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.pass2lead.com/kcna.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

How can you achieve cost optimization in the cloud environment?

A. Use On Demand instances

B. Use Spot Instances

C. Use Reserved Instances

D. Use Bare Metal

Correct Answer: C

QUESTION 2

Which of the following is not a stop on the cloud native trailmap?

A. Microservices

B. CI/CD

C. Containerization

D. Software distribution

Correct Answer: A

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

QUESTION 3

What are container runtimes with Kubernetes?

A. CRI-O

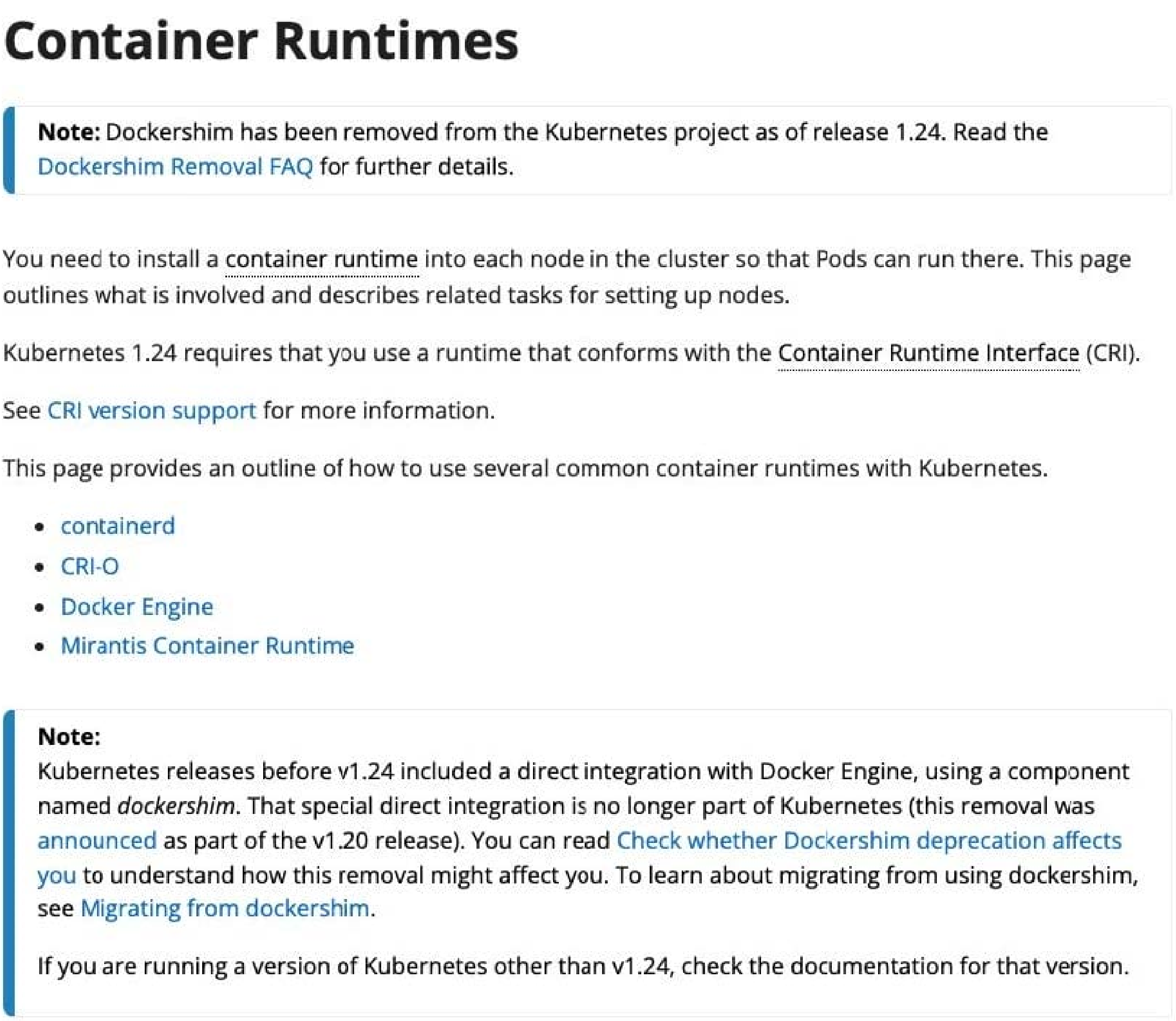
B. lxd

C. containerd

D. Dockershim

Correct Answer: AC

Explanation: https://kubernetes.io/docs/setup/production-environment/container-runtimes/



Container Runtimes

Note: Dockershim has been removed from the Kubernetes project as of release 1.24. Read the Dockershim Removal FAQ for further details.

You need to install a container runtime into each node in the cluster so that Pods can run there. This page outlines what is involved and describes related tasks for setting up nodes.

