

Practice Exam Questions



KCNA



Kubernetes and Cloud Native Associate



EXAMKILLER

Help Pass Your Exam At First Try

Cloud Native Computing Foundation

Exam KCNA

Kubernetes and Cloud Native Associate (KCNA)

Version: 3.0

[Total Questions: 126]

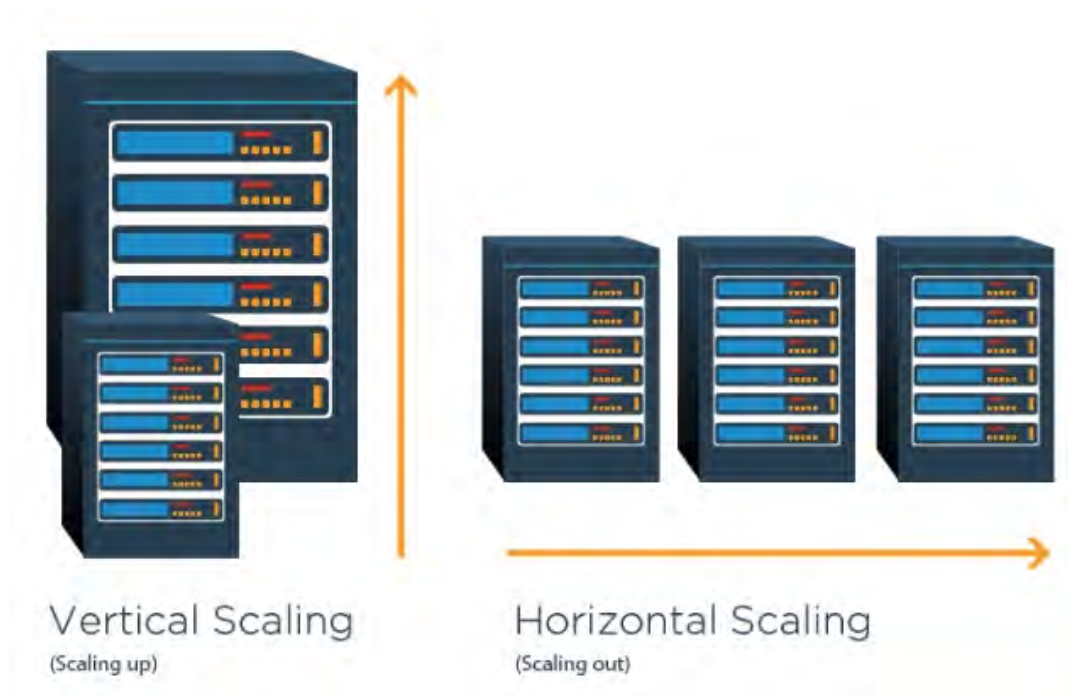
Question No : 1

An application that is nearing its usage limit. To increase the amount of users it can handle, you allo-cate additional memory resources to each instance of the application. What type of scaling is this?

- A. Horizontal Scaling
- B. Cluster Autoscaling
- C. Recursive Scaling
- D. Vertical Scaling

Answer: D

Explanation:



Graphical user interface, diagram

Description automatically generated

Question No : 2

Which of the following best describes a cloud-native app?

- A. An application where all logic is coded into a single large binary.
- B. An application that publishes an HTTPS web front-end.
- C. An application that takes advantages of cloud computing frameworks and their loosely coupled cloud services.

D. An application that leverages services that are native to public cloud platforms such as Azure, GCP, and/or AWS.

Answer: C

Explanation: Cloud-native apps leverage cloud computing frameworks and tend to be microservices based, where individual components of the app are coded as individual.

Question No : 3

Which project in this list is a leading project in the observability space?

- A.** Jaeger
- B.** Vitess
- C.** Argo
- D.** Kubernetes

Answer: A

Explanation: <https://github.com/cncf/landscape#trail-map>



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape (cncf.io) has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer.
cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider.
cncf.io/adopt

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally.
cncf.io/endorse

WHAT IS CLOUD NATIVE?

