

# Linux-Foundation

## Exam Questions KCNA

Kubernetes and Cloud Native Associate (KCNA)



#### 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

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 3

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 4

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

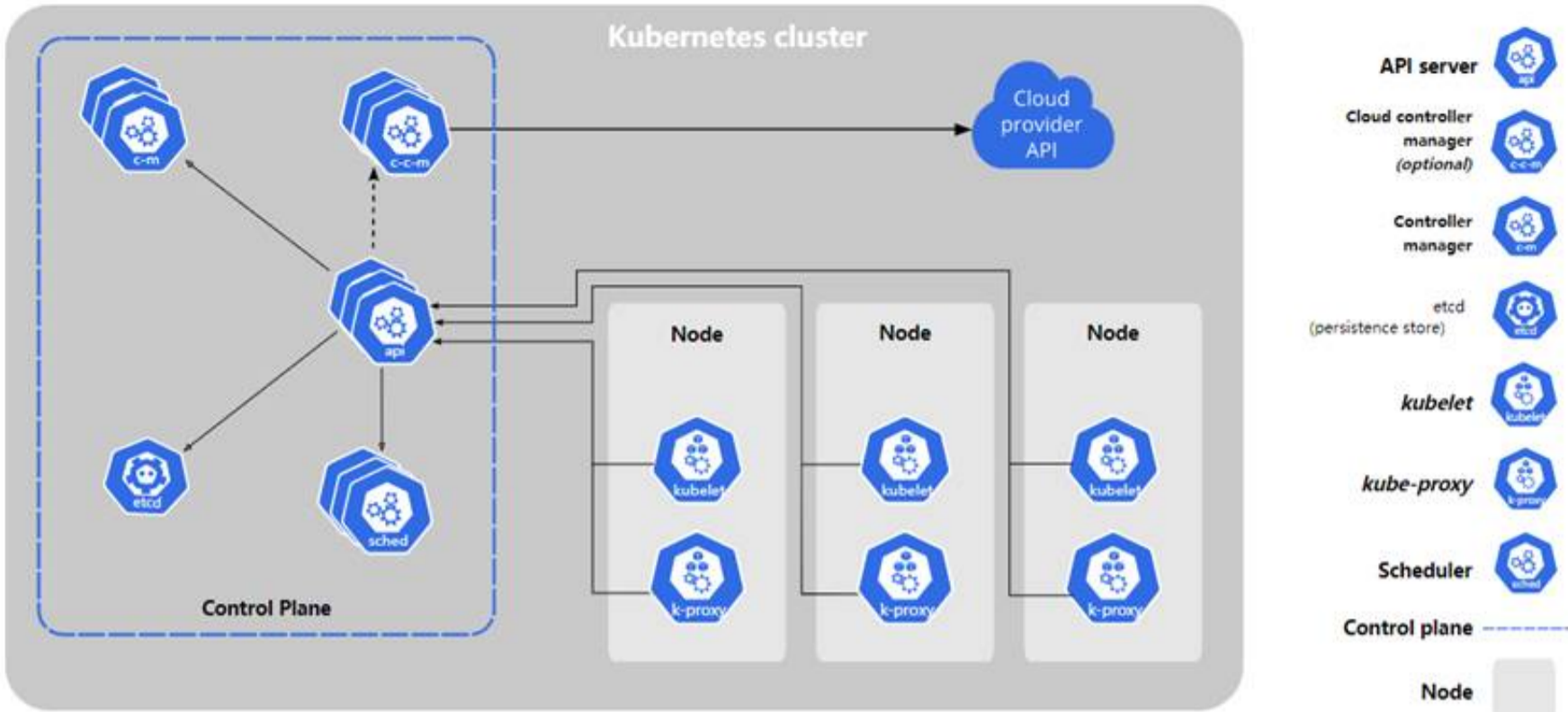
- A. kubelet
- B. systemd
- C. kube-proxy
- D. pod

**Answer: A**

**Explanation:**

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

Graphical user interface, diagram, application Description automatically generated



**NEW QUESTION 5**

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 6**

What Kubernetes resource would allow you to run one Pod on some of your Nodes?

- A. DaemonSet
- B. ClusterSet
- C. Deployment
- D. ReplicaSet

**Answer:** A

**Explanation:**

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

# 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

In a simple case, one DaemonSet, covering all nodes, would be used for each type of daemon. A more complex setup might use multiple DaemonSets for a single type of daemon, but with different flags and/or different memory and cpu requests for different hardware types.

## NEW QUESTION 7

What is the smallest possible unit in Kubernetes to run a container?

- A. pod
- B. docker
- C. service
- D. container

**Answer:** A

### Explanation:

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

# Pods

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

A *Pod* (as in a pod of whales or pea pod) is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context. A Pod models an application-specific "logical host": it contains one or more application containers which are relatively tightly coupled. In non-cloud contexts, applications executed on the same physical or virtual machine are analogous to cloud applications executed on the same logical host.

## NEW QUESTION 8

What kind of limitation cgroups allows?

