Linux-Foundation

Exam Questions KCNA

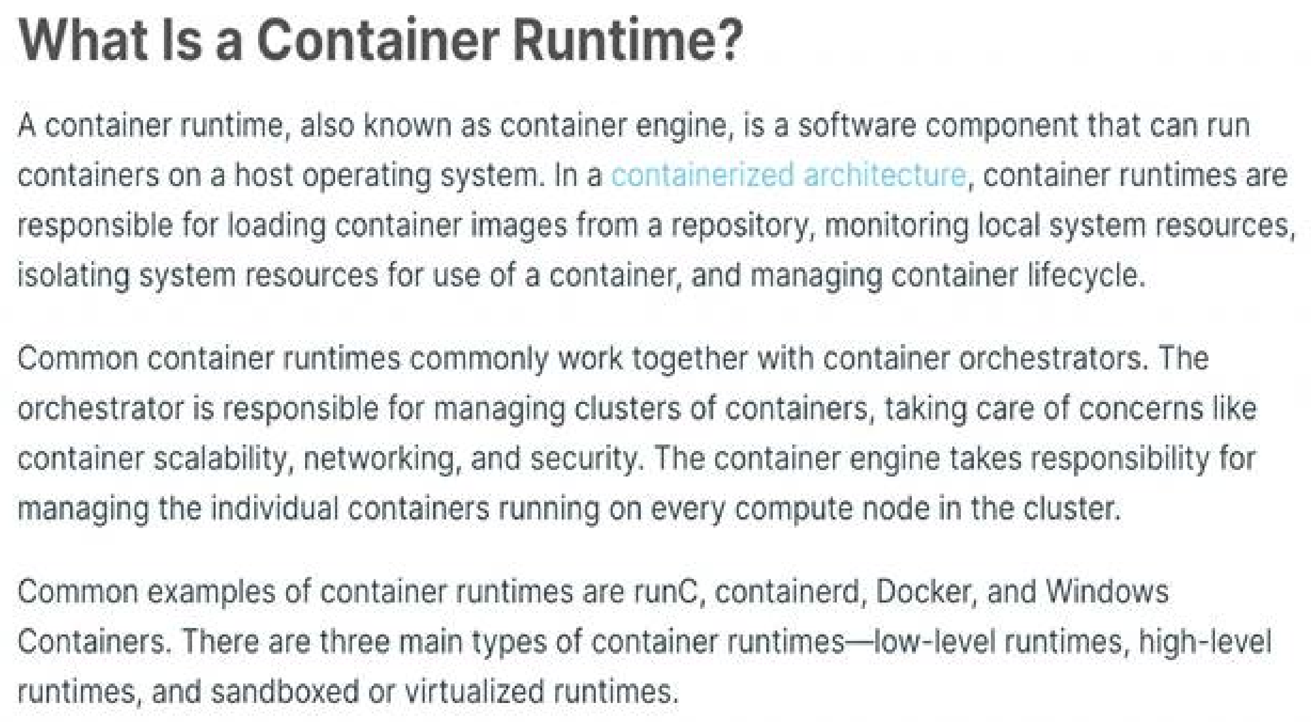
Kubernetes and Cloud Native Associate (KCNA)

NEW QUESTION 1
What is container runtime?

A. The amount of time it takes a container to execute
B. A container image format
C. Another term of kubelet or kubectl
D. Software that runs containers

Answer: D

Explanation:
https://www.aquasec.com/cloud-native-academy/container-security/container-runtime/ Text Description automatically generated



What Is a Container Runtime?

A container runtime, also known as container engine, is a software component that can run containers on a host operating system. In a containerized architecture, container runtimes are responsible for loading container images from a repository, monitoring local system resources, isolating system resources for use of a container, and managing container lifecycle.

Common container runtimes commonly work together with container orchestrators. The orchestrator is responsible for managing clusters of containers, taking care of concerns like container scalability, networking, and security. The container engine takes responsibility for managing the individual containers running on every compute node in the cluster.

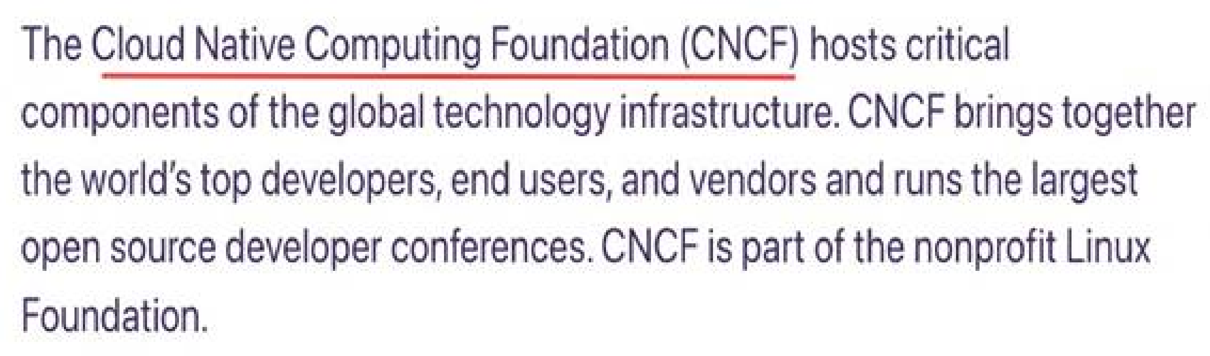
Common examples of container runtimes are runC, containerd, Docker, and Windows Containers. There are three main types of container runtimes—low-level runtimes, high-level runtimes, and sandboxed or virtualized runtimes.

