



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

Which of the following factors does scheduling take into account when selecting a Node?

- A. How many replicas there are in a Deployment
- B. Services
- C. Resource requirements
- D. The number of existing Pods on a Node

Correct Answer: C

Explanation: Scheduling takes resource requirements into account in the form of resource requests.

QUESTION 2

What is etcd used for in Kubernetes?

- A. Integration with cloud platforms
- B. Network routing for the cluster
- C. Kubernetes API security
- D. Backend object storage for the Kubernetes API

Correct Answer: D

Explanation: etcd serves as a distributed object store that backs the Kubernetes API.

QUESTION 3

What kubectl command is used to edit a resource on the server?

- A. kubectl resource modify
- B. kubectl update resource
- C. kubectl edit
- D. kubectl resource edit

Correct Answer: C

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



edit

Edit a resource from the default editor.

The edit command allows you to directly edit any API resource you can retrieve via the command-line tools. It will open the editor defined by your KUBE_EDITOR, or EDITOR environment variables, or fall back to 'vi' for Linux or 'notepad' for Windows. You can edit multiple objects, although changes are applied one at a time. The command accepts file names as well as command-line arguments, although the files you point to must be previously saved versions of resources.

Editing is done with the API version used to fetch the resource. To edit using a specific API version, fully-qualify the resource, version, and group.

The default format is YAML. To edit in JSON, specify "-o json".

The flag --windows-line-endings can be used to force Windows line endings, otherwise the default for your operating system will be used.

In the event an error occurs while updating, a temporary file will be created on disk that contains your unapplied changes. The most common error when updating a resource is another editor changing the resource on the server. When this occurs, you will have to apply your changes to the newer version of the resource, or update your temporary saved copy to include the latest resource version.

example

Edit the service named 'docker-registry'

```
kubectl edit svc/docker-registry
```

Use an alternative editor

```
KUBE_EDITOR="nano" kubectl edit svc/docker-registry
```

Edit the job 'myjob' in JSON using the v1 API format

```
kubectl edit job.v1.batch/myjob -o json
```

Edit the deployment 'mydeployment' in YAML and save the modified config in its annotation

```
kubectl edit deployment/mydeployment -o yaml --save-annotation
```

QUESTION 4

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

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

Correct Answer: A

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



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape [cncf.io](https://www.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](https://www.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/kscsp](https://www.cncf.io/kscsp)

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally [cncf.io/enduser](https://www.cncf.io/enduser)

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](https://www.cncf.io)

v20200501



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDF-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLOps

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: [cncf.io/ck](https://www.cncf.io/ck)
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant, distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-Rest, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.

10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

QUESTION 5

What is the command used to login to the pod?



- A. kubectl login
- B. kubectl list
- C. kubectl exec
- D. kubectl get

Correct Answer: C

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

List contents of /usr from the first container of pod mypod and sort by modification time # If the command you want to execute in the pod has any flags in common (e.g. -i), # you must use two dashes (--) to separate your command's flags/arguments # Also note, do not surround your command and its flags/arguments with quotes # unless that is how you would execute it normally (i.e., do ls -t /usr, not "ls -t /usr")

```
kubectl exec mypod -i -t -- ls -t /usr
```

QUESTION 6

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.