- A. Prioritization
- B. Resource limiting
- C. Accounting
- D. None of the options
- E. Control
- F. Server cpu and memory

**Answer:** ABCE

## NEW QUESTION 9

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 10**

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

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

**Answer: A**

**Explanation:**

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

**CLOUD NATIVE COMPUTING FOUNDATION**

**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](https://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](https://cncf.io/csp)

**C. Join CNCF's End User Community**  
 For companies that don't offer cloud native services externally. [cncf.io/enduser](https://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](https://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, orc1/kubk
- 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 10**

What do you call the pattern where you add a second container to the pod to collect logs information?

- A. Sidecar container logging
- B. Node level logging
- C. Application level logging
- D. Cluster level logging

**Answer: A**

**Explanation:**

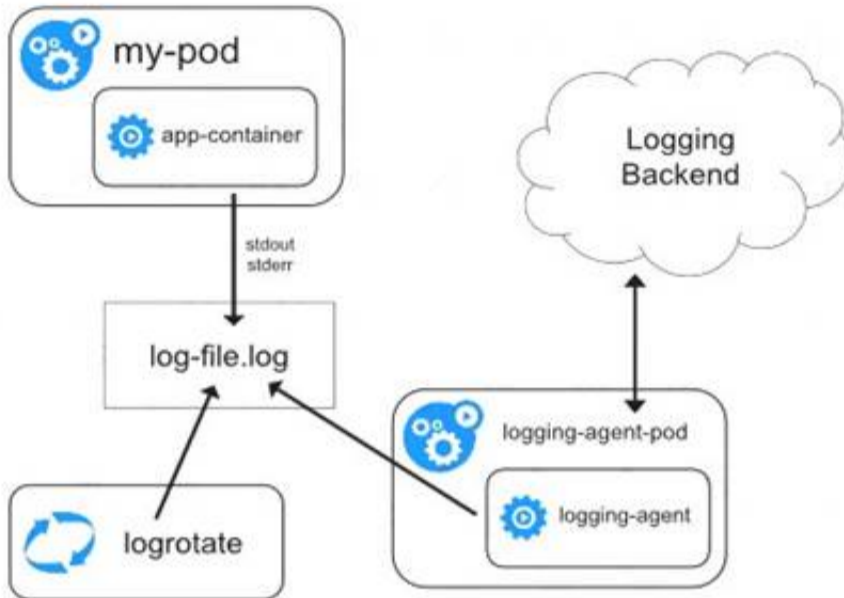
<https://kubernetes.io/docs/concepts/cluster-administration/logging/> Diagram Description automatically generated

# Cluster-level logging architectures

While Kubernetes does not provide a native solution for cluster-level logging, there are several common approaches you can consider. Here are some options:

- Use a node-level logging agent that runs on every node.
- Include a dedicated sidecar container for logging in an application pod.
- Push logs directly to a backend from within an application.

## Using a node logging agent [↔](#)



**NEW QUESTION 15**

A is a ready-to-run software package, containing everything needed to run an application.

- A. Container Repository
- B. Container Runtime
- C. Docker
- D. Container Image

**Answer:** D

**Explanation:**

<https://kubernetes.io/docs/concepts/containers/#container-images> Text, letter Description automatically generated

# Container images

A **container image** is a ready-to-run software package, containing everything needed to run an application: the code and any runtime it requires, application and system libraries, and default values for any essential settings.

By design, a container is immutable: you cannot change the code of a container that is already running. If you have a containerized application and want to make changes, you need to build a new image that includes the change, then recreate the container to start from the updated image.