NEW QUESTION 2
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



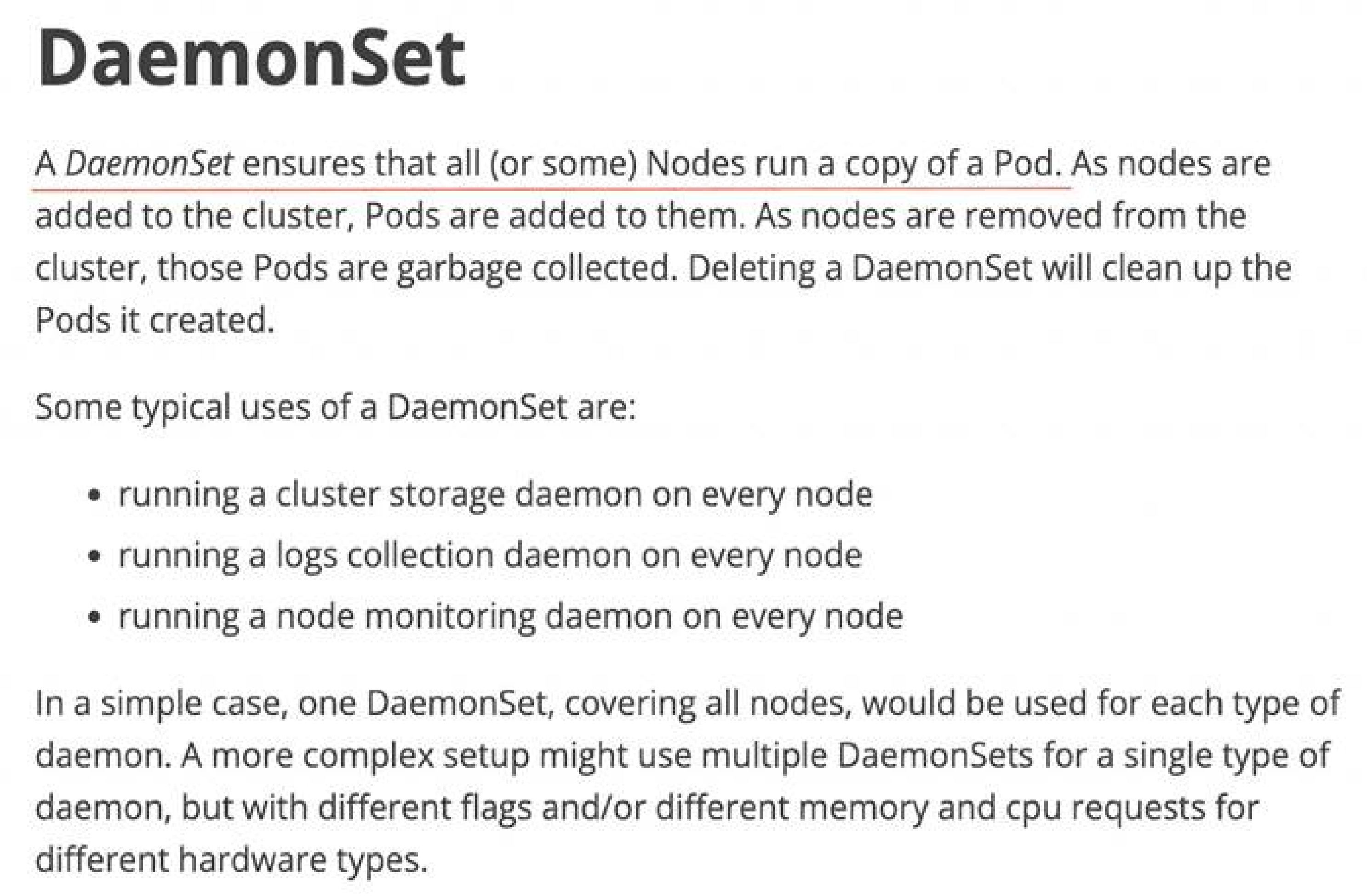
The Cloud Native Computing Foundation (CNCF) hosts critical components of the global technology infrastructure. CNCF brings together the world's top developers, end users, and vendors and runs the largest open source developer conferences. CNCF is part of the nonprofit Linux Foundation.

