

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.leads4pass.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 command use to get documentation about kubernetes resource type

- A. alias k='kubectl' k api-resources
- B. alias k='kubectl' k api-list
- C. alias k='kubectl' k explain
- D. alias k='kubectl' k get resource

Correct Answer: C

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

explain

List the fields for supported resources.

This command describes the fields associated with each supported API resource. Fields are identified via a simple JSONPath identifier:

```
<type>.<fieldName>[.<fieldName>]
```

Add the `--recursive` flag to display all of the fields at once without descriptions. Information about each field is retrieved from the server in OpenAPI format.

Use "kubectl api-resources" for a complete list of supported resources.

Usage

```
$ kubectl explain RESOURCE
```

Get the documentation of the resource and its fields

```
kubectl explain pods
```

Get the documentation of a specific field of a resource

```
kubectl explain pods.spec.containers
```

QUESTION 2

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` .

QUESTION 3

What command can you use to get documentation about a resource type from the command line?

- A. `kubectl api-resources`
- B. `kubectl explain`
- C. `kubectl get`

D. kubeadm get-resource

Correct Answer: B

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

explain

Get the documentation of the resource and its fields

```
kubectl explain pods
```

Get the documentation of a specific field of a resource

```
kubectl explain pods.spec.containers
```

List the fields for supported resources.

This command describes the fields associated with each supported API resource. Fields are identified via a simple JSONPath identifier:

```
<type>.<fieldName>[.<fieldName>]
```

Add the `--recursive` flag to display all of the fields at once without descriptions. Information about each field is retrieved from the server in OpenAPI format.

Use "kubectl api-resources" for a complete list of supported resources.

Usage

```
$ kubectl explain RESOURCE
```

QUESTION 4

Which of the following best describes the way Kubernetes Role-based access control (RBAC) works?

- A. Kubernetes does not do RBAC
- B. Kubernetes RBAC states which users can perform which actions against which resource
- C. Kubernetes RBAC lists which operations on which resources are denied to users

D. Kubernetes RBAC is responsible for authenticating subjects such as users and groups

Correct Answer: B

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

Using RBAC Authorization

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within your organization.

RBAC authorization uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing you to dynamically configure policies through the Kubernetes API.

To enable RBAC, start the API server with the `--authorization-mode` flag set to a comma-separated list that includes `RBAC`; for example:

```
kube-apiserver --authorization-mode=Example,RBAC --other-options --more-options
```

QUESTION 5

What CNCF project is the leading DNS project in the CNCF landscape?

- A. Kubernetes
- B. gRPC
- C. KubeDNS
- D. CoreDNS

Correct Answer: D

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



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape [Landscape](https://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/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally 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

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/ckd
- 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 performance, 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 CRIO.

10. SOFTWARE DISTRIBUTION

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

QUESTION 6

What is autoscaling?

- 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

Correct Answer: D

<https://kubernetes.io/blog/2016/07/autoscaling-in-kubernetes/> Autoscaling means automatically scaling up or down in response to real-time usage data.

QUESTION 7

What command to view the kube config?

- A. kubectl view config
- B. kubectl config view
- C. kubectl get kubeconfig

Correct Answer: B

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

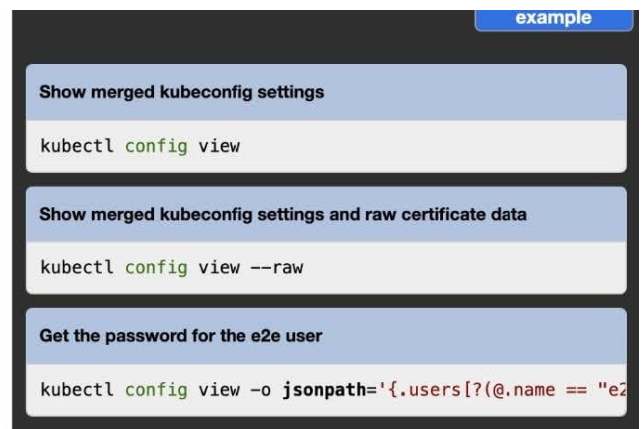
view

Display merged kubeconfig settings or a specified kubeconfig file.