**NEW QUESTION 20**

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

- A. Envoy

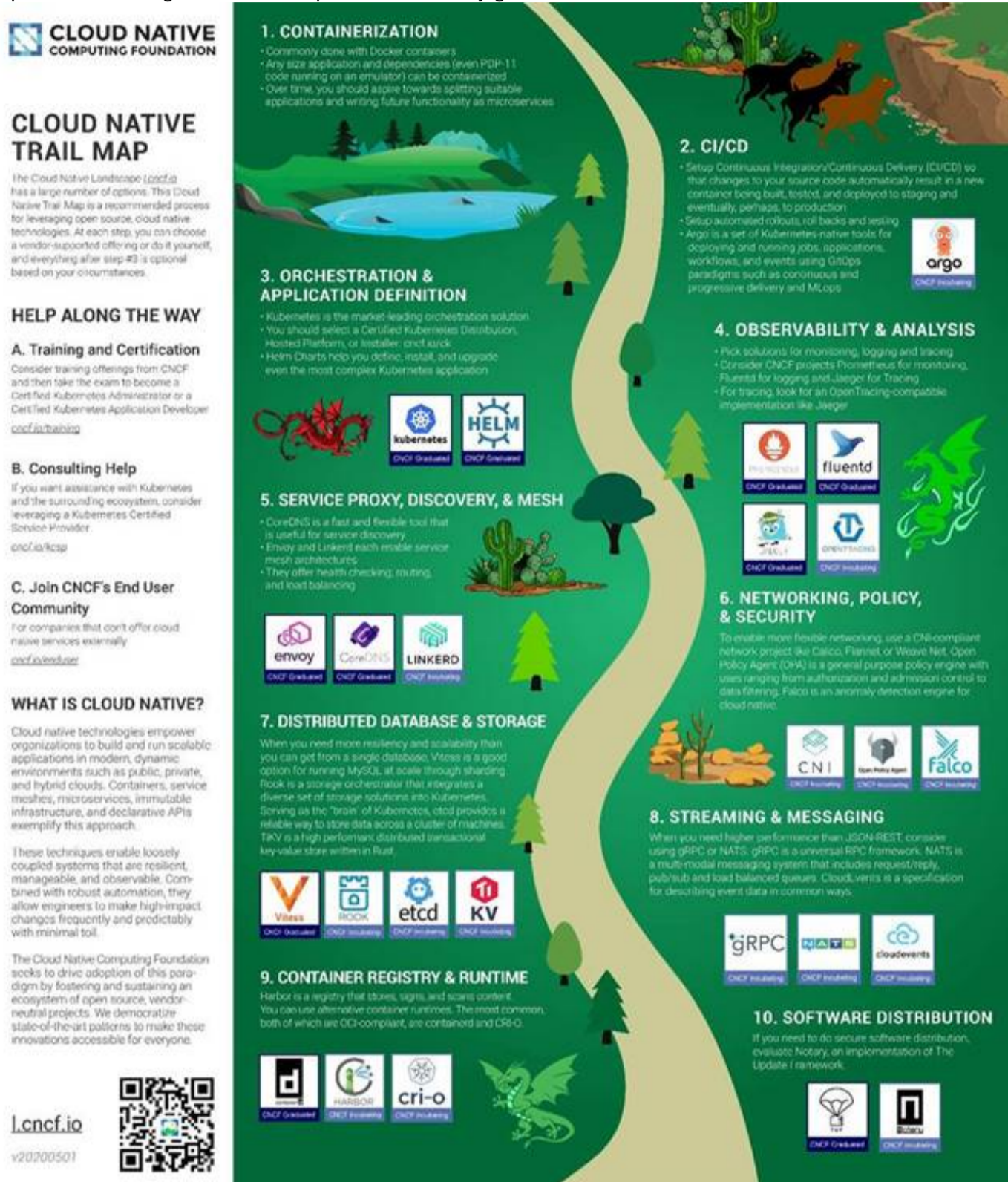
- B. Vitess
- C. Rook
- D. TiKV

**Answer: A**

**Explanation:**

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

A picture containing timeline Description automatically generated



**NEW QUESTION 23**

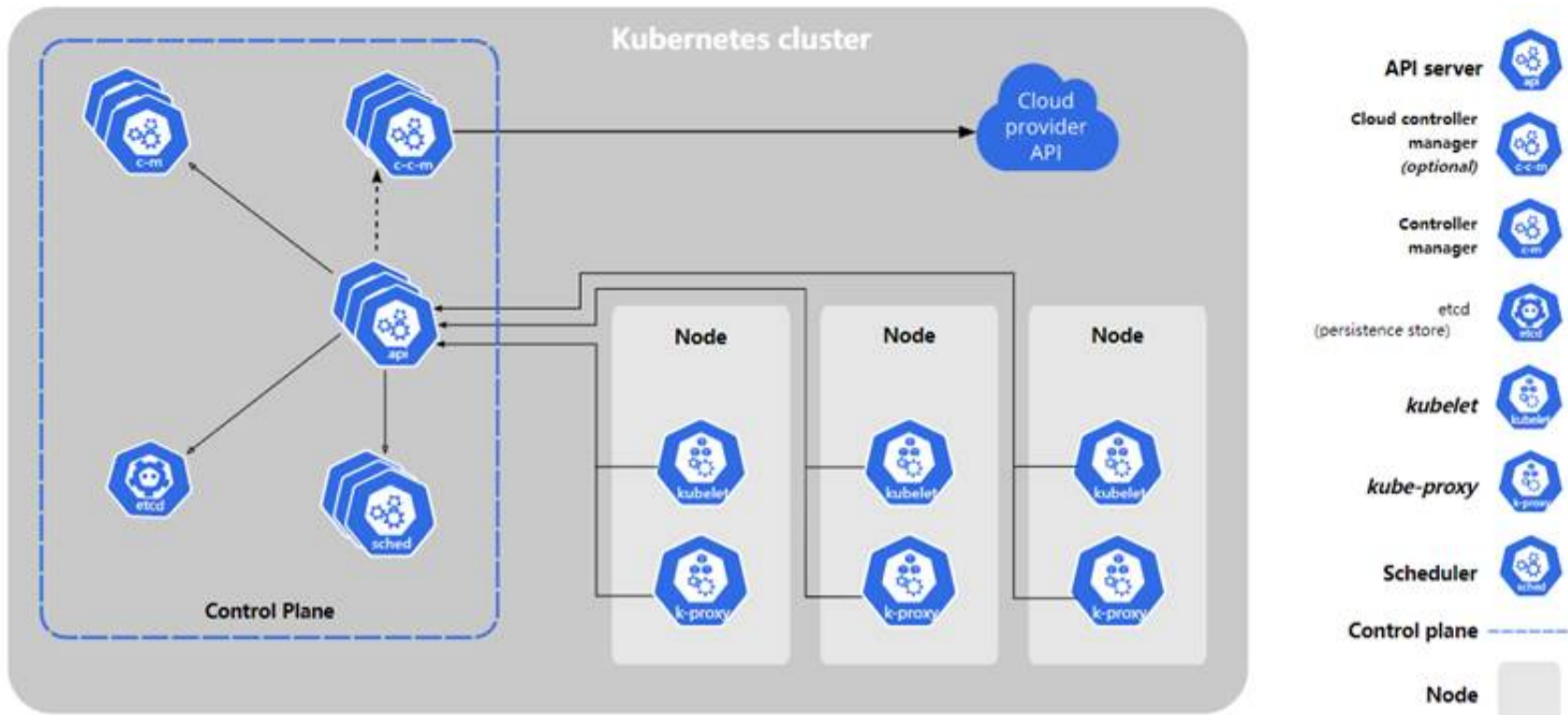
Which of the following components is part of the Kubernetes control panel

