

KCNA Dumps

Kubernetes and Cloud Native Associate (KCNA)

<https://www.certleader.com/KCNA-dumps.html>



NEW QUESTION 1

Which CNCF project is the dominant project with respect to container registries

- A. Envoy
- B. Harbor
- C. Kubernetes
- D. Rook

Answer: B

Explanation:

<https://goharbor.io/>

NEW QUESTION 2

kubeadm is an administrative dashboard for kubernetes

- A. False
- B. True

Answer: A

Explanation:

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

Graphical user interface, text, application Description automatically generated

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.



NEW QUESTION 3

How to create deployment name app-dep, image=nginx, and replicas 5 using imperative command?

- A. `kubectl create app-dep deployment --image=nginx --replicas=5`
- B. `kubectl create deployment app-dep --image=nginx --replicas=5`
- C. `kubectl create app-dep deployment --replicas=5 --image=nginx`

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-deployment-em-> Text Description automatically generated with medium confidence

Create a deployment named my-dep that runs the nginx image with 3 replicas

```
kubectl create deployment my-dep --image=nginx --replicas=3
```

NEW QUESTION 4

What is the most common way to scale the application in the cloud environment?

- A. Parallel Scaling
- B. Horizontal Scaling
- C. Vertical Scaling

Answer: B

Explanation:

<https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>

NEW QUESTION 5

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/> Graphical user interface, text Description automatically generated

Understanding Kubernetes Objects

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

NEW QUESTION 6

What does CNCF stand for?

- A. Cloud Native Computing Foundation
- B. Cloud Native Cloud Foundation
- C. Cloud Native Container Foundation

Answer: A

Explanation:

<https://www.cncf.io/about/who-we-are/>

Graphical user interface, text, application Description automatically generated

The Cloud Native Computing Foundation (CNCF) hosts critical components of the global technology infrastructure. CNCF brings together the world's top developers, end users, and vendors and runs the largest open source developer conferences. CNCF is part of the nonprofit Linux Foundation.