You can use `--output jsonpath={...}` to extract specific values using a jsonpath expression.

Usage

```
$ kubectl config view
```

**QUESTION 8**

You might need to run a stateless application in kubernetes, and you want to be able to scale easily and perform rolling updates. What kubernetes resource type can you use to do this

- A. Dameon set
- B. Replica set
- C. Deployment
- D. pod
- E. service
- F. Stateful set

Correct Answer: C

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

Deployments

A *Deployment* provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

Note: Do not manage ReplicaSets owned by a Deployment. Consider opening an issue in the main Kubernetes repository if your use case is not covered below.

QUESTION 9

kubeadm is an administrative dashboard for kubernetes

- A. False
- B. True

Correct Answer: A

Explanation: <https://kubernetes.io/docs/reference/setup-tools/kubeadm/>

Kubeadm

Kubeadm is a tool built to provide `kubeadm init` and `kubeadm join` as best-practice "fast paths" for creating Kubernetes clusters.

kubeadm performs the actions necessary to get a minimum viable cluster up and running. By design, it cares only about bootstrapping, not about provisioning machines. Likewise, installing various nice-to-have addons, like the Kubernetes Dashboard, monitoring solutions, and cloud-specific addons, is not in scope.

Instead, we expect higher-level and more tailored tooling to be built on top of kubeadm, and ideally, using kubeadm as the basis of all deployments will make it easier to create conformant clusters.



QUESTION 10

What is the functionality of the daemon set?

- A. To run a copy of the pod in all the nodes of the cluster
- B. To initialize the pod before starting the main pod
- C. To run a copy of the pod in a single node of the cluster

Correct Answer: A

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

DaemonSet

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

Some typical uses of a DaemonSet are:

- running a cluster storage daemon on every node
 - running a logs collection daemon on every node
 - running a node monitoring daemon on every node
-

QUESTION 11

The three typical opentelemetry data is?

- A. Metrics
- B. Traces
- C. Logs
- D. All of the options

Correct Answer: D

Explanation: <https://opentelemetry.io/docs/concepts/data-sources/>

What is OpenTelemetry?

OpenTelemetry is a set of APIs, SDKs, tooling and integrations that are designed for the creation and management of *telemetry data* such as traces, metrics, and logs. The project provides a vendor-agnostic implementation that can be configured to send telemetry data to the backend(s) of your choice. It supports a variety of popular open-source projects including Jaeger and Prometheus.

QUESTION 12

Which command-line tool is used to interact with the Kubernetes cluster?

- A. kube-api
- B. kubectl
- C. kube-scheduler

Correct Answer: B

Explanation: <https://kubernetes.io/docs/reference/kubectl/>

Command line tool (kubectl)

Kubernetes provides a command line tool for communicating with a Kubernetes cluster's control plane, using the Kubernetes API.

This tool is named `kubectl`.

For configuration, `kubectl` looks for a file named `config` in the `$HOME/.kube` directory. You can specify other `kubeconfig` files by setting the `KUBECONFIG` environment variable or by setting the `--kubeconfig` flag.

This overview covers `kubectl` syntax, describes the command operations, and provides common examples. For details about each command, including all the supported flags and subcommands, see the [kubectl](#) reference documentation.

For installation instructions, see [Installing kubectl](#); for a quick guide, see the [cheat sheet](#). If you're used to using the `docker` command-line tool, [kubectl for Docker Users](#) explains some equivalent commands for Kubernetes.

QUESTION 13

Which style of operations are preferred for kubernetes and cloud-native applications?

- A. Imperative
- B. None of the above

C. Declarative

Correct Answer: C

Explanation: <https://kubernetes.io/docs/tasks/manage-kubernetes-objects/declarative-config/#trade-offs>

QUESTION 14

What framework allows developers to write code without worrying about the servers and operating systems they will run on?

- A. Virtualization
- B. Docker
- C. Serverless D. Kubernetes

Correct Answer: C

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 Study Guide](#)

[KCNA Exam Questions](#)

[KCNA Braindumps](#)