- A. kubectl
- B. kube-proxy
- C. Service Mesh
- D. kubelet
- E. Cloud control manager

**Answer: E**

**Explanation:**

<https://kubernetes.io/docs/concepts/overview/components/> Diagram Description automatically generated



**NEW QUESTION 26**

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 30**

Which authentication method allows JWTs to authenticate?

- A. OpenId connect
- B. Client 'TLS' certificates
- C. OPA gatekeeper
- D. Anonymous

**Answer: A**

**NEW QUESTION 32**

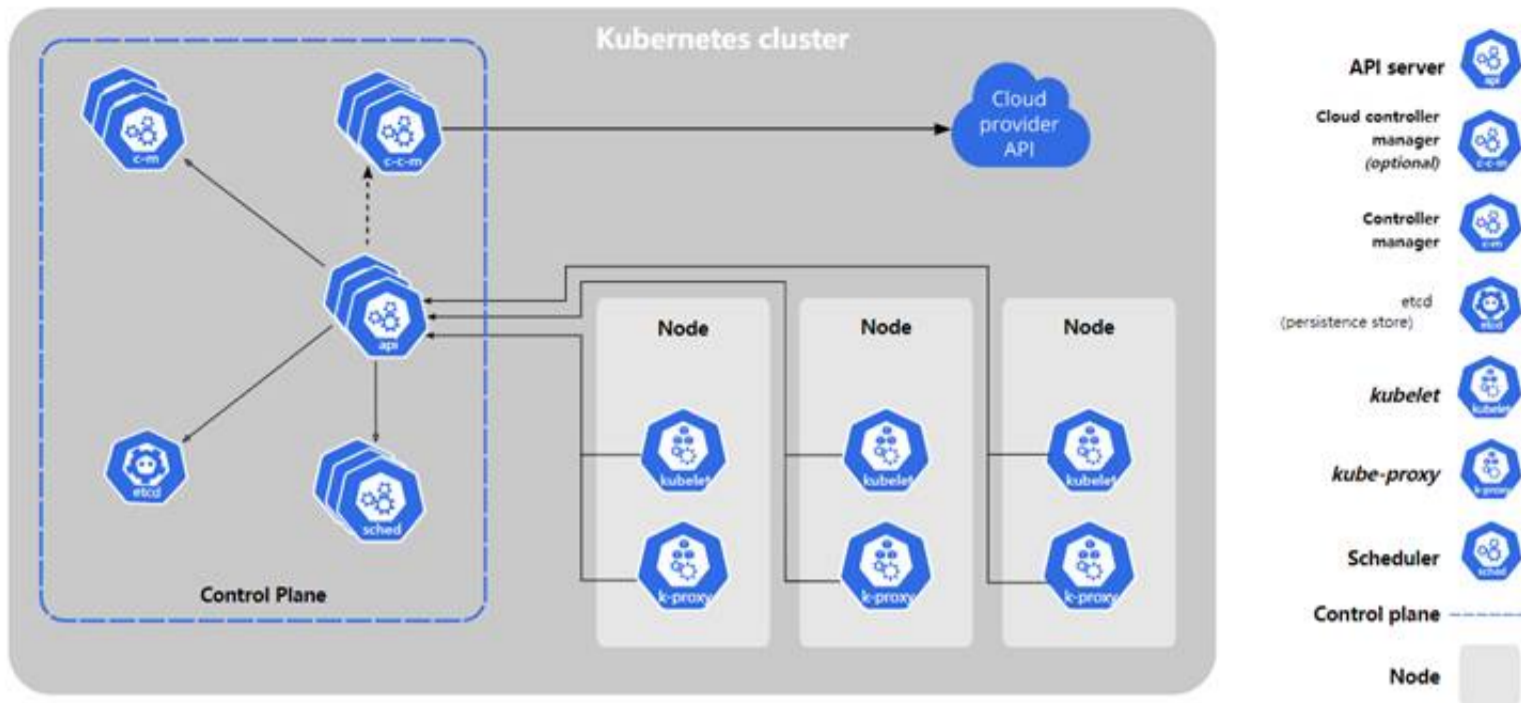
Which of the following is not the part of Kubernetes Control Plane?