NEW QUESTION 7

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

Answer: C

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

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
```



NEW QUESTION 8

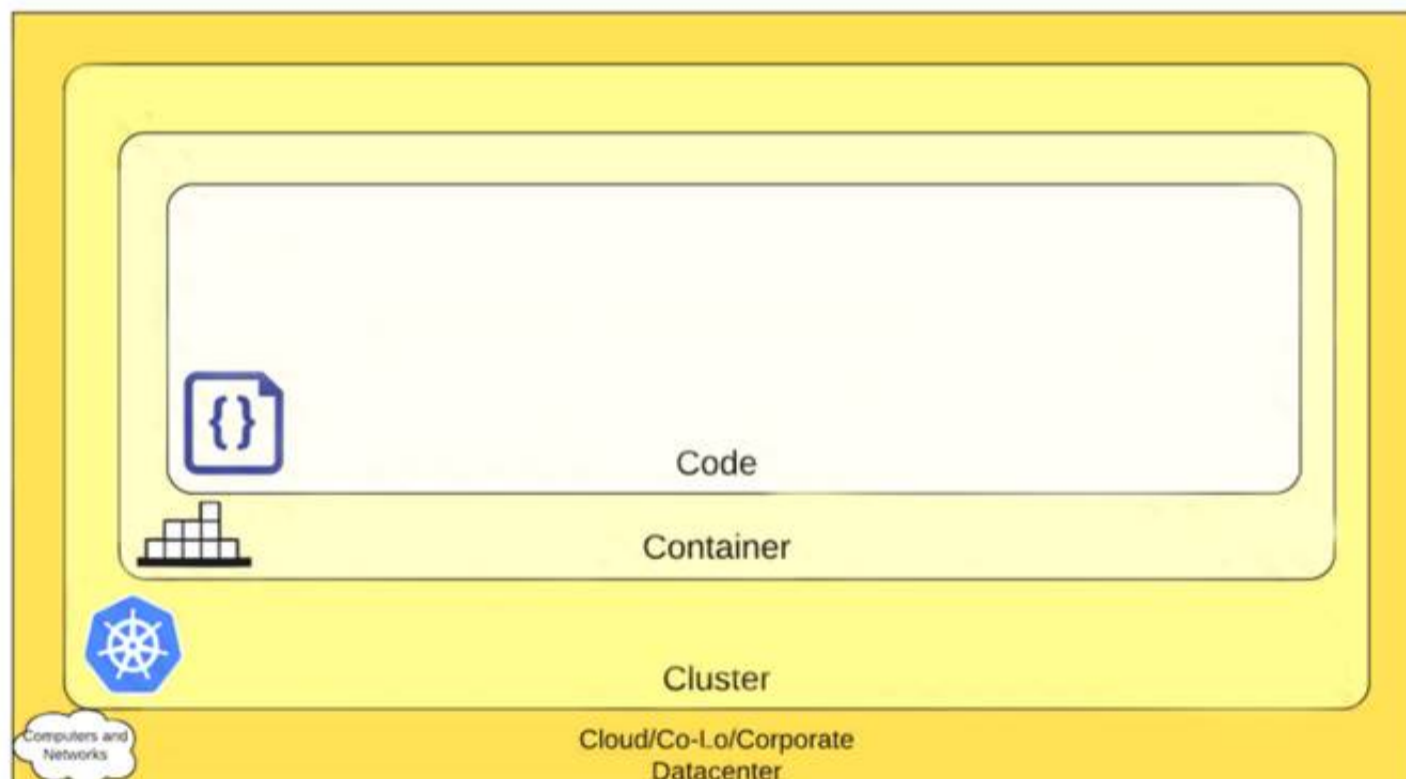
There are three Nodes in a cluster, and want to run exactly one replica of a Pod on each Node. Prefer to automatically create a replica on any new Nodes when they are added. Which Kubernetes re-source should you use?

- A. DaemonSet
- B. ReplicaSet
- C. NodeSet
- D. StatefulSet
- E. Deployment

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>
A DaemonSet runs replicas on all (or just some) Nodes in the cluster.
Table Description automatically generated with medium confidence



NEW QUESTION 9

What tool allows us to build useful visual representations of prometheus data?

- A. Grafana
- B. kubectl
- C. Distributed system tracing
- D. Rook
- E. Kibana

Answer: A

Explanation:

<https://prometheus.io/>
Graphical user interface, text, application Description automatically generated

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

NEW QUESTION 10

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)

Answer: D

Explanation:

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

NEW QUESTION 10

Which of the following is used to request storage in Kubernetes?

- A. PersistentVolume 'PV'
- B. PersistentVolumeClaim 'PVC'
- C. Container Storage Interface 'CSI'
- D. StorageClasses

Answer: B

Explanation:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/>

NEW QUESTION 13

A new Pod is created. Then, the Pod is assigned to a Node. Which Kubernetes component was re-sponsible for determining which Node to assign the Pod to?

- A. kubelet
- B. Scheduler
- C. API Server
- D. Controller manager

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-scheduler/> Graphical user interface, text, application Description automatically generated

The Kubernetes scheduler is a control plane process which assigns Pods to Nodes. The scheduler determines which Nodes are valid placements for each Pod in the scheduling queue according to constraints and available resources. The scheduler then ranks each valid Node and binds the Pod to a suitable Node. Multiple different schedulers may be used within a cluster; kube-scheduler is the reference implementation. See [scheduling](#) for more information about scheduling and the kube-scheduler component.