Kubernetes 1.24 requires that you use a runtime that conforms with the Container Runtime Interface (CRI).

See CRI version support for more information.

This page provides an outline of how to use several common container runtimes with Kubernetes.

- containerd
- CRI-O
- Docker Engine
- Mirantis Container Runtime

Note:
Kubernetes releases before v1.24 included a direct integration with Docker Engine, using a component named *dockershim*. That special direct integration is no longer part of Kubernetes (this removal was announced as part of the v1.20 release). You can read Check whether Dockershim deprecation affects you to understand how this removal might affect you. To learn about migrating from using dockershim, see Migrating from dockershim.

If you are running a version of Kubernetes other than v1.24, check the documentation for that version.

QUESTION 4

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

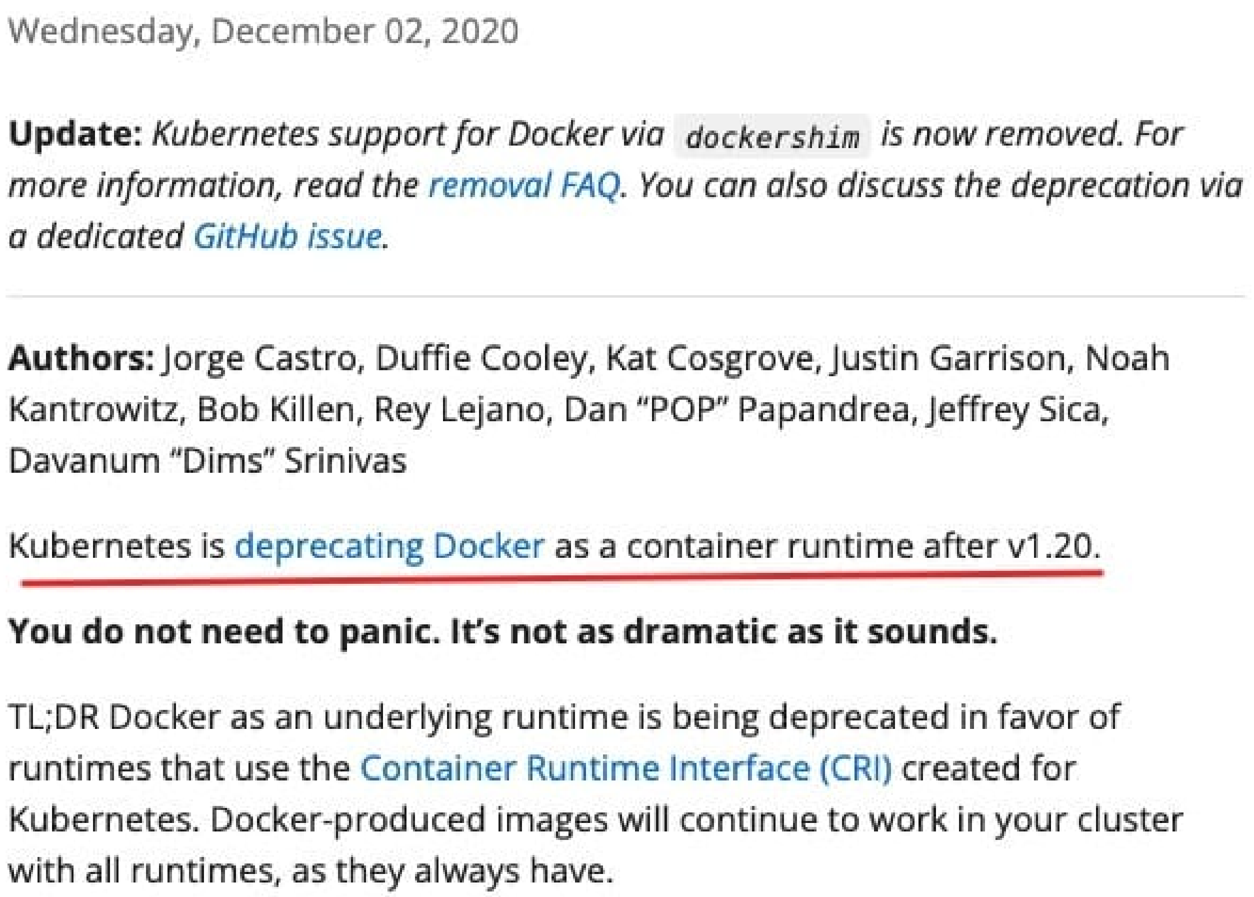
C. docker

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 5

What command use to get documentation about kubernetes resource type

A. alias k=\\'kubectl\\' k api-resources

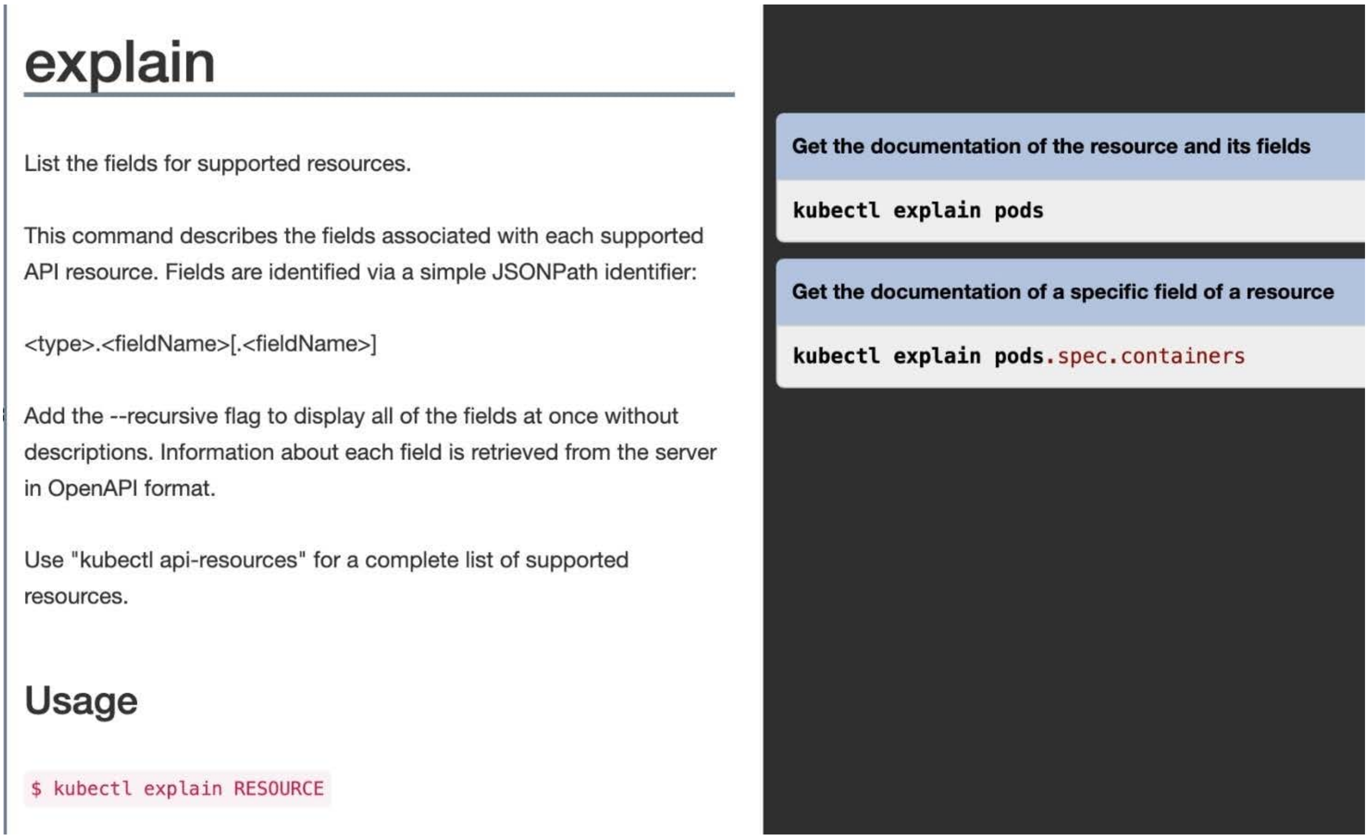
B. alias k=\\'kubectl\\' k api-list

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 6

Various Container Orchestrator Systems (COS)?

A. Apache Mesos

B. None of the options

C. Docker Swarm

D. Kubernetes

Correct Answer: ACD

QUESTION 7

What standard does kubelet use to communicate with the container runtime?

A. Service Mesh Interface (SMI)

B. CRI-O

C. ContainerD

D. Container Runtime Interface (CRI)

Correct Answer: D

Explanation: kubelet can communicate with any runtime that supports the CRI standard.

QUESTION 8

Which of the following computing model doesn\\'t require you to provision infrastructure?