Correct 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 7

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>

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)
```

The screenshot shows four examples of the `kubectl scale` command:

- 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 8

What is the name for a service that has no clusterIp address?

- A. Headless
- B. NodePort
- C. ClusterIP
- D. LoadBalancer

Correct Answer: A

Explanation: <https://kubernetes.io/docs/concepts/services-networking/service/#headless-services>



Headless Services

Sometimes you don't need load-balancing and a single Service IP. In this case, you can create what are termed "headless" Services, by explicitly specifying "None" for the cluster IP (`.spec.clusterIP`).

You can use a headless Service to interface with other service discovery mechanisms, without being tied to Kubernetes' implementation.

For headless Services , a cluster IP is not allocated, kube-proxy does not handle these Services, and there is no load balancing or proxying done by the platform for them. How DNS is automatically configured depends on whether the Service has selectors defined:

QUESTION 9

What is the main difference between Argo vs. Flux CD?

- A. Argo is pull-based, and Flux is push-based
- B. No difference; both are pull-based
- C. Argo is push-based, and Flux is pull-based
- D. No difference; both are push-based

Correct Answer: C

Explanation: ArgoCD: <https://argo-cd.readthedocs.io/en/stable/developer-guide/ci/#can-i-retrigger-thechecks-without-pushing-a-new-commit> FluxCD: <https://fluxcd.io/>

QUESTION 10

What cloud-native construct does a kubernetes pod wrap?

- A. Container
- B. Virtual Machine (VM)
- C. side car process



D. Docker image

Correct Answer: A

Explanation: Kubernetes is an orchestrator of containerized apps. However, containers must be wrapped in pods before they can be deployed on kubernetes.

QUESTION 11

Fluentd is the leading project in the CNCF space for logging?

A. TRUE

B. FALSE

Correct Answer: A

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



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape ([cncf.io](https://www.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](https://www.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/kcsp](https://www.cncf.io/kcsp)

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally
[cncf.io/enduser](https://www.cncf.io/enduser)

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.

[i.cncf.io](https://www.cncf.io)

v20200501



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: [cncf.io/ck](https://www.cncf.io/ck)
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-Rest, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.

10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

QUESTION 12

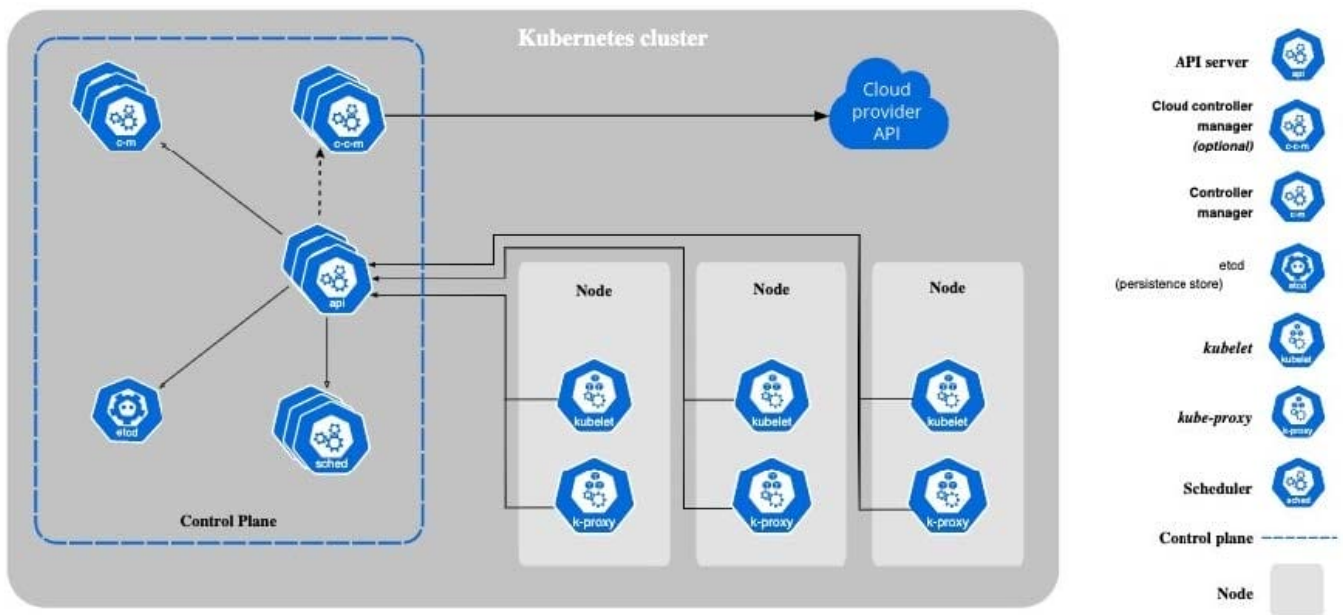
What is the name of the Kubernetes agent that runs on each worker nodes?



- A. kubelet
- B. systemd
- C. kube-proxy
- D. pod

Correct Answer: A

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



QUESTION 13

Open Container Initiative set container standards for

- A. Code, Build, Distribute, Deploy containers
- B. Run, build, and image
- C. Code, Build, Distribute containers
- D. Run, Build, Distribute containers

Correct Answer: D

QUESTION 14

Which of the following container runtime is planned to be deprecated in Kubernetes 1.20 and high-er?

- A. cri-o



B. None of the options

C. docker

D. podman

E. containerd

Correct Answer: C

Explanation: <https://kubernetes.io/blog/2020/12/02/dont-panic-kubernetes-and-docker/>

Wednesday, December 02, 2020

Update: *Kubernetes support for Docker via `dockershim` is now removed. For more information, read the [removal FAQ](#). You can also discuss the deprecation via a dedicated [GitHub issue](#).*

Authors: Jorge Castro, Duffie Cooley, Kat Cosgrove, Justin Garrison, Noah Kantrowitz, Bob Killen, Rey Lejano, Dan "POP" Papandrea, Jeffrey Sica, Davanum "Dims" Srinivas

Kubernetes is **deprecating Docker** as a container runtime after v1.20.

You do not need to panic. It's not as dramatic as it sounds.

TL;DR Docker as an underlying runtime is being deprecated in favor of runtimes that use the [Container Runtime Interface \(CRI\)](#) created for Kubernetes. Docker-produced images will continue to work in your cluster with all runtimes, as they always have.

QUESTION 15

The 4C\\'s of Cloud Native security

A. Chroot, Compute, Cluster and Container

B. Cluster, Cloud, Compute, and Containers

C. Code, Containers, Compute, and Cloud

D. Cloud, Clusters, Containers, and Code

Correct Answer: D



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

[KCNA Practice Test](#)

[KCNA Study Guide](#)

[KCNA Exam Questions](#)