```
kube-scheduler [flags]
```

NEW QUESTION 16

Which project is not a dominant CNCF project in the storage landscape?

- A. Envoy
- B. Vitess
- C. Rook
- D. TiKV

Answer: A

Explanation:

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

A picture containing timeline Description automatically generated

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape (CNLF) 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/csp

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 erasable 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.

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, and/or kubectl
• 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-RPC, 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.

NEW QUESTION 21

What is the command used to scale the application?

- A. kubectl run
- B. kubectl explain
- C. kubectl scale

Answer: C

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#scale> Graphical user interface, text, application, email Description automatically generated

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)
```

```
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/baz
```

NEW QUESTION 23

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

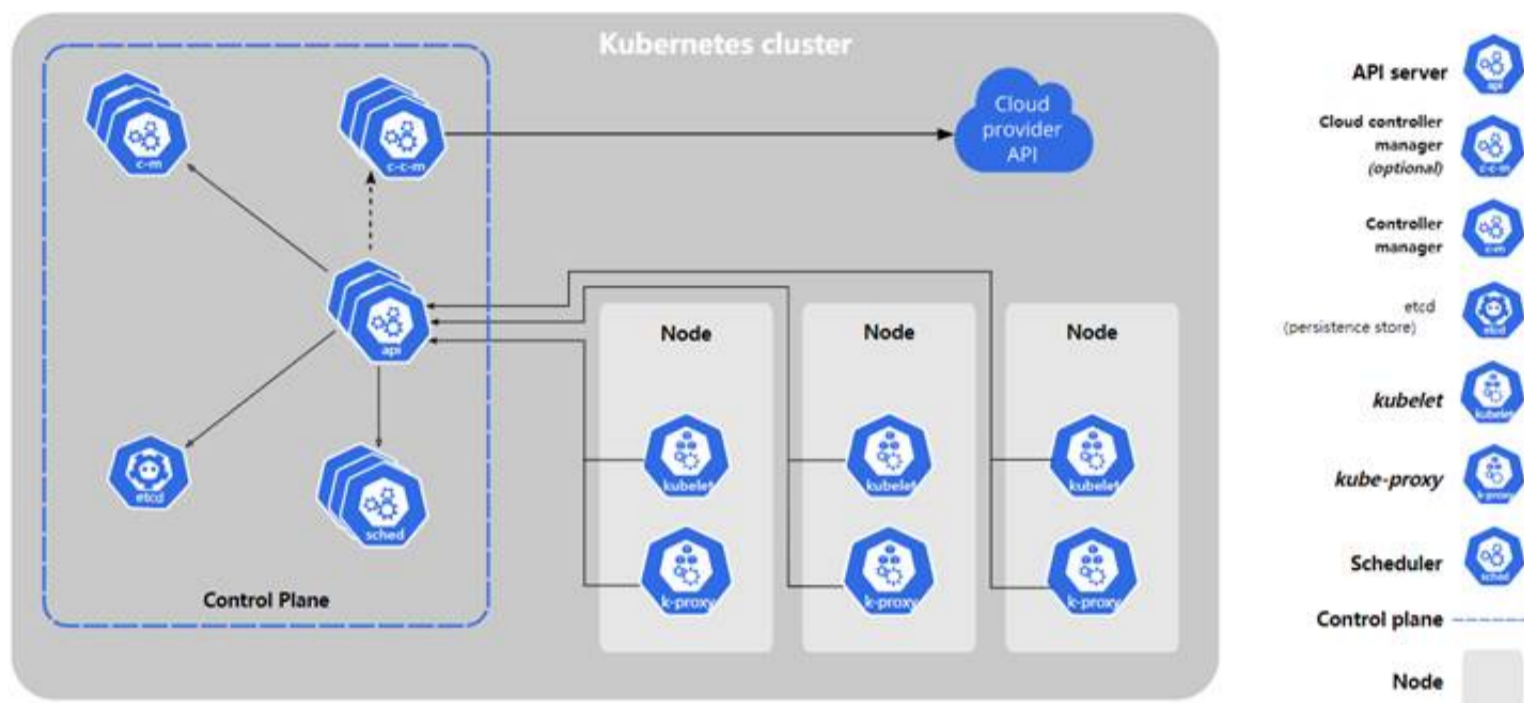
- A. kube scheduler
- B. etcd (pronounce: esty-d)
- C. kube api-server
- D. kube-proxy

Answer: D

Explanation:

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

Graphical user interface, diagram, application Description automatically generated



NEW QUESTION 24

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

- A. Imperative
- B. None of the above
- C. Declarative

Answer: C

Explanation:

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

NEW QUESTION 27

What is the command to list all the available objects in your Kubernetes cluster?