A. None of the above

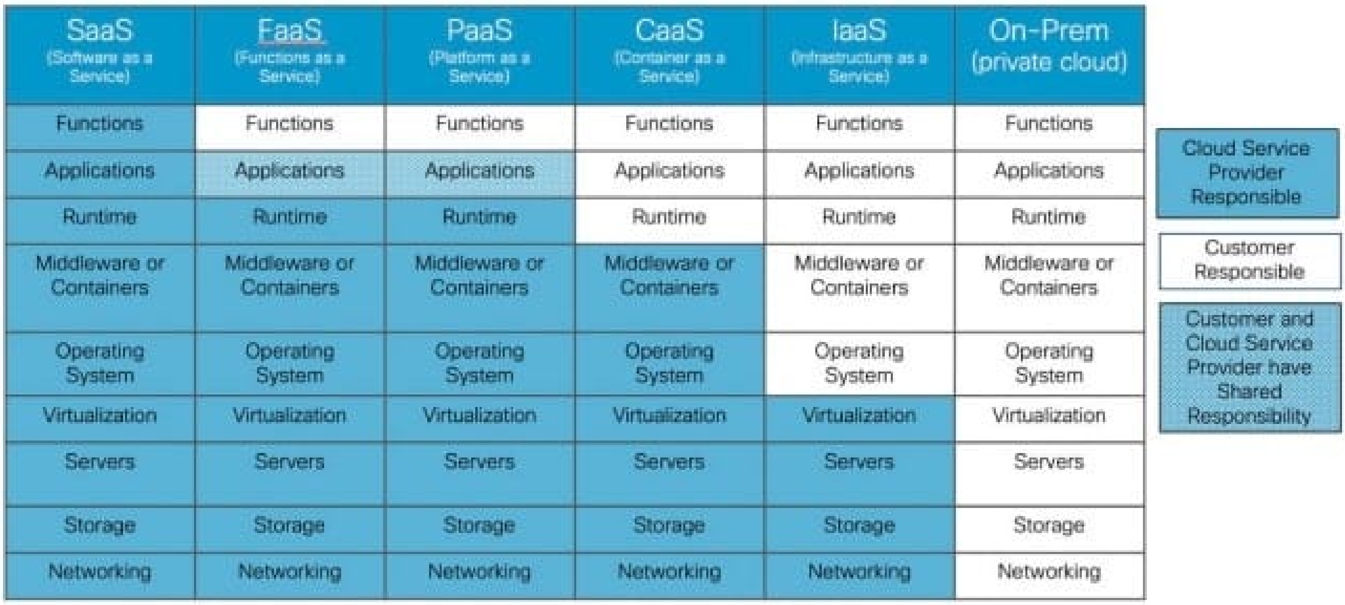
B. Bare Metal

C. Compute Engine

D. Virtual Machines

E. Serverless

Correct Answer: E



SaaS (Software as a Service)	FaaS (Functions as a Service)	PaaS (Platform as a Service)	CaaS (Container as a Service)	IaaS (Infrastructure as a Service)	On-Prem (private cloud)
Functions	Functions	Functions	Functions	Functions	Functions
Applications	Applications	Applications	Applications	Applications	Applications
Runtime	Runtime	Runtime	Runtime	Runtime	Runtime
Middleware or Containers	Middleware or Containers	Middleware or Containers	Middleware or Containers	Middleware or Containers	Middleware or Containers
Operating System	Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking	Networking	Networking

Cloud Service Provider Responsible

Customer Responsible

Customer and Cloud Service Provider have Shared Responsibility

QUESTION 9

What feature is used for selecting the container runtime configuration?