NEW QUESTION 3
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

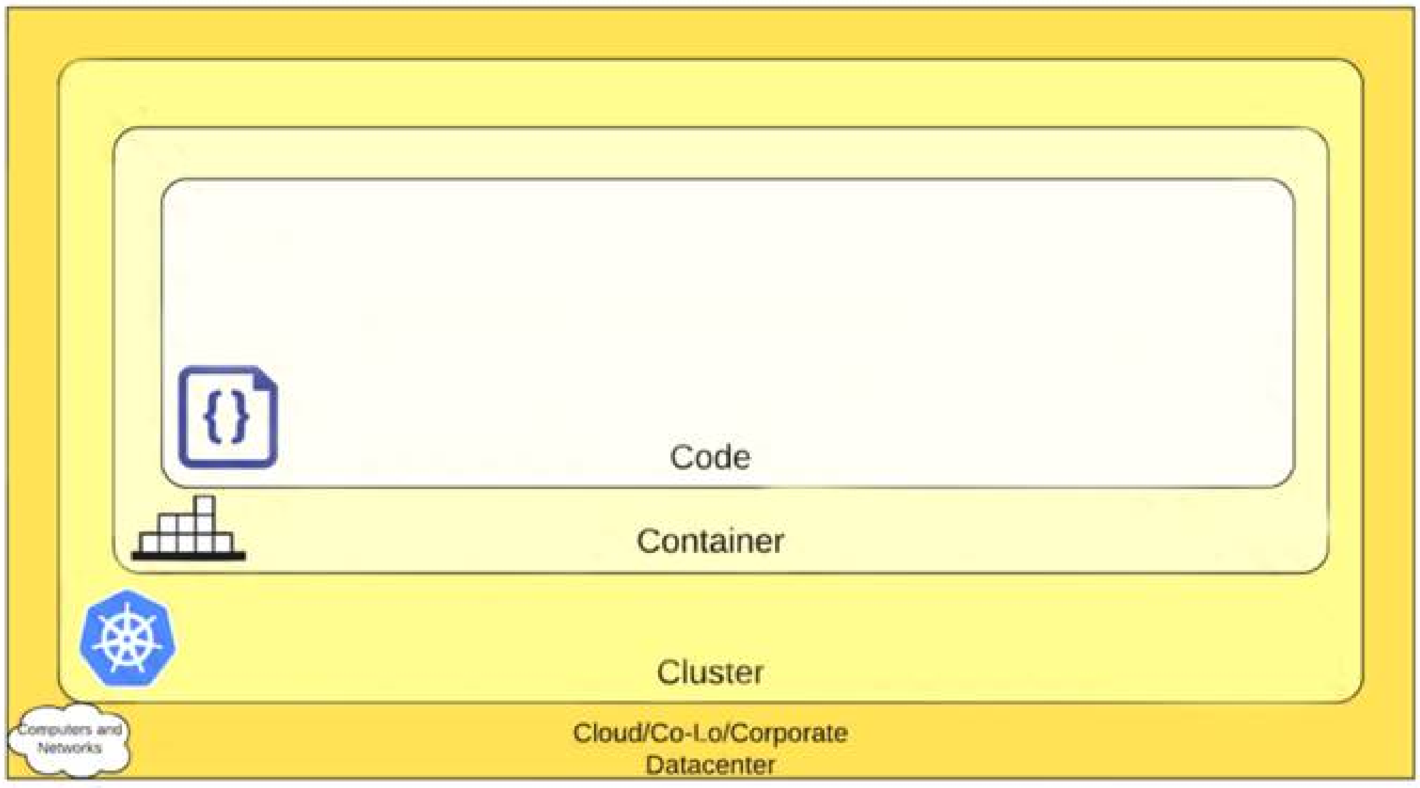
In a simple case, one DaemonSet, covering all nodes, would be used for each type of daemon. A more complex setup might use multiple DaemonSets for a single type of daemon, but with different flags and/or different memory and cpu requests for different hardware types.

NEW QUESTION 4
There are three Nodes in a cluster, and want to run exactly one replica of a Pod on each Node. Pre-fer 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 5
What is the main difference between Argo vs. Flux CD?

A. Argo is pull-based, and Flux is push-based
B. No difference; both are pull-based
C. Argo is push-based, and Flux is pull-based
D. No difference; both are push-based

Answer: C

Explanation:
AgroCD:
https://argo-cd.readthedocs.io/en/stable/developer-guide/ci/#can-i-retrigger-the-checks-without-pushing-a-new-c
FluxCD: https://fluxcd.io/

NEW QUESTION 6
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

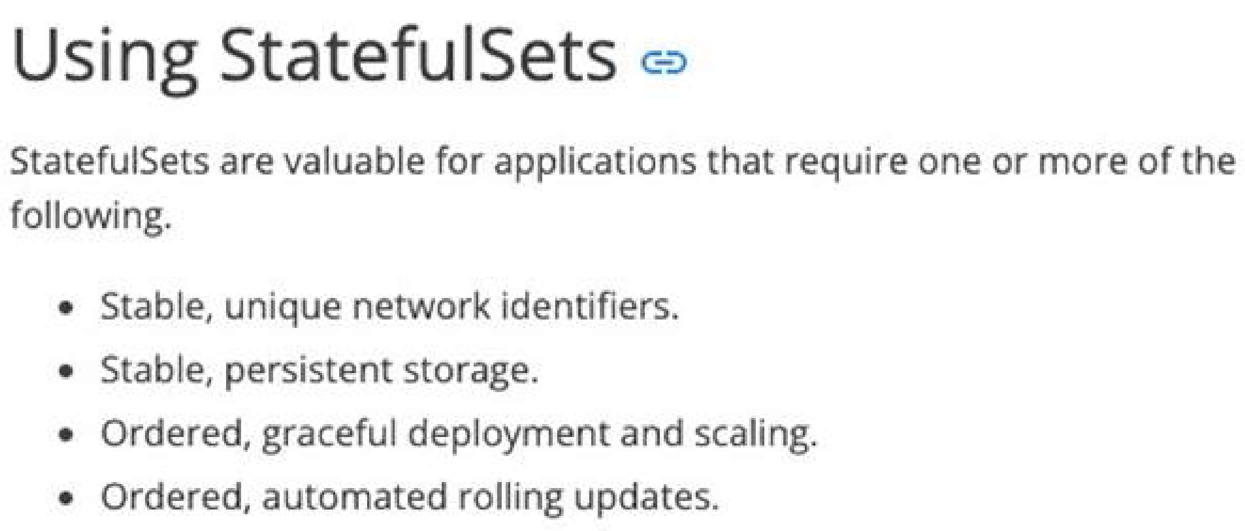
Explanation:
kubelet can communicate with any runtime that supports the CRI standard.

NEW QUESTION 7
Stateful set requires which service for the network identity of pods?