- A. kubectl get all
- B. kubectl get api-resources
- C. kubectl api-resources
- D. kubectl get pods

Answer: C

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

Resource types

List all supported resource types along with their shortnames, API group, whether they are namespaced, and Kind:

```
kubectl api-resources
```

NEW QUESTION 28

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

Answer: C

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#describe> Graphical user interface, application Description automatically generated



NEW QUESTION 31

Which of the following is not the Kubernetes AutoScaling Strategy?

- A. Horizontal Pod Autoscaler
- B. Cluster Autoscaler
- C. Vertical Pod Autoscaler
- D. Load Balancing AutoScaler

Answer: D

Explanation:

<https://learnk8s.io/kubernetes-autoscaling-strategies>

Graphical user interface, text Description automatically generated with medium confidence

In Kubernetes, several things are referred to as "autoscaling", including:

- [Horizontal Pod Autoscaler.](#)
- [Vertical Pod Autoscaler.](#)
- [Cluster Autoscaler.](#)

NEW QUESTION 35

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

Answer: C

NEW QUESTION 37

How would you return all the pod data in the json format using kubectl command?

- A. kubectl get pods -o json
- B. kubectl get pods --all-namespaces
- C. kubectl get pods -o wide
- D. kubectl get pods -o jsonpath

Answer: A

Explanation:

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

NEW QUESTION 42

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.

NEW QUESTION 43

How should folks new to the cloud native ecosystem, go about learning the different aspects of the ecosystem?

- A. by signing up the CNCF slack
- B. by reading the Kubernetes documentation
- C. by looking at the cloud native landscape
- D. by looking at the cloud native trail-map

Answer: D

Explanation:

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

NEW QUESTION 48

Flux is built using which toolkit?

- A. CI/CD
- B. DevSecOps
- C. GitOps
- D. DevOps

Answer: C

Explanation:

<https://fluxcd.io/>

Graphical user interface, text, application Description automatically generated

**Flux provides
GitOps for both
apps and
infrastructure**

Flux and Flagger deploy apps with canaries, feature flags, and A/B rollouts. Flux can also manage any Kubernetes resource. Infrastructure and workload dependency management is built in.

**Just push to Git
and Flux does the
rest**

Flux enables application deployment (CD) and (with the help of Flagger) progressive delivery (PD) through automatic reconciliation. Flux can even push back to Git for you with automated container image updates to Git (image scanning and patching).