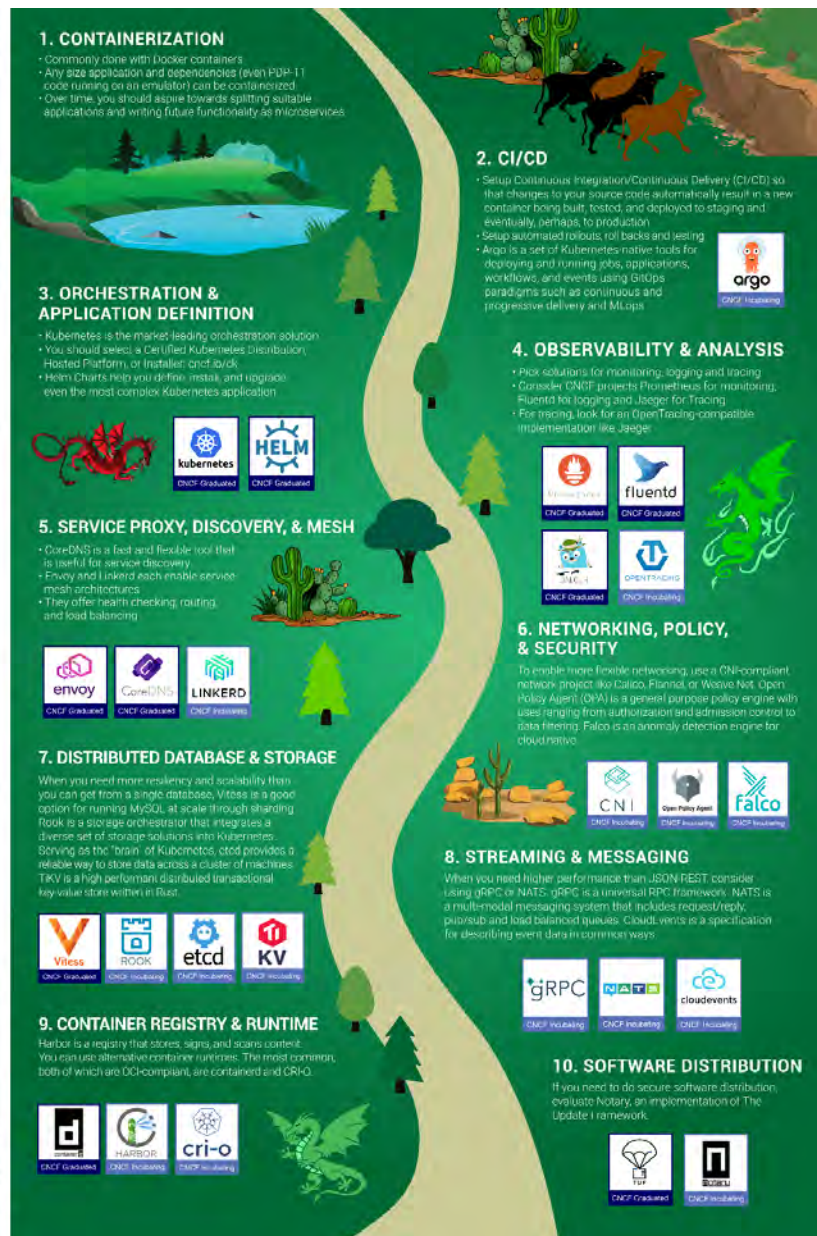
Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501



Question No : 4

To specify a Kubernetes object which language is used?

- A. JSON
- B. Go
- C. YAML
- D. Node
- E. Python

Answer: C

Explanation: <https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/>

Understanding Kubernetes Objects

This page explains how Kubernetes objects are represented in the Kubernetes API, and how you can express them in `.yaml` format.

Graphical user interface, text

Description automatically generated

Question No : 5

What is OPA?

- A. Open Permission Agent
- B. Online Policy Audit
- C. Open Policy Agent
- D. Offline Policy Accessor

Answer: C

Explanation: <https://www.cncf.io/projects/open-policy-agent-opa/>



Graphical user interface, text, application, email

Description automatically generated

Question No : 6

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

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`.

Text, letter

Description automatically generated

Question No : 7

Which authentication method allows JWTs to authenticate?