A. Ingress
B. Load Balancer Service
C. Headless Service

Answer: C

Explanation:
https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/ Graphical user interface, text, application Description automatically generated



Using StatefulSets

StatefulSets are valuable for applications that require one or more of the following.

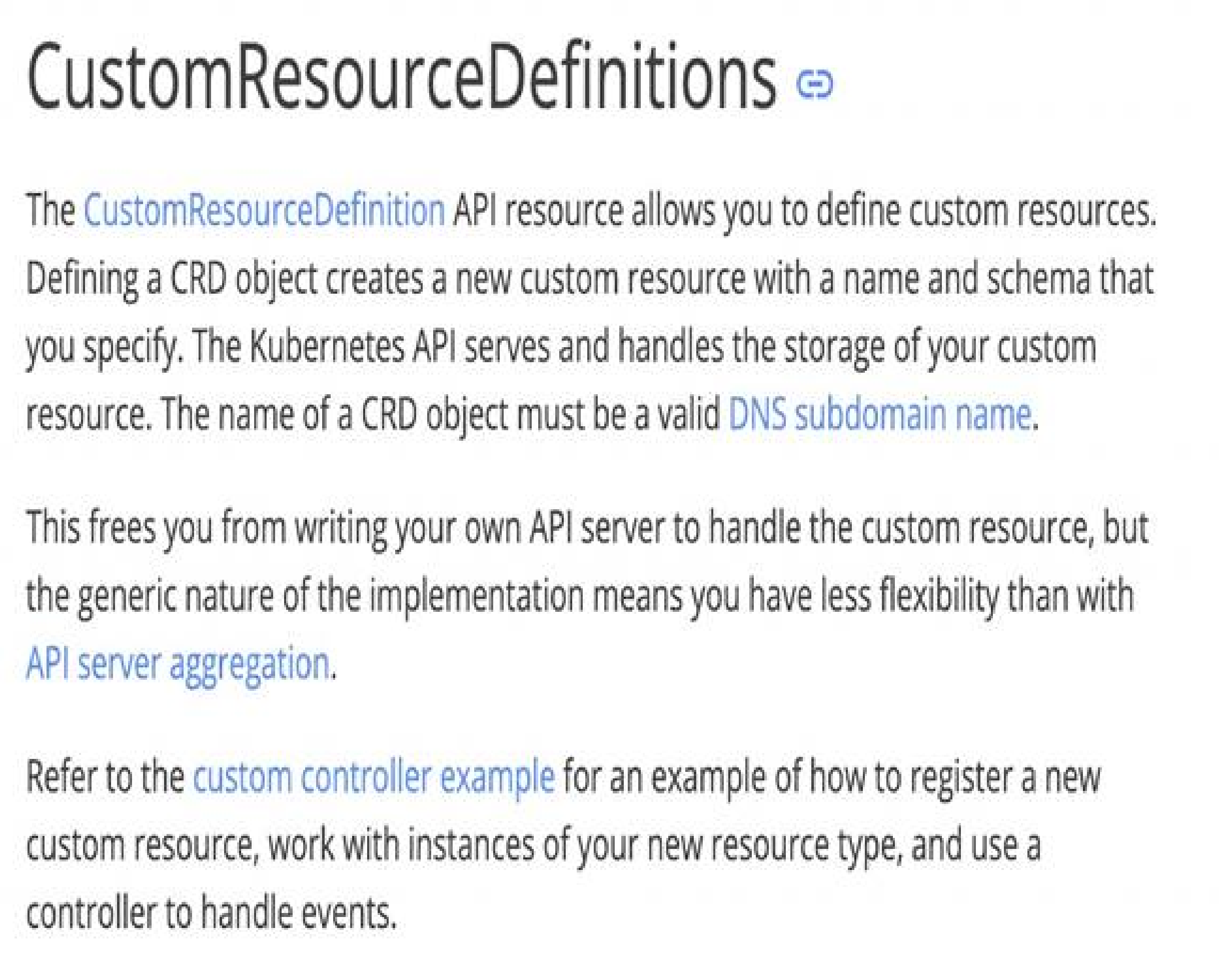
- Stable, unique network identifiers.
- Stable, persistent storage.
- Ordered, graceful deployment and scaling.
- Ordered, automated rolling updates.

NEW QUESTION 8
What can you use to add new resource types to your cluster?

A. start container
B. CustomResourceDefinitions
C. init container
D. Flux
E. CRI-O

Answer: B

Explanation:
https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/ Graphical user interface, text, application Description automatically generated



CustomResourceDefinitions

The CustomResourceDefinition API resource allows you to define custom resources. Defining a CRD object creates a new custom resource with a name and schema that you specify. The Kubernetes API serves and handles the storage of your custom resource. The name of a CRD object must be a valid DNS subdomain name.

This frees you from writing your own API server to handle the custom resource, but the generic nature of the implementation means you have less flexibility than with API server aggregation.