NEW QUESTION 52

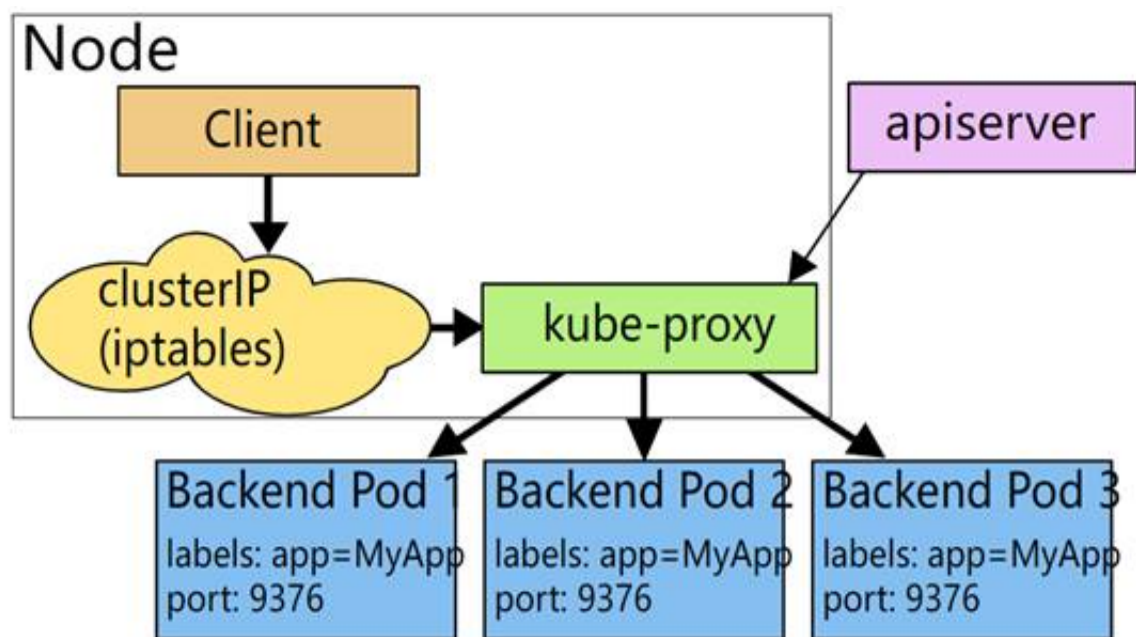
How does service logical group set of pods?

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

Answer: B

Explanation:

<https://kubernetes.io/docs/concepts/services-networking/service/> Diagram Description automatically generated



NEW QUESTION 54

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.

Answer: C

Explanation:

<https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/> Text Description automatically generated

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.

NEW QUESTION 58

What are cluster-wide objects

- A. Service and Pods
- B. Volumes and Nodes
- C. ConfigMaps and Secrets

Answer: B

Explanation:

https://kubernetes.io/docs/concepts/overview/working-with-objects/_print/ Text, letter Description automatically generated

4 - Namespaces

In Kubernetes, *namespaces* provides a mechanism for isolating groups of resources within a single cluster. Names of resources need to be unique within a namespace, but not across namespaces. Namespace-based scoping is applicable only for namespaced objects (e.g. *Deployments, Services, etc*) and not for cluster-wide objects (e.g. *StorageClass, Nodes, PersistentVolumes, etc*).

NEW QUESTION 63

What feature is used for selecting the container runtime configuration?

- A. RuntimeClass
- B. RuntimeContainer
- C. Runtime

D. RuntimeConfig

Answer: A

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

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.

NEW QUESTION 64

Which role is responsible of creating service level indicator 'SLI', service level objective 'SLO', & Service Level Agreements 'SLA'?

- A. Site reliability engineer 'SRE'
- B. DevOps
- C. GitOps
- D. Security and compliance engineer
- E. Developer

Answer: A

Explanation:

<https://www.atlassian.com/incident-management/kpis/sla-vs-slo-vs-sli> Text Description automatically generated

How does this impact SREs?

For those of you following Google's model and using [Site Reliability Engineering \(SRE\) teams](#) to bridge the gap between development and operations, SLAs, SLOs, and SLIs are foundational to success. SLAs help teams set boundaries and error budgets. SLOs help prioritize work. And SLIs tell SREs when they need to freeze all launches to save an endangered error budget—and when they can loosen up the reins.

NEW QUESTION 67

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

kOps - Kubernetes Operations

go report A+ reference

The easiest way to get a production grade Kubernetes cluster up and running.

What is kOps?

We like to think of it as `kubectl` for clusters.

`kops` will not only help you create, destroy, upgrade and maintain production-grade, highly available, Kubernetes cluster, but it will also provision the necessary cloud infrastructure.

