

What is a significant advantage of using Helm in Kubernetes environments?

- (A) It is a container runtime engine.
- (B) It provides a way to group and manage related Kubernetes resources.

- (C) It is a continuous integration tool.
- (D) It functions as a network policy engine in Kubernetes.

Answer

(B) It provides a way to group and manage related Kubernetes resources.

GitOps

Question 1

What is GitOps primarily used for in the context of Cloud Native Application Delivery?

- (A) To manage cloud storage services.
- (B) To automate the deployment of applications using Git as the single source of truth.
- (C) To provide security services for cloud applications.
- (D) To build container images from source code.

Answer

(B) To automate the deployment of applications using Git as the single source of truth.

Question 2

Which of the following are key principles of GitOps?

- (A) Declarative descriptions and versioned and immutable storage.
- (B) Building web applications with serverless functions.
- (C) Using a centralized database management system.
- (D) Real-time collaboration between development teams.

Answer

(A) Declarative descriptions and versioned and immutable storage.

CI/CD

Question 1

Which of the following best describes the primary purpose of Continuous Integration (CI) in cloud native applications?

- (A) To automate the testing and building of code.
- (B) To manage container orchestration.

- (C) To deploy applications to production.
- (D) To handle infrastructure monitoring.

Answer

(A) To automate the testing and building of code.

Question 2

What is a common advantage of implementing Continuous Deployment (CD) in a cloud-native environment?

- (A) It eliminates the need for version control systems.
- (B) It allows for automated rollback in case of failures.
- (C) It reduces the need for automated testing.
- (D) It requires less collaboration among team members.

Answer

(B) It allows for automated rollback in case of failures.