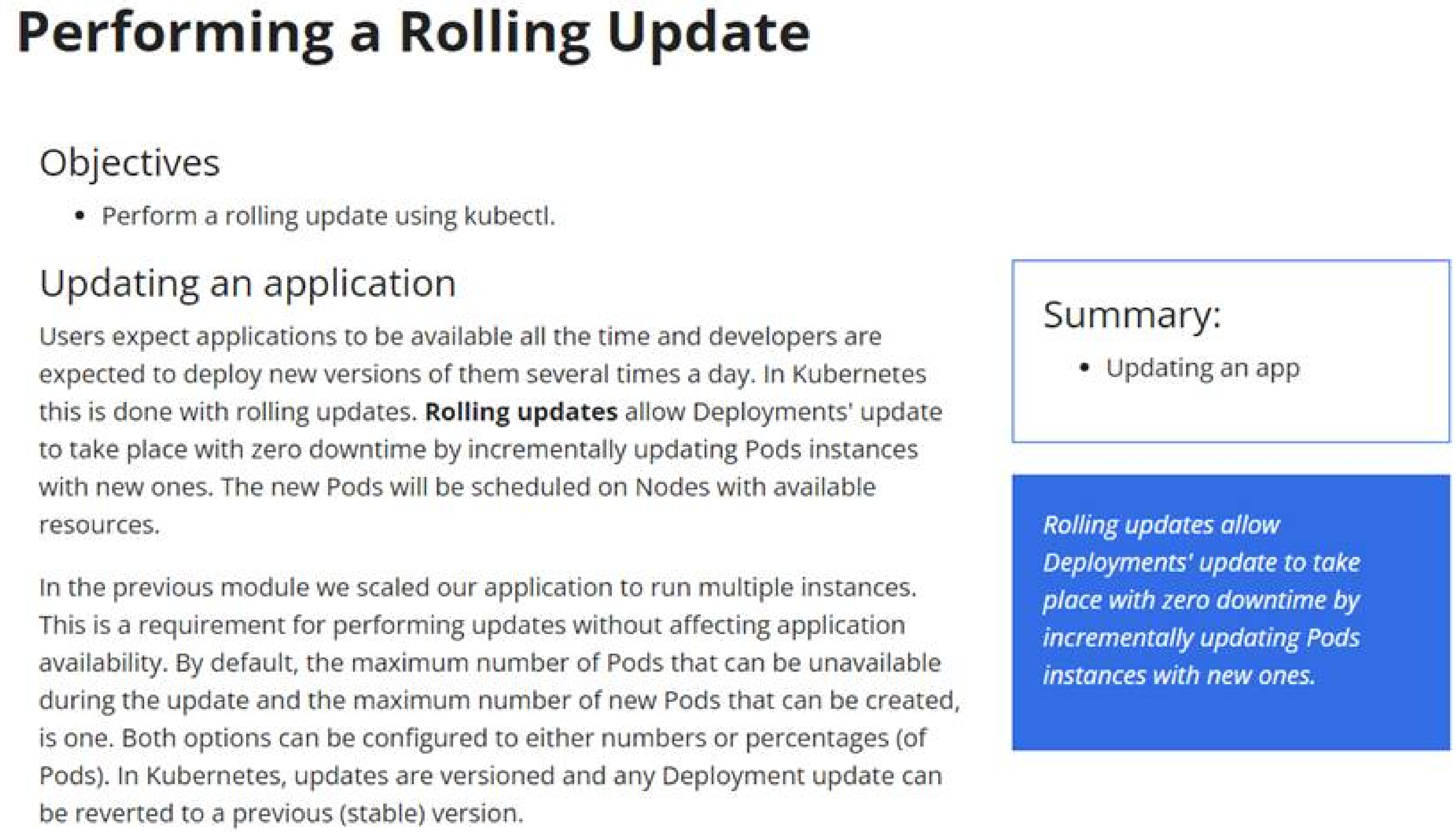
Refer to the custom controller example for an example of how to register a new custom resource, work with instances of your new resource type, and use a controller to handle events.

NEW QUESTION 9
The Kubernetes rolling update is used for _______ .

A. Updating a service
B. Scaling an application
C. Updating a deployment

Answer: C

Explanation:
https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/
Graphical user interface, text Description automatically generated with medium confidence



Performing a Rolling Update

Objectives

- Perform a rolling update using kubectl.

Updating an application

Users expect applications to be available all the time and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. **Rolling updates** allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones. The new Pods will be scheduled on Nodes with available resources.

In the previous module we scaled our application to run multiple instances. This is a requirement for performing updates without affecting application availability. By default, the maximum number of Pods that can be unavailable during the update and the maximum number of new Pods that can be created, is one. Both options can be configured to either numbers or percentages (of Pods). In Kubernetes, updates are versioned and any Deployment update can be reverted to a previous (stable) version.

Summary:

- Updating an app

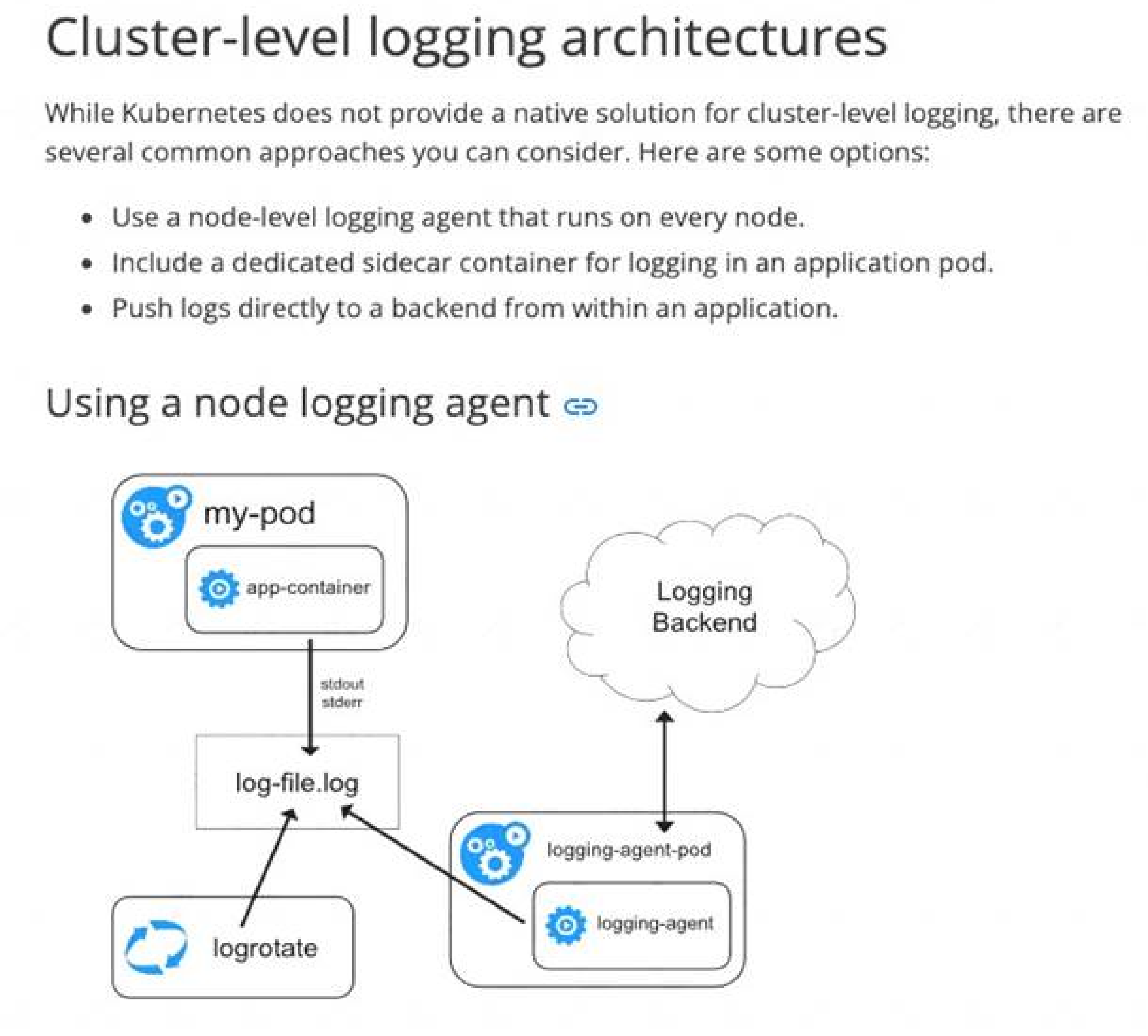
Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones.

NEW QUESTION 10
What do you call the pattern where you add a second container to the pod to collect logs infor-mation?

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.