- 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 36**

Fluentd is the only way to export logs from Kubernetes cluster or applications running in cluster

- A. True
- B. False

**Answer: B**

**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](https://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/ksap](https://cncf.io/ksap)

**C. Join CNCF's End User Community**

For companies that don't offer cloud native services externally [cncf.io/enduser](https://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](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/kick](https://cncf.io/kick)
  - 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-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 39**

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 41**

What are the two major components of service mesh?

- A. Control plane and Data plane
- B. Master plane and Data plane
- C. None of the options
- D. Controller plane and User plane
- E. Master plane and User plane

**Answer:** A

**Explanation:**

<https://istio.io/latest/about/service-mesh/>

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

## How it Works

Istio has two components: the data plane and the control plane.

The data plane is the communication between services. Without a service mesh, the network doesn't understand the traffic being sent over, and can't make any decisions based on what type of traffic it is, or who it is from or to.

**NEW QUESTION 42**

What is OPA?

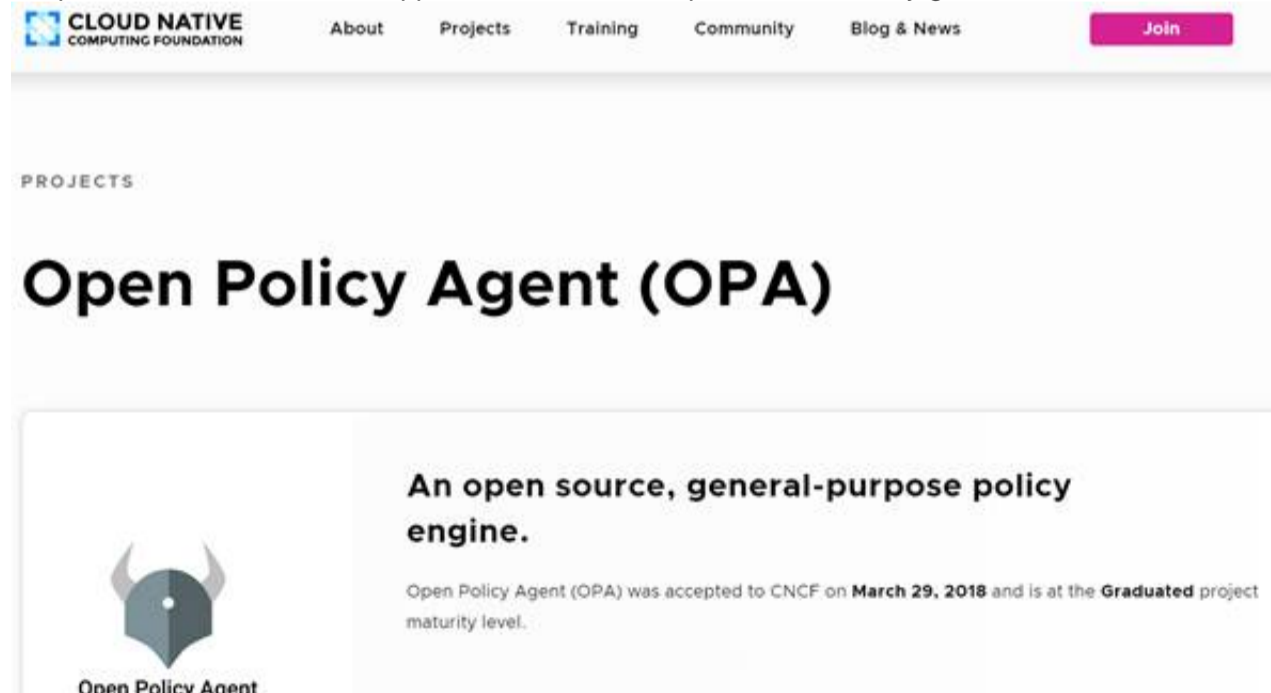
- A. Open Permission Agent
- B. Online Policy Audit
- C. Open Policy Agent
- D. Offline Policy Accessor