- A. OpenId connect
- B. Client 'TLS' certificates
- C. OPA gatekeeper
- D. Anonymous

Answer: A

Question No : 8

Which of the following are characteristics of Statefulsets?

- A. Ordered, graceful deployment and scaling
- B. Creates replica sets
- C. Uses headless services

Answer: A

Explanation: <https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/>

Using StatefulSets

StatefulSets are valuable for applications that require one or more of the following.

- Stable, unique network identifiers.
- Stable, persistent storage.
- Ordered, graceful deployment and scaling.
- Ordered, automated rolling updates.

Graphical user interface, text, application, email

Description automatically generated

Question No : 9

Observability and monitoring are not the same?

- A. True
- B. False

Answer: A

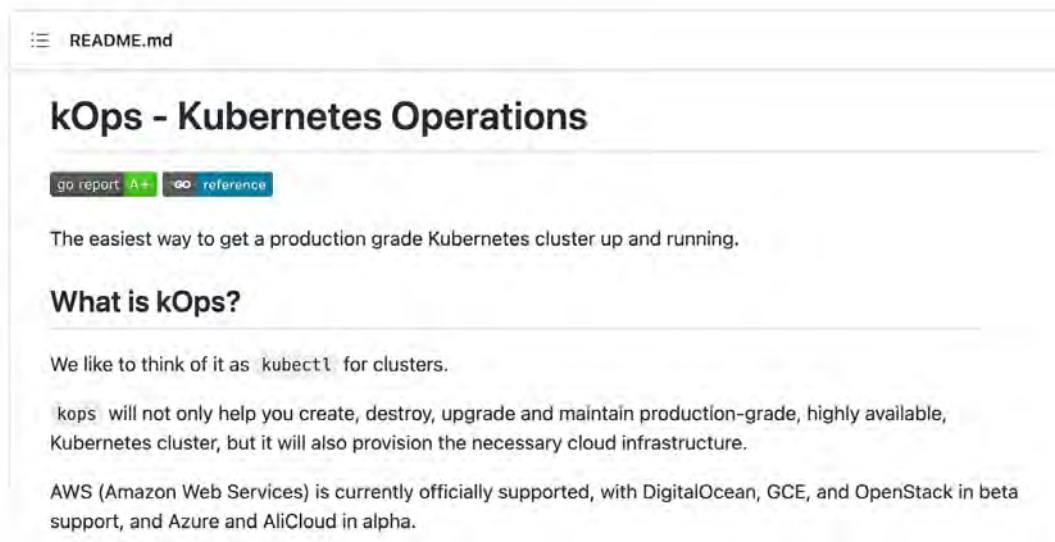
Question No : 10

What does the 'kops' acronym means?

- A. Kubernetes Open Platform Specification
- B. Kubernetes Operations
- C. Kubernetes Operators
- D. Kubernetes Operation Policy Specification

Answer: B

Explanation: <https://github.com/kubernetes/kops>



Graphical user interface, text, application, email

Description automatically generated

Question No : 11

Which prometheus metric type represents a single number value that can increase and decrease over time?

- A. Gauge
- B. Histogram
- C. Summary
- D. Counter

Answer: A

Explanation: https://prometheus.io/docs/concepts/metric_types/#gauge

Gauge

A *gauge* is a metric that represents a single numerical value that can arbitrarily go up and down.

Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests.

Graphical user interface, text

Description automatically generated

Question No : 12

In distributed system tracing, is the term used to refer to a request as it passes through a single component of the distributed system?

- A. Log
- B. Span
- C. Trace
- D. Bucket

Answer: B

Explanation: https://www.splunk.com/en_us/data-insider/what-is-distributed-tracing.html

How does distributed tracing work?

To quickly grasp how distributed tracing works, it's best to look at how it handles a single request. Tracing starts the moment an end user interacts with an application. When the user sends an initial request — an HTTP request, to use a common example — it is assigned a unique trace ID. As the request moves through the host system, every operation performed on it (called a “span” or a “child span”) is tagged with that first request's trace ID, as well as its own unique ID, plus the ID of the operation that originally generated the current request (called the “parent span”).