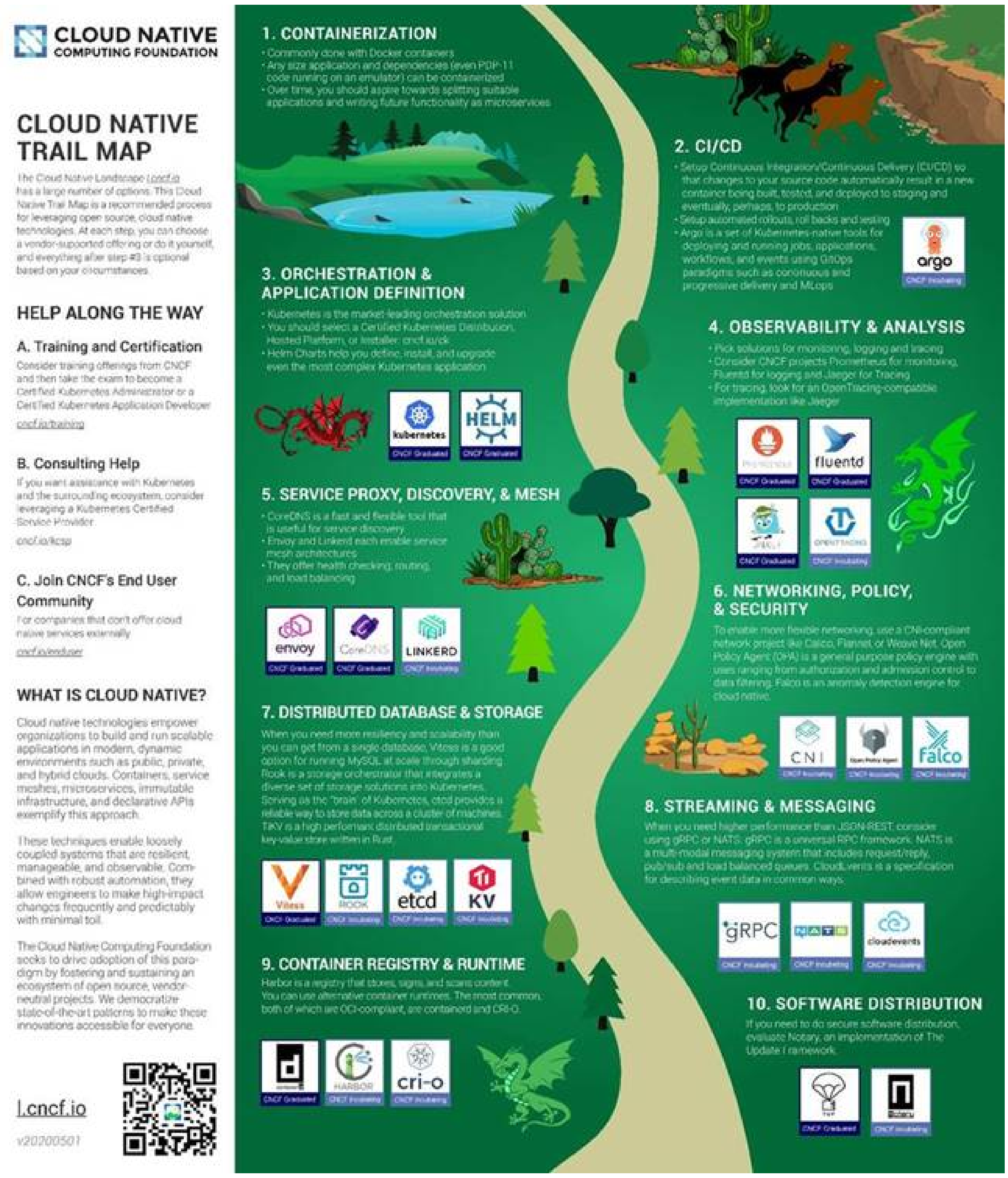
Using a node logging agent

NEW QUESTION 10
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 l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer

cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. 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

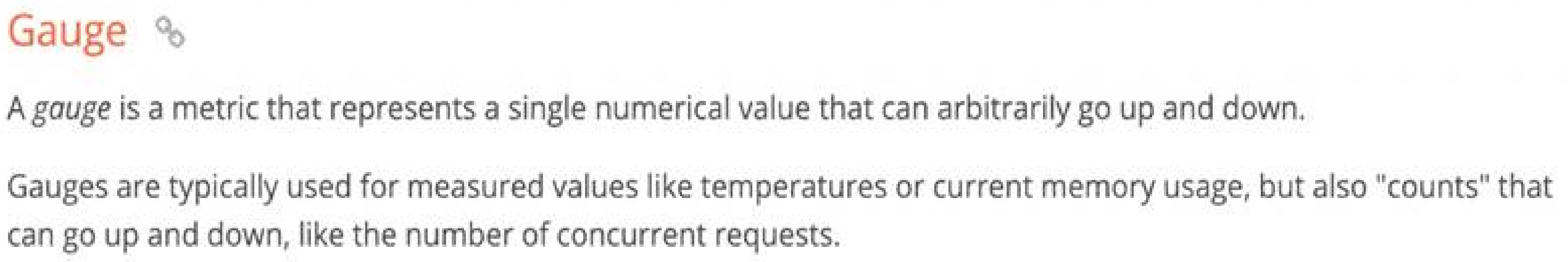
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

NEW QUESTION 14
Which prometheus metric type represents a single number value that can increase and decrease over time?

A. Gauge
B. Histogram
C. Summary
D. Counter

Answer: A

Explanation:
https://prometheus.io/docs/concepts/metric_types/#gauge Graphical user interface, text Description automatically generated



Gauge

A *gauge* is a metric that represents a single numerical value that can arbitrarily go up and down.

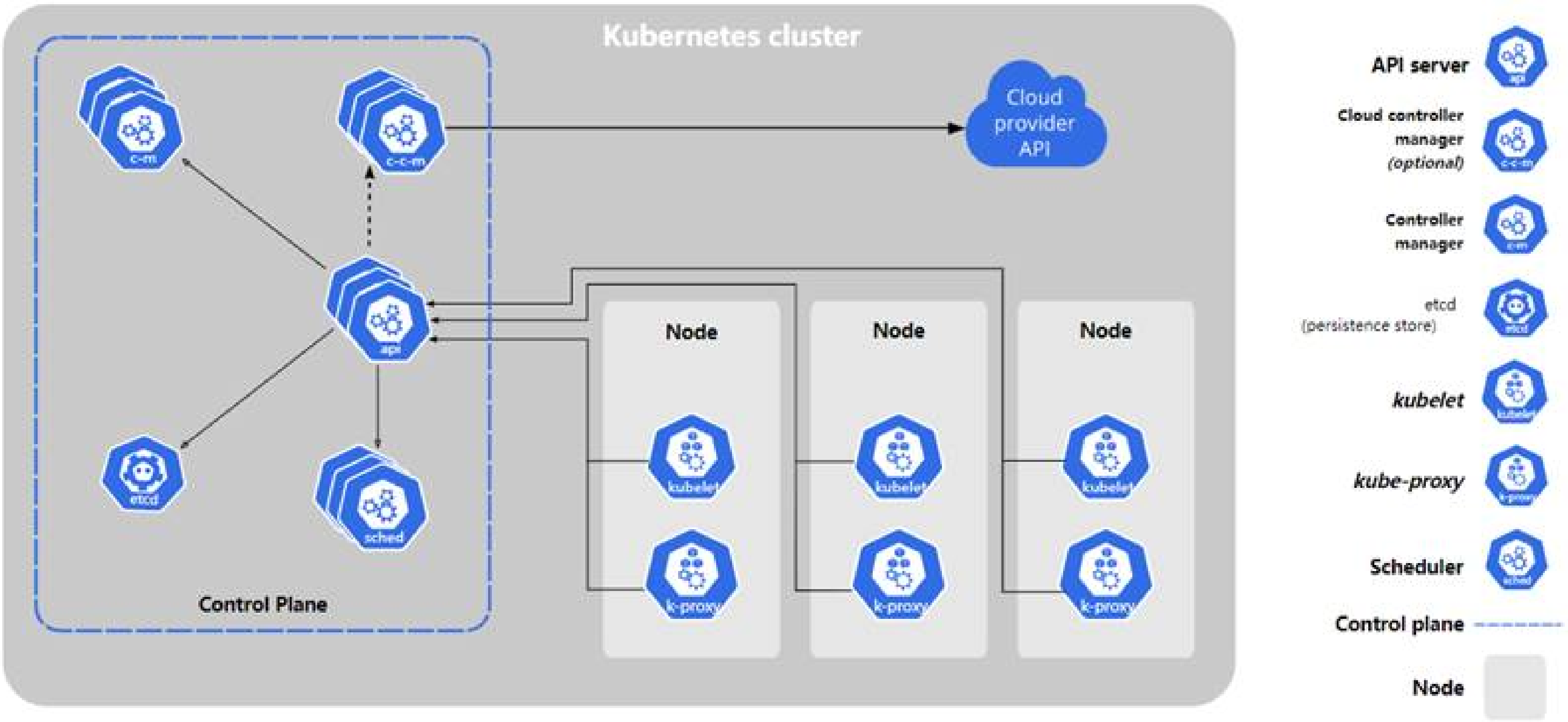
Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests.

