



Free Questions for **KCNA**

Shared by **Ortiz** on **20-10-2022**

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)



Question 1

Question Type: MultipleChoice

Which component of the kubernetes control-plane (master) are all requests to deploy and manage objects posted to?

Options:

- A- ETCD
- B- Controller Manager
- C- Kube-proxy
- D- API Server
- E- Kubelet



Answer:

D

Explanation:

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/>

Synopsis

The Kubernetes API server validates and configures data for the api objects which include pods, services, replicationcontrollers, and others. The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.

```
kube-apiserver [flags]
```



Question 2

Question Type: MultipleChoice

What is not semantic versioning?

Options:

- A- 1.0.0
- B- 2022-05-04
- C- 1.0.0-alpha
- D- 1.0.0-beta.2

Answer:

B

Explanation:

<https://semver.org/>

RegEx SemVer at <https://regex101.com/r/vkijKf/1/>



Question 3

Question Type: MultipleChoice

Continuous delivery is _____.

Options:

- A- Manually deploying the code
- B- Coding, Building and Testing the code
- C- Automatically deploying code to [container or server] environment

Answer:

C

Question 4

Question Type: MultipleChoice

What is autoscaling?

Options:

- A- Automatically measuring resource usage
- B- Automatically assigning workloads to nodes in a cluster
- C- Automatically repairing broken application instances
- D- Automatically adding or removing compute resources as needed

Answer:

D

Explanation:

<https://kubernetes.io/blog/2016/07/autoscaling-in-kubernetes/>

Autoscaling means automatically scaling up or down in response to real-time usage data.

Question 5

Question Type: MultipleChoice

What is container orchestration?

Options:

- A- Packaging code and all of its dependencies into a single executable
- B- Adding code to a container image so it can run as a container
- C- Using automation to manage containers
- D- Spinning a new containers to replace old ones

Answer:

C

Explanation:

<https://www.redhat.com/en/topics/containers/what-is-container-orchestration>

Container orchestration automates the deployment, management, scaling, and networking of containers. Enterprises that need to deploy and manage hundreds or thousands of **Linux® containers** and hosts can benefit from container orchestration.

Container orchestration can be used in any environment where you use containers. It can help you to deploy the same application across different environments without needing to redesign it. And **microservices** in containers make it easier to orchestrate services, including storage, networking, and security.

Question 6

Question Type: MultipleChoice

Which kubernetes resource type allows defining which pods are isolated when it comes to network-ing?

Options:

- A- Network policy
- B- Domain Name System 'DNS'
- C- Role Binding
- D- Service

Answer:

A

Explanation:

<https://kubernetes.io/docs/concepts/services-networking/network-policies/#the-two-sorts-of-pod-isolation>

The Two Sorts of Pod Isolation

There are two sorts of isolation for a pod: isolation for egress, and isolation for ingress. They concern what connections may be established. "Isolation" here is not absolute, rather it means "some restrictions apply". The alternative, "non-isolated for \$direction", means that no restrictions apply in the stated direction. The two sorts of isolation (or not) are declared independently, and are both relevant for a connection from one pod to another.

By default, a pod is non-isolated for egress; all outbound connections are allowed. A pod is isolated for egress if there is any NetworkPolicy that both selects the pod and has "Egress" in its `policyTypes`; we say that such a policy applies to the pod for egress. When a pod is isolated for egress, the only allowed connections from the pod are those allowed by the `egress` list of some NetworkPolicy that applies to the pod for egress. The effects of those `egress` lists combine additively.

By default, a pod is non-isolated for ingress; all inbound connections are allowed. A pod is isolated for ingress if there is any NetworkPolicy that both selects the pod and has "Ingress" in its `policyTypes`; we say that such a policy applies to the pod for ingress. When a pod is isolated for ingress, the only allowed connections into the pod are those from the pod's node and those allowed by the `ingress` list of some NetworkPolicy that applies to the pod for ingress. The effects of those `ingress` lists combine additively.



To Get Premium Files for KCNA Visit

<https://www.p2pexams.com/products/kcna>

For More Free Questions Visit

<https://www.p2pexams.com/linux-foundation/pdf/kcna>

20%
DISCOUNT

P2P
exams