Each span is a single step on the request's journey and is encoded with important data relating to the microservice process that is performing that operation. These include:

- The service name and address of the process handling the request.
- Logs and events that provide context about the process's activity.
- Tags to query and filter requests by session ID, database host, HTTP method, and other identifiers.
- Detailed stack traces and error messages in the event of a failure.

A distributed tracing tool like Zipkin or Jaeger (both of which we will explore in more detail in a bit) can correlate the data from all the spans and format them into visualizations that are available on request through a web interface.

Now think of a popular online video game with millions of users, the epitome of a modern microservices-driven app. It must track each end user's location, each interaction with other players and the environment, every item the player acquires, end time, and a host of other in-game data. Keeping the game running smoothly would be unthinkable with traditional tracing methods. But distributed request tracing makes it possible.

Text, letter

Description automatically generated

Question No : 13

What is scheduling in Kubernetes

- A. Determining when to execute a cron-job
- B. Assigning pods to nodes
- C. Joining a new nodes to the clusters
- D. Setting a time for automated tasks

Answer: B

Explanation: <https://kubernetes.io/docs/concepts/scheduling-eviction/>

Scheduling

- [Kubernetes Scheduler](#)
- [Assigning Pods to Nodes](#)
- [Pod Overhead](#)
- [Taints and Tolerations](#)
- [Scheduling Framework](#)
- [Scheduler Performance Tuning](#)
- [Resource Bin Packing for Extended Resources](#)

Graphical user interface, application

Description automatically generated

Question No : 14

Have a pod 'hello' and a container in that pod 'green'. Which of the following commands would get the logs for that container?

A.

```
alias k='kubectl'  
k logs -p hello -c green
```

B.

```
alias k='kubectl'  
k logs hello -c green
```

C.

```
alias k='kubectl'  
k get logs -p hello -c green
```

D.

```
alias k='kubectl'  
k logs -p hello green
```

Answer: B

Explanation: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#logs>

logs

Print the logs for a container in a pod or specified resource. If the pod has only one container, the container name is optional.

Usage

```
$ kubectl logs [-f] [-p] [POD | TYPE/NAME] [-c CONTAINER]
```

Flags

Name	Shorthand	Default	Usage
all-containers		false	Get all containers' logs in the pod(s).
container	c		Print the logs of this container
follow	f	false	Specify if the logs should be streamed.

Return snapshot logs from pod nginx with only one container

```
kubectl logs nginx
```

Return snapshot logs from pod nginx with multi containers

```
kubectl logs nginx --all-containers=true
```

Return snapshot logs from all containers in pods defined by label app=nginx

```
kubectl logs -l app=nginx --all-containers=true
```

Return snapshot of previous terminated ruby container logs from pod web-1

```
kubectl logs -p -c ruby web-1
```

Begin streaming the logs of the ruby container in pod web-1

```
kubectl logs -f -c ruby web-1
```