A. RuntimeClass

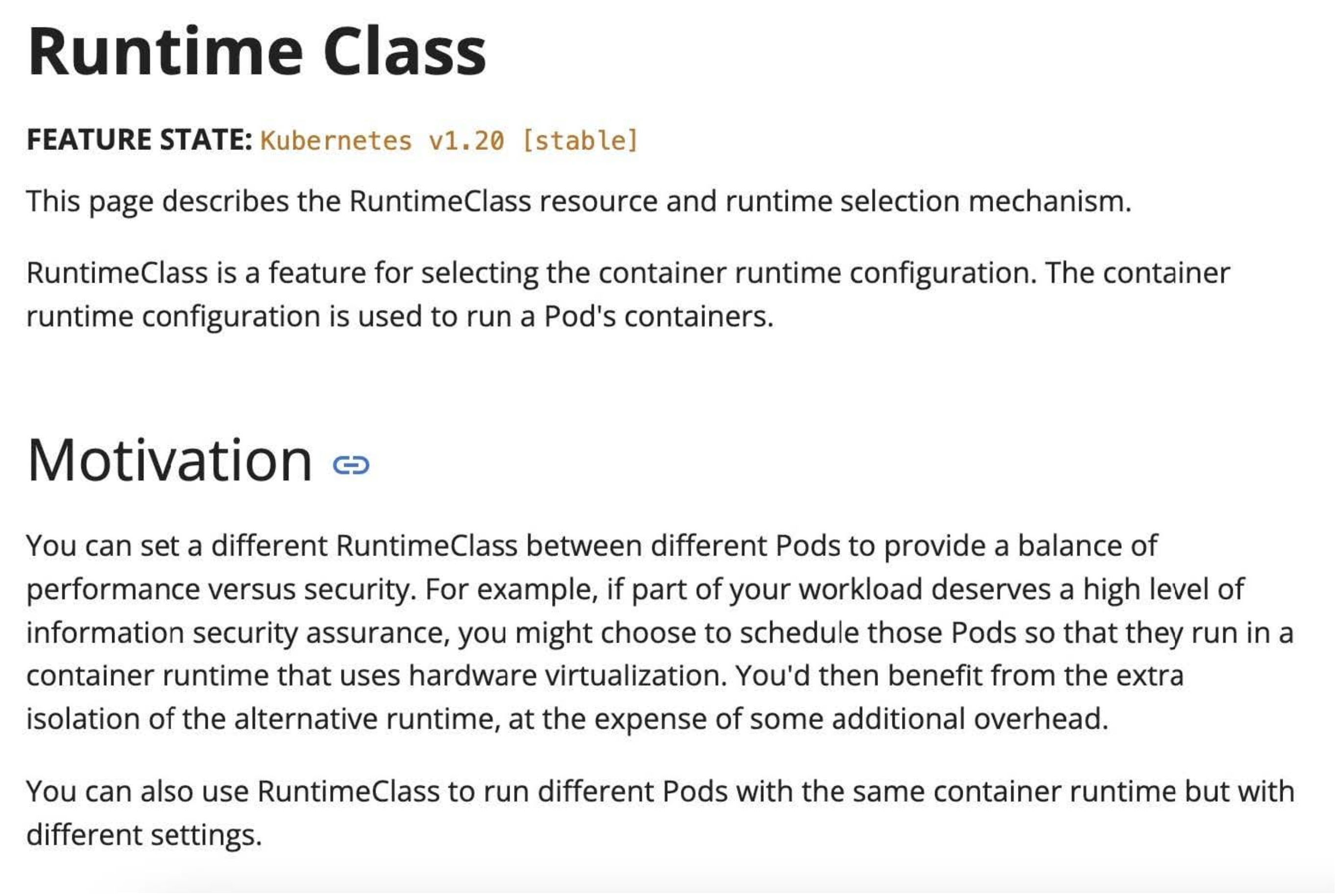
B. RuntimeContainer

C. Runtime

D. RuntimeConfig

Correct Answer: A

Explanation: https://kubernetes.io/docs/concepts/containers/runtime-class/



Runtime Class

FEATURE STATE: `Kubernetes v1.20 [stable]`

This page describes the RuntimeClass resource and runtime selection mechanism.

RuntimeClass is a feature for selecting the container runtime configuration. The container runtime configuration is used to run a Pod's containers.

Motivation