**Answer:** C

**Explanation:**

<https://www.cncf.io/projects/open-policy-agent-opa/>

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



**NEW QUESTION 45**

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 50**

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

### 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
```

**NEW QUESTION 51**

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 56**

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 57**

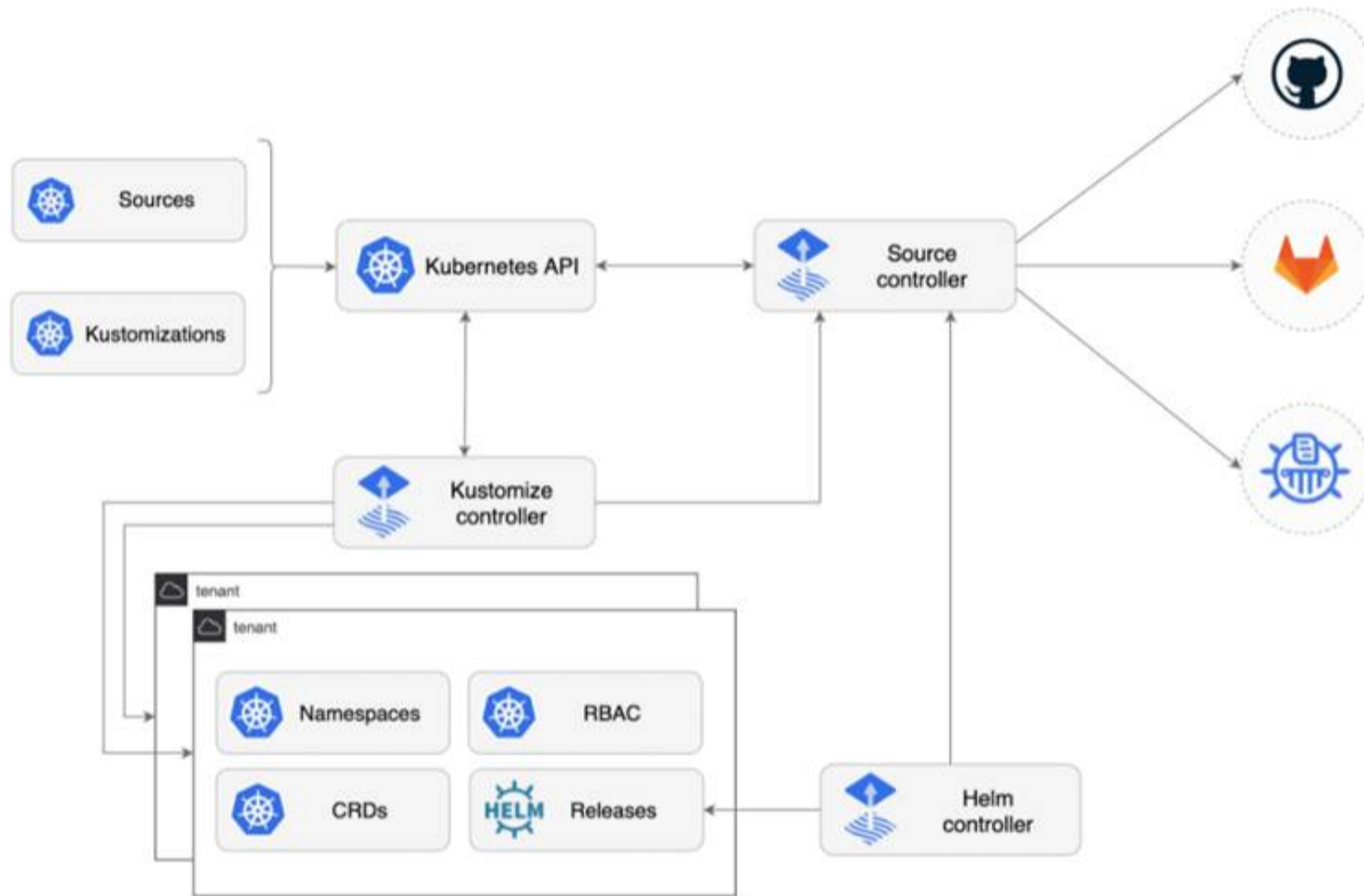
What do GitOps tools do in kubernetes?

- A. They allow us to make changes to a kubernetes cluster using a Git repository
- B. They manage the source code of kubernetes itself
- C. They allow us to store software code in Git
- D. They allows us to store container images in repositories

**Answer:** A

**Explanation:**

<https://fluxcd.io/docs/components/>  
 Diagram Description automatically generated



**NEW QUESTION 62**

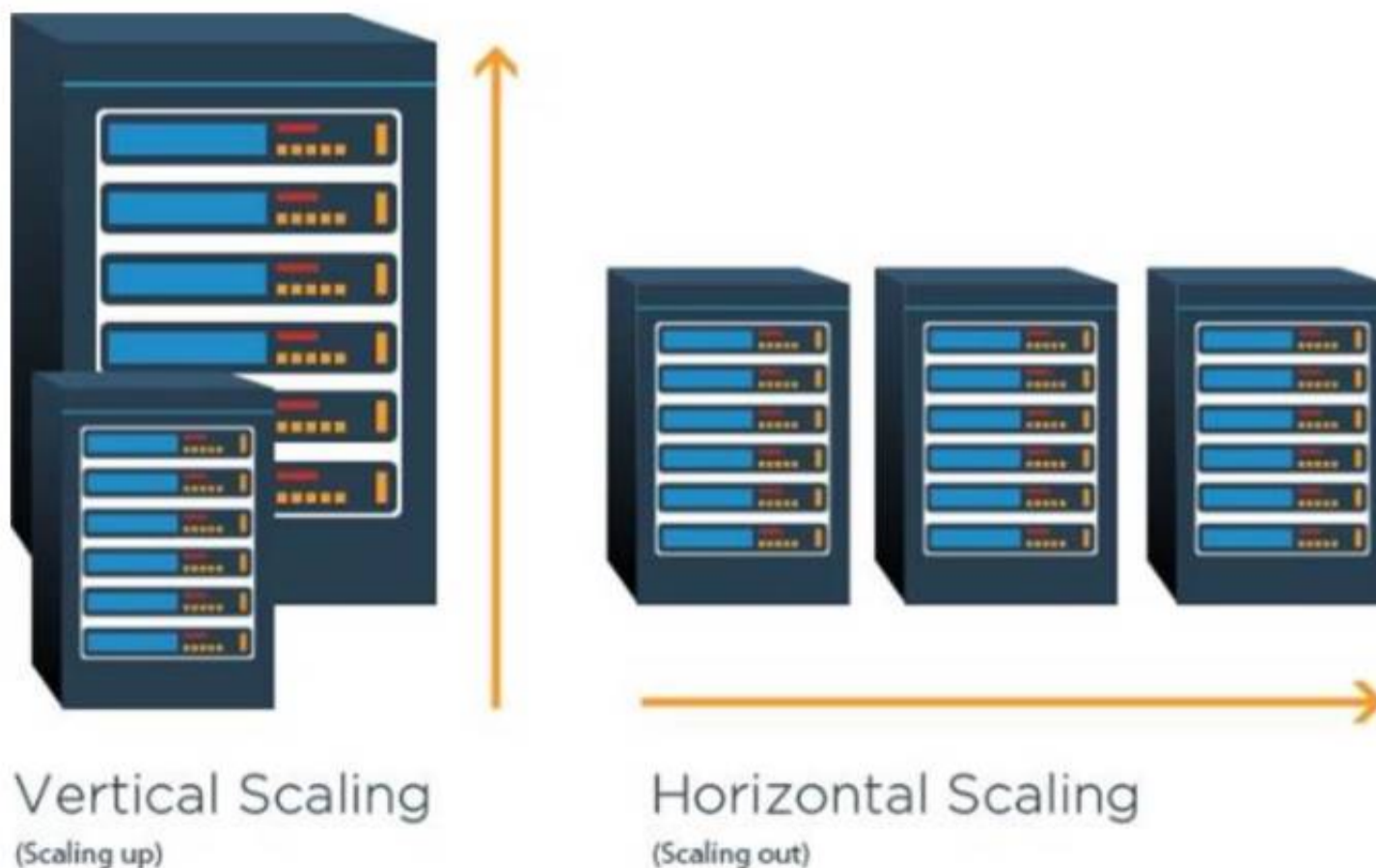
An application that is nearing its usage limit. To increase the amount of users it can handle, you allo-cate additional memory resources to each instance of the application. What type of scaling is this?

- A. Horizontal Scaling
- B. Cluster Autoscaling
- C. Recursive Scaling
- D. Vertical Scaling

**Answer:** D

**Explanation:**

Graphical user interface, diagram Description automatically generated



#### NEW QUESTION 64

Which part of a Kubernetes cluster is responsible for running container workloads?

- A. Worker Node
- B. kube-proxy
- C. Control plane
- D. etcd

**Answer:** A

**Explanation:**

Worker Nodes are responsible for executing containerized workloads.

#### NEW QUESTION 68

Which of the following is an example of vertical scaling?

- A. Using cluster autoscaler
- B. Adding more resources (memory and/or cpu) to a kubernetes node
- C. Adding more nodes to kubernetes cluster
- D. Adding more replica pods to a deployment

**Answer:** B

**Explanation:**

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

Horizontal scaling means that the response to increased load is to deploy more Pods. This is different from *vertical* scaling, which for Kubernetes would mean assigning more resources (for example: memory or CPU) to the Pods that are already running for the workload.

