



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.pass4itsure.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

Notary and the update framework leading security projects in CNCF

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) 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.

[i.cncf.io](https://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
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

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

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

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 2

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 3

Which of the following command is used to get detailed information about the pod?

- A. kubectl info
- B. kubectl get
- C. kubectl describe
- D. kubectl explain

Correct Answer: C

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

```
Describe a pod  
kubectl describe pods/nginx  
  
Describe a pod identified by type and name in "pod.json"  
kubectl describe -f pod.json  
  
Describe all pods  
kubectl describe pods
```

QUESTION 4

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
```

The screenshot shows three terminal examples for the `kubectl config view` command. Each example is presented in a dark-themed terminal window with a blue header bar. The first example, titled "Show merged kubeconfig settings", shows the command `kubectl config view`. The second example, titled "Show merged kubeconfig settings and raw certificate data", shows the command `kubectl config view --raw`. The third example, titled "Get the password for the e2e user", shows the command `kubectl config view -o jsonpath='{.users[?(@.name == "e2e-...)]}'`.

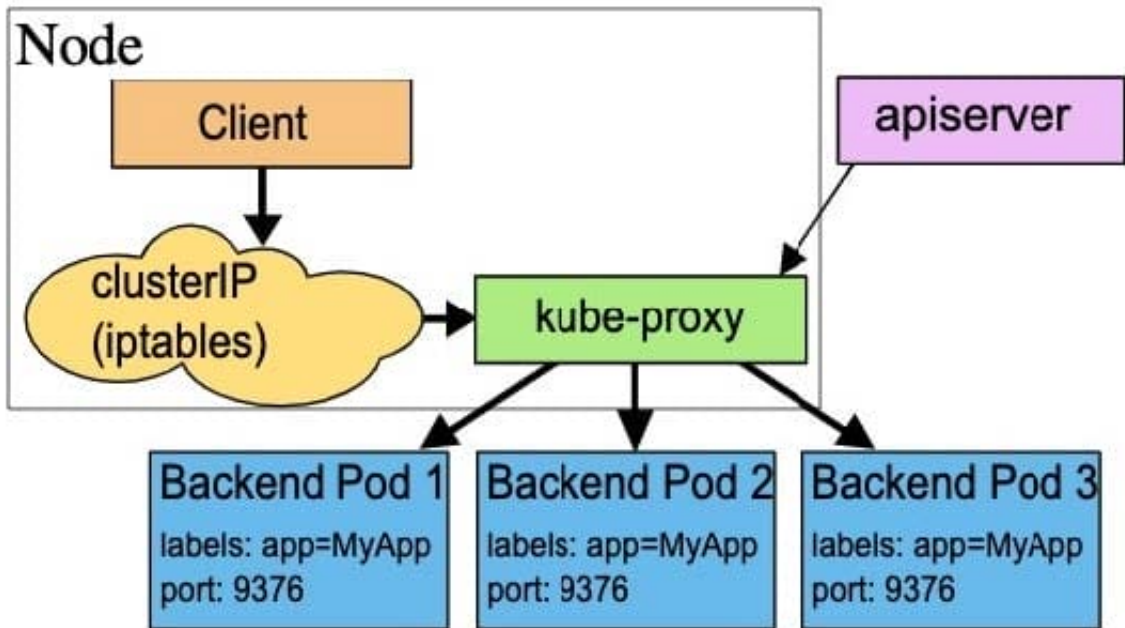
QUESTION 5

How does service logical group set of pods?

- A. Using hostname
- B. Using label and selectors
- C. Using IP address

Correct Answer: B

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



[KCNA VCE Dumps](#)

[KCNA Practice Test](#)

[KCNA Study Guide](#)