You can set a different RuntimeClass between different Pods to provide a balance of performance versus security. For example, if part of your workload deserves a high level of information security assurance, you might choose to schedule those Pods so that they run in a container runtime that uses hardware virtualization. You'd then benefit from the extra isolation of the alternative runtime, at the expense of some additional overhead.

You can also use RuntimeClass to run different Pods with the same container runtime but with different settings.

QUESTION 10

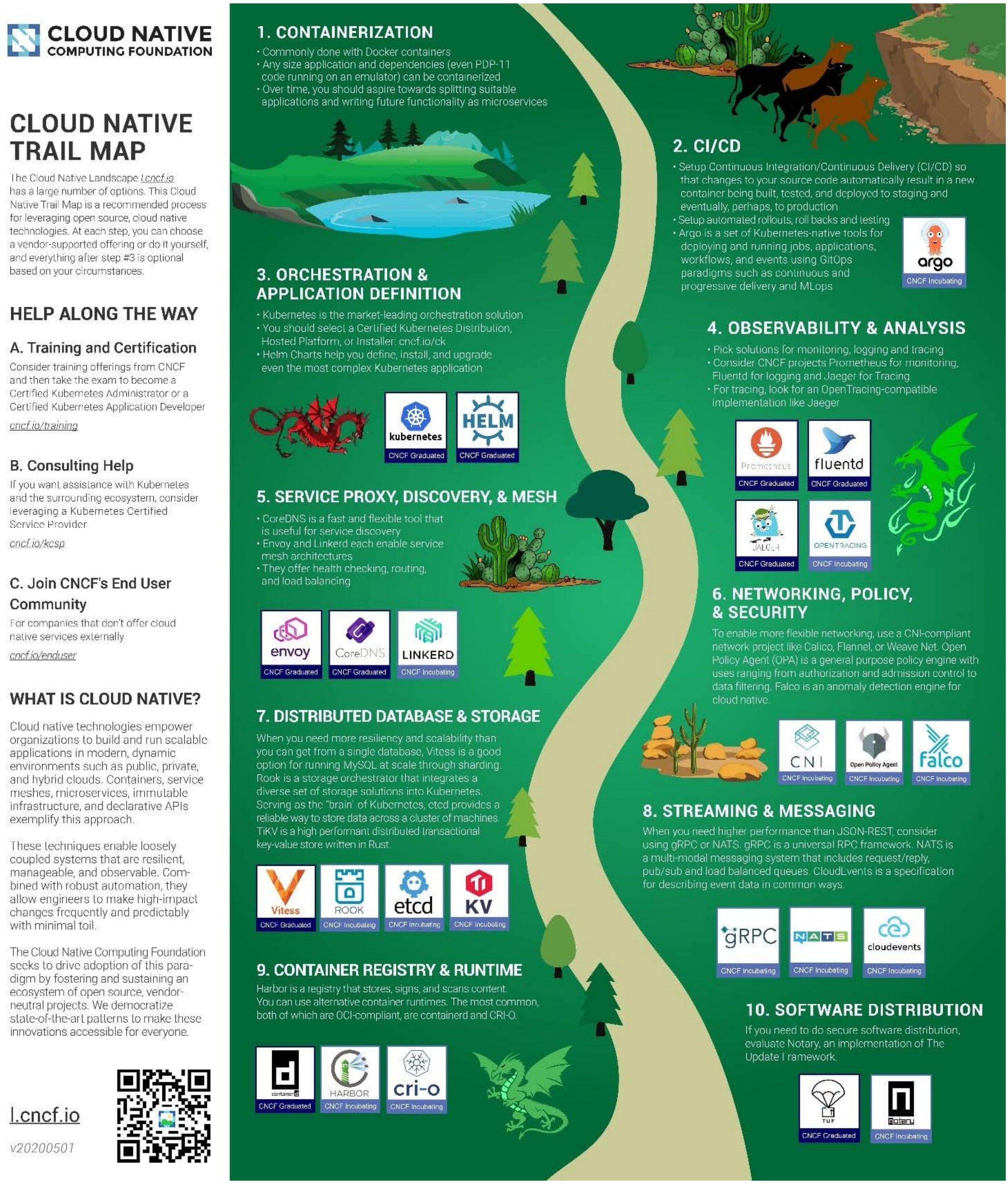
Notary and the update framework leading security projects in CNCF