#### NEW QUESTION 69

What is the name for the tool that manages communication between pods, injects a sidecar proxy container into each pod and directs network traffic through the proxy container?

- A. namespace
- B. Deployment
- C. Network policy
- D. Service mesh
- E. Service

**Answer:** D

#### NEW QUESTION 72

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 `kubect1` 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 77

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 78

Which of the following is not the required field to describe Kubernetes objects?

- A. metadata
- B. apiVersion
- C. Kind
- D. Container
- E. spec

**Answer:** D

#### Explanation:

<https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/> Graphical user interface, text, application Description automatically generated

## Required Fields

In the `.yaml` file for the Kubernetes object you want to create, you'll need to set values for the following fields:

- `apiVersion` - Which version of the Kubernetes API you're using to create this object
- `kind` - What kind of object you want to create
- `metadata` - Data that helps uniquely identify the object, including a `name` string, `UID`, and optional `namespace`
- `spec` - What state you desire for the object

The precise format of the object `spec` is different for every Kubernetes object, and contains nested fields specific to that object. The [Kubernetes API Reference](#) can help you find the `spec` format for all of the objects you can create using Kubernetes.

### NEW QUESTION 82

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

**Answer:** D

**Explanation:**

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

**NEW QUESTION 84**

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

**Answer:** C

**Explanation:**

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

# 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.

**NEW QUESTION 89**

Have a pod 'hello' and a container in that pod 'green'. Which of the following commands would get the logs for that container?

- A. alias k='kubectl'k logs -p hello -c green
- B. alias k='kubectl'k logs hello -c green
- C. alias k='kubectl'k get logs -p hello -c green
- D. alias k='kubectl'k logs -p hello green

**Answer:** B

**Explanation:**

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

## logs

Print the logs for a container in a pod or specified resource. If the pod has only one container, the container name is optional.

### Usage