Graphical user interface, text, application, email

Description automatically generated

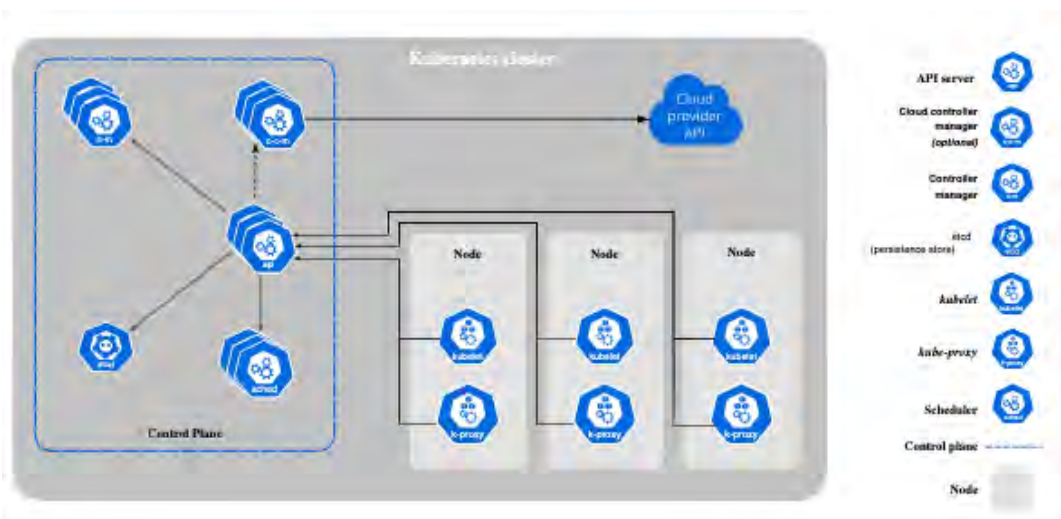
Question No : 15

Which of the following is not the part of Kubernetes Control Plane?

- A. kube scheduler
- B. etcd (pronounce: esty-d)
- C. kube api-server
- D. kube-proxy

Answer: D

Explanation: <https://kubernetes.io/docs/concepts/overview/components/>



Graphical user interface, diagram, application

Description automatically generated

Question No : 16

What Linux feature is used to provide isolation for containers?

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

Answer: D

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

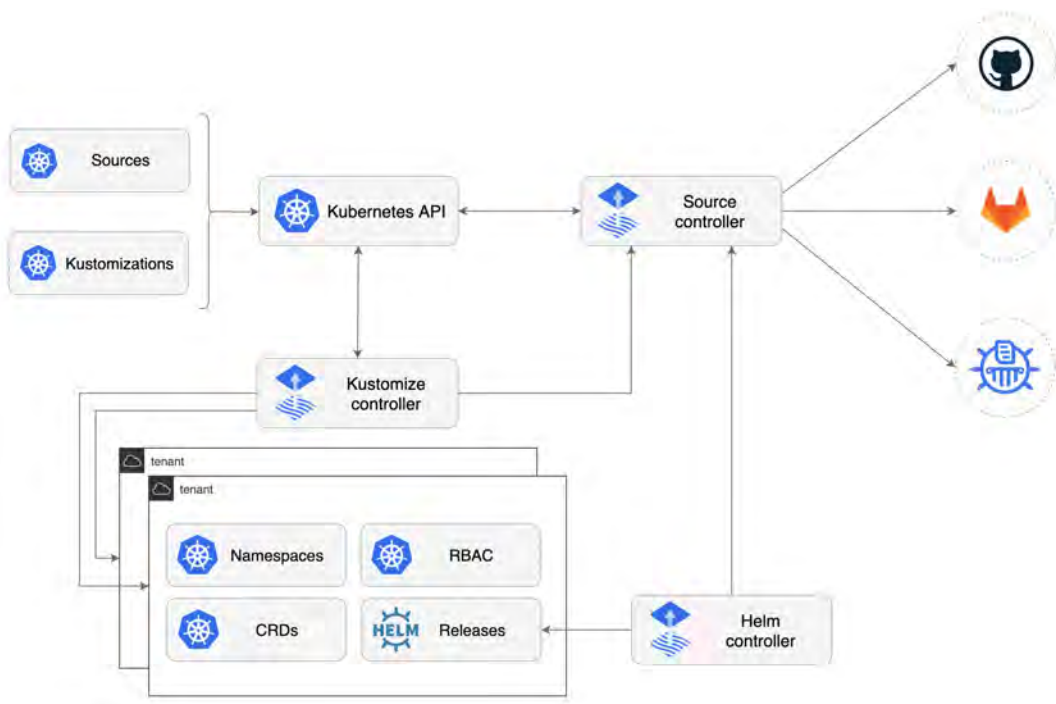
Question No : 17

What do GitOps tools do in kubernetes?

- A. They allow us to make changes to a kubernetes cluster using a Git repository
- B. They manage the source code of kubernetes itself
- C. They allow us to store software code in Git
- D. They allows us to store container images in repositories

Answer: A

Explanation: <https://fluxcd.io/docs/components/>



Diagram

Description automatically generated

Question No : 18

Which control plane component is responsible for scheduling pods?

- A. kube-proxy
- B. kube scheduler
- C. kubelet
- D. kube api-server

Answer: B

Explanation: <https://kubernetes.io/docs/concepts/overview/components/>

kube-scheduler

Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.