A. TRUE

B. FALSE

Correct Answer: A

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



CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape *l.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/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. CloudLvents 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 11

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

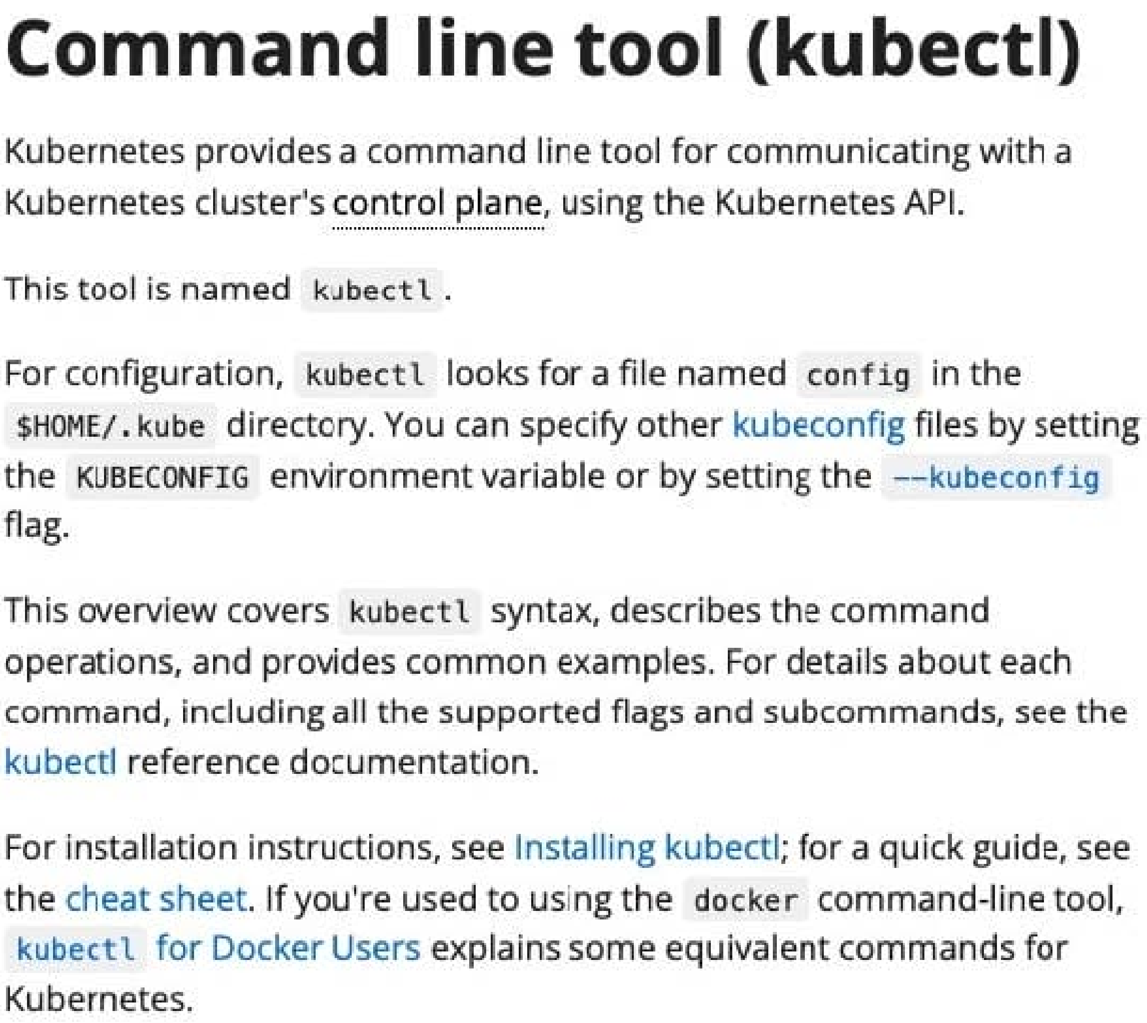
A. kube-api

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 12

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

A. Gauge

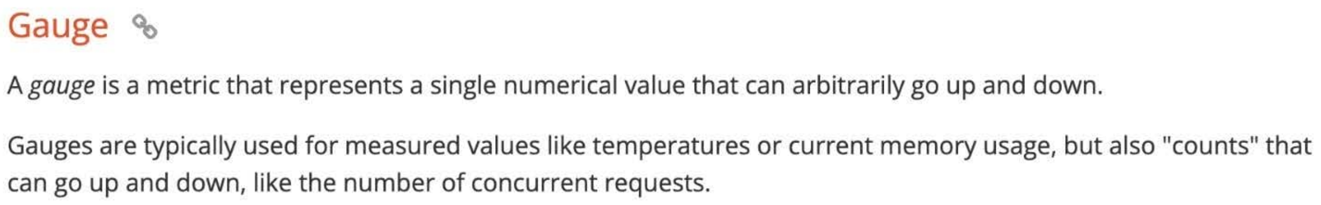
B. Histogram

C. Summary

D. Counter

Correct 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.

QUESTION 13

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

A. Headless

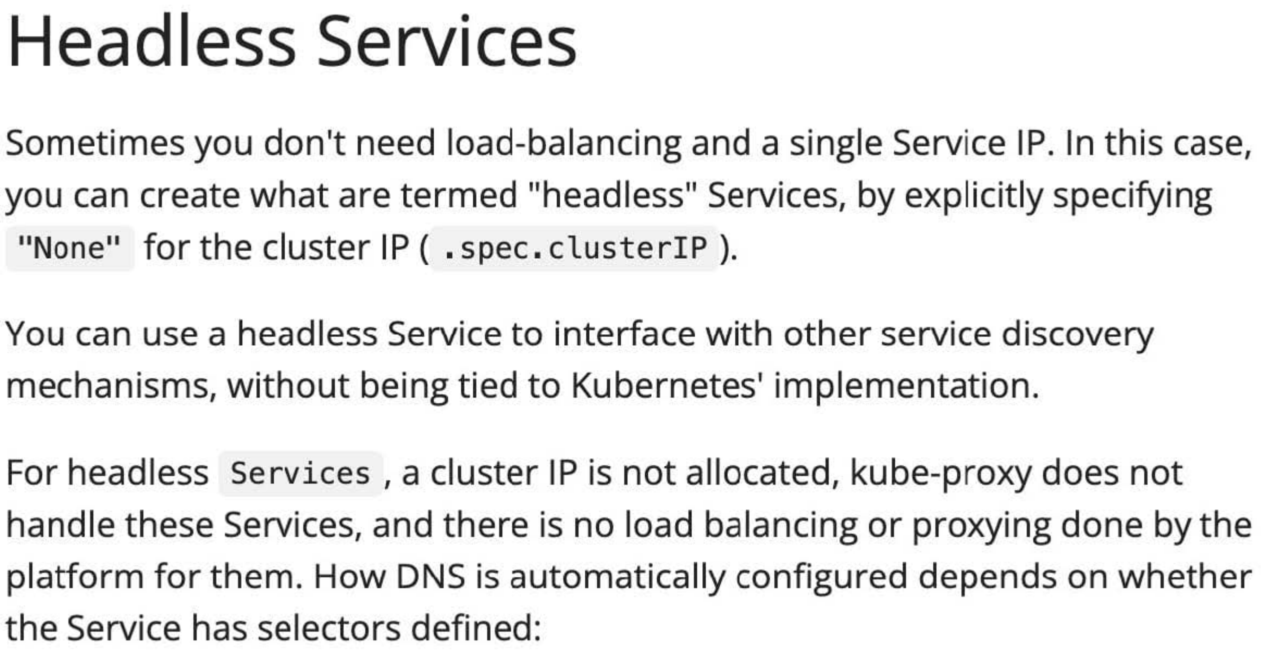
B. NodePort

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 14

What is the use of labels in Kubernetes?