AWS (Amazon Web Services) and GCE (Google Cloud Platform) are currently officially supported, with DigitalOcean, Hetzner and OpenStack in beta support, and Azure in alpha.

NEW QUESTION 71

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

Answer: C

Explanation:

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

NEW QUESTION 74

What command to view the kube config?

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

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-view-em-> Graphical user interface, text, application Description automatically generated

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
```



NEW QUESTION 79

The Kubernetes API provides an interface for storing objects. Which of the following describes the type of objects stored by the Kubernetes API?

- A. Containers
- B. REST
- C. YAML
- D. ETCD

Answer: B

Explanation:

Kubernetes objects are RESTful objects.

NEW QUESTION 84

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

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

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/services-networking/service/#headless-services>
Text, letter Description automatically generated

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:

NEW QUESTION 87

Which of the following provides cloud-native storage orchestration?

- A. Cloud Provider Specific storage (EBS, EFS, Cloud Storage)
- B. Cloud Storage
- C. Storage IO

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes> Table Description automatically generated with medium confidence

Types of Persistent Volumes

PersistentVolume types are implemented as plugins. Kubernetes currently supports the following plugins:

- [awsElasticBlockStore](#) - AWS Elastic Block Store (EBS)
- [azureDisk](#) - Azure Disk
- [azureFile](#) - Azure File
- [cephfs](#) - CephFS volume
- [csi](#) - Container Storage Interface (CSI)
- [fc](#) - Fibre Channel (FC) storage
- [gcePersistentDisk](#) - GCE Persistent Disk
- [glusterfs](#) - Glusterfs volume
- [hostPath](#) - HostPath volume (for single node testing only; WILL NOT WORK in a multi-node cluster; consider using `local` volume instead)
- [iscsi](#) - iSCSI (SCSI over IP) storage
- [local](#) - local storage devices mounted on nodes.
- [nfs](#) - Network File System (NFS) storage
- [portworxVolume](#) - Portworx volume
- [rbd](#) - Rados Block Device (RBD) volume
- [vsphereVolume](#) - vSphere VMDK volume

The following types of PersistentVolume are deprecated. This means that support is still available but will be removed in a future Kubernetes release.

- [cinder](#) - Cinder (OpenStack block storage) (**deprecated** in v1.18)

NEW QUESTION 92

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

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

Answer: D

Explanation:

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

A picture containing timeline Description automatically generated



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/kspp

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.

cncf.io

v20200501



NEW QUESTION 96

Notary and the update framework leading security projects in CNCF

- A. TRUE
- B. FALSE

Answer: A

Explanation:

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

A picture containing timeline Description automatically generated



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/kspp

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.

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/quick
- 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 CNF-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. TKV 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.

NEW QUESTION 98

What is a commonly used package manager for kubernetes applications?

- A. npm
- B. apt
- C. helm
- D. kubernetes manifest

Answer: C

Explanation:

<https://helm.sh/>

NEW QUESTION 102

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 Text, letter Description automatically generated

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.

NEW QUESTION 104

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

Answer: C

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

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.

NEW QUESTION 107

Which kubernetes object do deployments use behind the scenes when they need to scale pods?

- A. Horizontal pod autoscaler
- B. ReplicaSets
- C. kubectl
- D. Replication controller

Answer: B

Explanation:

<https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/>
Graphical user interface, text, application, email Description automatically generated

ReplicaSet

A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

NEW QUESTION 111

Various Container Orchestrator Systems (COS)?

- A. Apache Mesos
- B. None of the options
- C. Docker Swarm
- D. Kubernetes

Answer: ACD

NEW QUESTION 113

Which statement is true about Pod Networking?

- A. All pod requires an external DNS server to get the hostname
- B. All containers in a pod get a unique IP address
- C. All containers in a pod share a single IP address
- D. All pod requires NAT to get a unique IP address.

Answer: C

Explanation:

<https://kubernetes.io/docs/concepts/workloads/pods/#pod-networking> Text Description automatically generated