```
$ kubectl logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER]
```

### Flags

Name	Shorthand	Default	Usage
all-containers		false	Get all containers' logs in the pod(s).
container	c		Print the logs of this container
follow	f	false	Specify if the logs should be streamed.

Return snapshot logs from pod nginx with only one container

```
kubectl logs nginx
```

Return snapshot logs from pod nginx with multi containers

```
kubectl logs nginx --all-containers=true
```

Return snapshot logs from all containers in pods defined by label app=nginx

```
kubectl logs -l app=nginx --all-containers=true
```

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

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

Begin streaming the logs of the ruby container in pod web-1

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

**NEW QUESTION 91**

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

**Answer:** D

**NEW QUESTION 94**

Observability and monitoring are not the same?

- A. True
- B. False

**Answer:** A

**NEW QUESTION 99**

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 102**

Continuous delivery is .

- A. Manually deploying the code
- B. Coding, Building and Testing the code
- C. Automatically deploying code to [container or server] environment

**Answer:** C

**NEW QUESTION 107**

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 109

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

Answer: C

### Explanation:

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

## 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.



The screenshot shows a list of kubectl edit command examples:

- Edit the service named 'registry'**  
`kubectl edit svc/registry`
- Use an alternative editor**  
`KUBE_EDITOR="nano" kubectl edit svc/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-config`
- Edit the deployment/mydeployment's status subresource**  
`kubectl edit deployment mydeployment --subresource='status'`

## NEW QUESTION 110

Which is NOT a use case for the Kubernetes dashboard?

- A. Troubleshooting any issues with applications
- B. Managing running applications
- C. Installing new Kubernetes cluster
- D. Managing the entire Kubernetes cluster

Answer: C

## NEW QUESTION 111

What cloud-native construct does a kubernetes pod wrap?

- A. Container
- B. Virtual Machine (VM)

- C. side car process
- D. Docker image

**Answer:** A

**Explanation:**

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

**NEW QUESTION 115**

Which control plane component is responsible for scheduling pods?

- A. kube-proxy
- B. kube scheduler
- C. kubelet
- D. kube api-server

**Answer:** B

**Explanation:**

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

Graphical user interface, text, application Description automatically generated

## kube-scheduler

Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

**NEW QUESTION 118**

Which style of operations are preferred for K8S and cloud native applications?

- A. JSON
- B. Declarative
- C. Imperative

**Answer:** B

**Explanation:**

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

**NEW QUESTION 121**

What is FinOps?

- A. The first step in any cloud transformation
- B. Stage beyond DevOps or DevSecOps, where organization transition to serverless technologies
- C. Using data to make cost savings decisions about cloud usage
- D. Specialized cloud features used by financial industries (example: banks, insurance, etc)

**Answer:** C

**Explanation:**

<https://www.servicenow.com/products/it-asset-management/what-is-finops.html> Text, letter Description automatically generated

## What is the origin of FinOps?

Unlike many modern tech movements, FinOps is not a single advancement or policy change pioneered by any specific company or organization; it's a natural evolution of technology management to account for on-demand cloud resources.

With the rise and proliferation of cloud computing in the new millennium, many companies began to see a shift from standard, traditional pricing to usage-based pricing models. And, while this allowed businesses to take a more cost effective approach to technology—paying only for the time and resources they used, rather than paying a set rate—it created a crisis for CFOs. After all, it's next to impossible to predict tool usage with any degree of accuracy, which can make budgeting an exercise in futility.

To address this issue, prevent runaway expenses, and promote business profitability, organizations around the world began to develop the concept of financial operations, (FinOps). This revolution was guided by respected technology companies around the world, first taking shape as cloud cost management, developing into cloud cost optimization, and then into cloud financial management.

Finally, taking inspiration from the success of DevOps, FinOps was born, bringing cross-functionality and agility to financial management of cloud technologies.

### NEW QUESTION 122

What is a benefits of Kubernetes federation?

- A. Avoids scalability limits on pods and nodes
- B. Creates highly available clusters in different regions
- C. Low latency

**Answer:** ABC

### NEW QUESTION 126

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](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 130

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 133

Which organizational persona creates Service Level Agreements 'SLA', Service Level Objectives 'SLO', and Service Level Indicator 'SLI'?

- A. Developer
- B. DevSecOps
- C. Site Reliability Engineer (SRE)
- D. Security and Compliance Engineer
- E. DevOps

**Answer:** C

#### Explanation:

SREs create SLAs, SLOs, and SLIs to define and implement standards for application and infra-structure reliability.

#### NEW QUESTION 138

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

**Answer:** D

#### Explanation:

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

#### NEW QUESTION 142

Various Container Orchestrator Systems (COS)?

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

**Answer:** ACD

#### NEW QUESTION 146

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 150

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 155

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 157

What is the command used to login to the pod?

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

**Answer:** C

### Explanation:

<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
```

Text Description automatically generated

## NEW QUESTION 159

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

**Answer:** E

## NEW QUESTION 162

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

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

**Answer:** A

### Explanation:

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

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

## 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](#).

### NEW QUESTION 167

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 168

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### KCNA Practice Exam Features:

- \* KCNA Questions and Answers Updated Frequently
- \* KCNA Practice Questions Verified by Expert Senior Certified Staff
- \* KCNA Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* KCNA Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
[Order The KCNA Practice Test Here](#)