A. All of the options

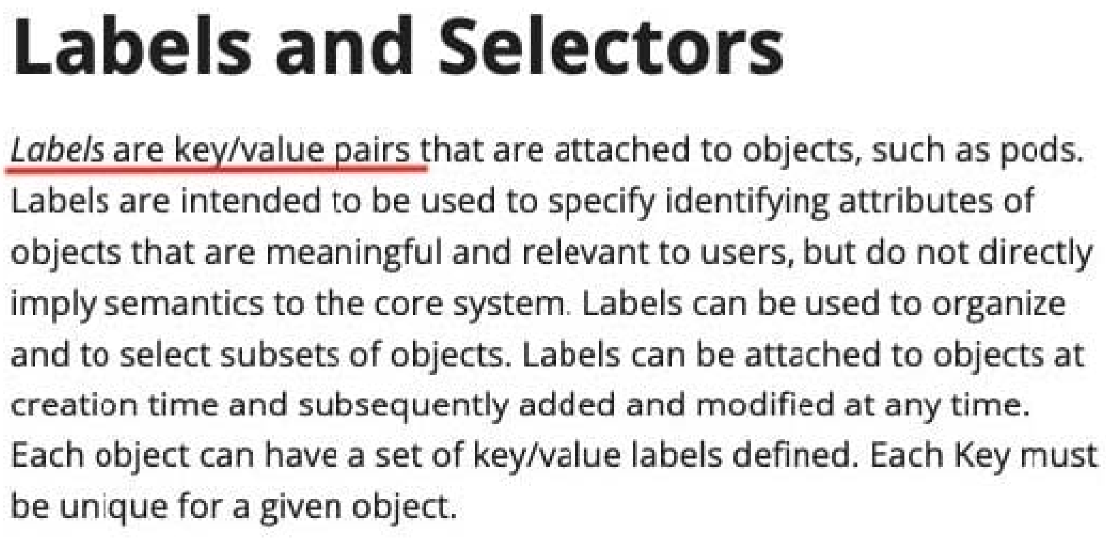
B. It is used to assign annotation to an object

C. It is used to assign key-value pair to an object

D. It is used to assign a name to an object.

Correct Answer: C

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



Labels and Selectors

Labels are key/value pairs that are attached to objects, such as pods. Labels are intended to be used to specify identifying attributes of objects that are meaningful and relevant to users, but do not directly imply semantics to the core system. Labels can be used to organize and to select subsets of objects. Labels can be attached to objects at creation time and subsequently added and modified at any time. Each object can have a set of key/value labels defined. Each Key must be unique for a given object.

QUESTION 15

In Kubernetes, what is considered the primary cluster data source?

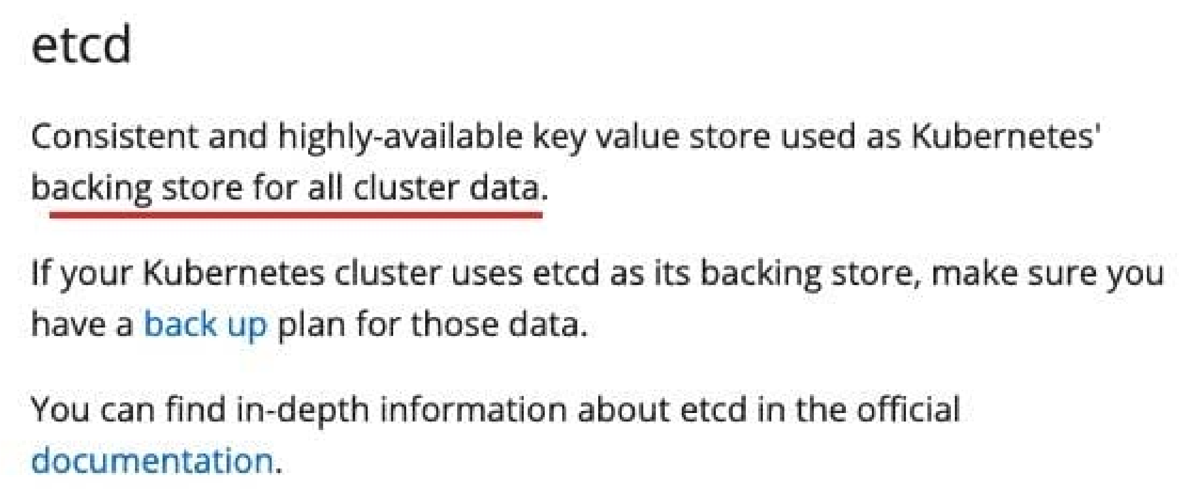
A. etcd (pronounce: esty-d)

B. api server

C. kubelet

D. scheduler

Correct Answer: A



etcd

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

If your Kubernetes cluster uses etcd as its backing store, make sure you have a back up plan for those data.

You can find in-depth information about etcd in the official documentation.