Pod networking

Each Pod is assigned a unique IP address for each address family. Every container in a Pod shares the network namespace, including the IP address and network ports. Inside a Pod (and **only** then), the containers that belong to the Pod can communicate with one another using `localhost`. When containers in a Pod communicate with entities *outside the Pod*, they must coordinate how they use the shared network resources (such as ports). Within a Pod, containers share an IP address and port space, and can find each other via `localhost`. The containers in a Pod can also communicate with each other using standard inter-process communications like SystemV semaphores or POSIX shared memory. Containers in different Pods have distinct IP addresses and can not communicate by OS-level IPC without special configuration. Containers that want to interact with a container running in a different Pod can use IP networking to communicate.

Containers within the Pod see the system hostname as being the same as the configured `name` for the Pod. There's more about this in the [networking](#) section.

NEW QUESTION 116

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/>
Graphical user interface, text, application, email Description automatically generated

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.

NEW QUESTION 120

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

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>
Graphical user interface, text, application Description automatically generated with medium confidence

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

NEW QUESTION 121

How to get the logs of the previously terminated nginx container from the web pod?

- A. `kubectl logs -p -c nginx web`
- B. `kubectl logs nginx`
- C. `kubectl logs -p -c web nginx`
- D. `kubectl logs -f -c nginx web`

Answer: A

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#logs> Text Description automatically generated with medium confidence

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

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

NEW QUESTION 125

Which kubernetes object do deployments use behind the scenes when they need to scale pods?

- A. POD
- B. Deployment
- C. Horizontal pod autoscaler
- D. Api Scheduler
- E. Replicasets

Answer: E

Explanation:

<https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/> Graphical user interface, text, application, email Description automatically generated

ReplicaSet

A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

NEW QUESTION 128

'`kubectl delete -n my-ns po,svc --all`' will delete pods and services including uninitialized ones in the namespace 'my-ns'

- A. FALSE
- B. TRUE

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#delete> Graphical user interface, text, application, email Description automatically generated

IMPORTANT: Force deleting pods does not wait for confirmation that the pod's processes have been terminated, which can leave those processes running until the node detects the deletion and completes graceful deletion. If your processes use shared storage or talk to a remote API and depend on the name of the pod to identify themselves, force deleting those pods may result in multiple processes running on different machines using the same identification which may lead to data corruption or inconsistency. Only force delete pods when you are sure the pod is terminated, or if your application can tolerate multiple copies of the same pod running at once. Also, if you force delete pods, the scheduler may place new pods on those nodes before the node has released those resources and causing those pods to be evicted immediately.

Note that the delete command does NOT do resource version checks, so if someone submits an update to a resource right when you submit a delete, their update will be lost along with the rest of the resource.

After a CustomResourceDefinition is deleted, invalidation of discovery cache may take up to 10 minutes. If you don't want to wait, you might want to run "kubectl api-resources" to refresh the discovery cache.

Usage

```
$ kubectl delete ([-f FILENAME] | [-k DIRECTORY] | TYPE [(NAME | -l label | --all)])
```

Delete a pod based on the type and name in the JSON passed into stdin
<code>cat pod.json kubectl delete -f -</code>
Delete pods and services with same names "baz" and "foo"
<code>kubectl delete pod,service baz foo</code>
Delete pods and services with label name=myLabel
<code>kubectl delete pods,services -l name=myLabel</code>
Delete a pod with minimal delay
<code>kubectl delete pod foo --now</code>
Force delete a pod on a dead node
<code>kubectl delete pod foo --force</code>
Delete all pods
<code>kubectl delete pods --all</code>

NEW QUESTION 131

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

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

Answer: B

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

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.

NEW QUESTION 136

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your KCNA Exam with Our Prep Materials Via below:

<https://www.certleader.com/KCNA-dumps.html>