



KCNA^{Q&As}

Kubernetes and Cloud Native Associate (KCNA)

Pass Linux Foundation KCNA Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/kcna.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

What Linux feature is used to provide isolation for containers?

- A. Processes
- B. Services
- C. NetworkPolicy
- D. Control groups

Correct Answer: D

Explanation: Control groups provide isolation for container processes, keeping them separate from other process-es on the host.

QUESTION 2

What is the command used to scale the application?

- A. kubectl run
- B. kubectl explain
- C. kubectl scale

Correct Answer: C

Explanation: [https://kubernetes.io/docs/reference/generated/kubectl/kubectl- commands#scale](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#scale)



scale

Set a new size for a deployment, replica set, replication controller, or stateful set.

Scale also allows users to specify one or more preconditions for the scale action.

If `--current-replicas` or `--resource-version` is specified, it is validated before the scale is attempted, and it is guaranteed that the precondition holds true when the scale is sent to the server.

Usage

```
$ kubectl scale [--resource-version=version] [--current-replicas=count] --replicas=COUNT (-f FILENAME | TYPE NAME)
```

example

Scale a replica set named 'foo' to 3

```
kubectl scale --replicas=3 rs/foo
```

Scale a resource identified by type and name specified in "foo.yaml" to 3

```
kubectl scale --replicas=3 -f foo.yaml
```

If the deployment named mysql's current size is 2, scale mysql to 3

```
kubectl scale --current-replicas=2 --replicas=3 deployment/mysql
```

Scale multiple replication controllers

```
kubectl scale --replicas=5 rc/foo rc/bar rc/c
```

QUESTION 3

What is the most common way to scale the application in the cloud environment?

- A. Parallel Scaling
- B. Horizontal Scaling
- C. Vertical Scaling

Correct Answer: B

Explanation: <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>

QUESTION 4

What is a benefits of Kubernetes federation?

- A. Avoids scalability limits on pods and nodes
- B. Creates highly available clusters in different regions
- C. Low latency

Correct Answer: ABC



QUESTION 5

Which access control component of Kubernetes is responsible for authorization and decides what requestor is allowed to do?

- A. Service Account
- B. Role-based access control \\RBAC\\
- C. Deployment

Correct Answer: B

Explanation: <https://kubernetes.io/docs/reference/access-authn-authz/authorization/>



Authorization Modes

The Kubernetes API server may authorize a request using one of several authorization modes:

- **Node** - A special-purpose authorization mode that grants permissions to kubelets based on the pods they are scheduled to run. To learn more about using the Node authorization mode, see [Node Authorization](#).
- **ABAC** - Attribute-based access control (ABAC) defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, object, environment attributes, etc). To learn more about using the ABAC mode, see [ABAC Mode](#).
- **RBAC** - Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file. To learn more about using the RBAC mode, see [RBAC Mode](#)
 - When specified RBAC (Role-Based Access Control) uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing admins to dynamically configure permission policies through the Kubernetes API.
 - To enable RBAC, start the apiserver with `--authorization-mode=RBAC`.

[Latest KCNA Dumps](#)

[KCNA PDF Dumps](#)

[KCNA Study Guide](#)