NEW QUESTION 19
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

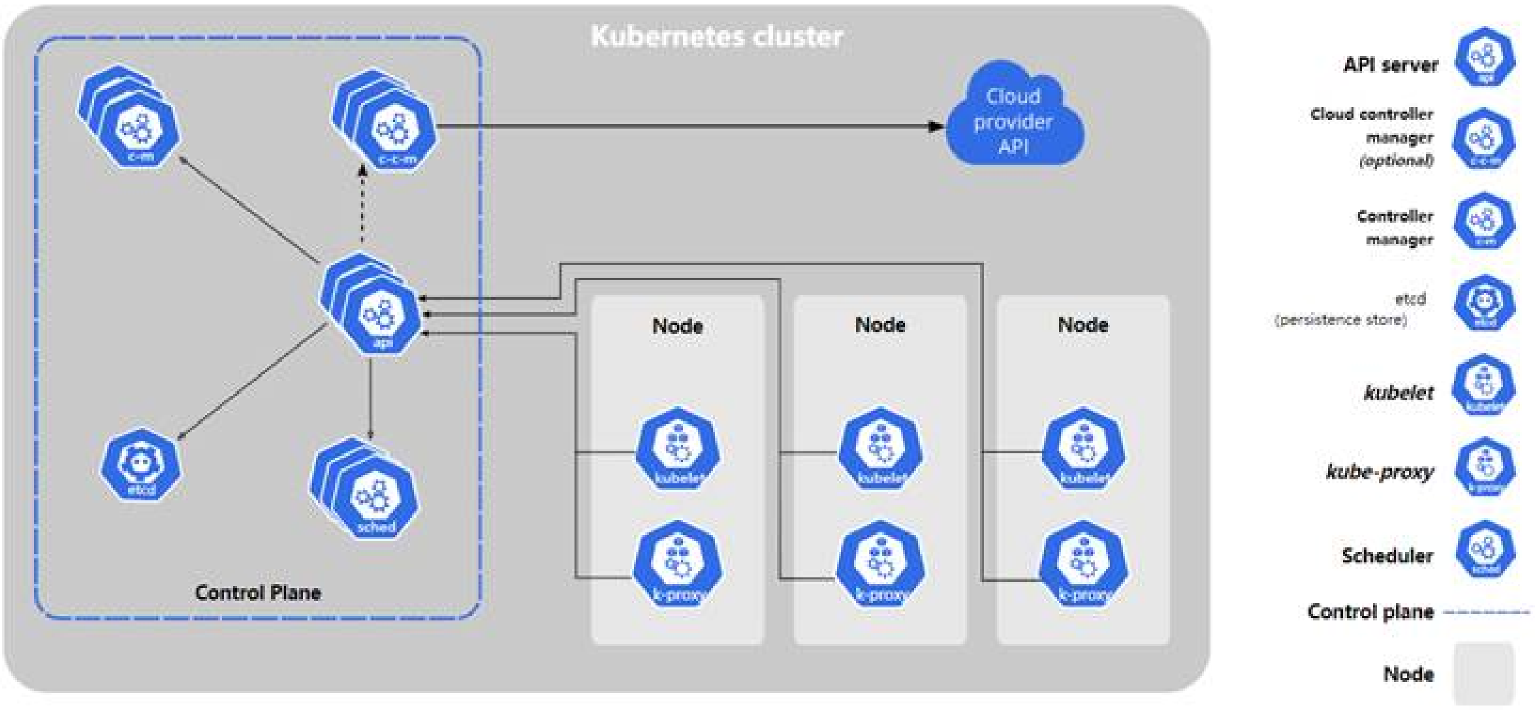


NEW QUESTION 23
Which of the following is not the part of Kubernetes Control Plane?

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

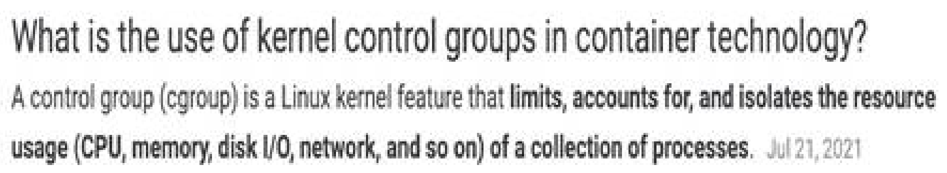


NEW QUESTION 27
What do control groups provide when it come to containers

A. Permission
B. Image Storage
C. Isolation
D. Logging

Answer: C

Explanation:
Text Description automatically generated



What is the use of kernel control groups in container technology?

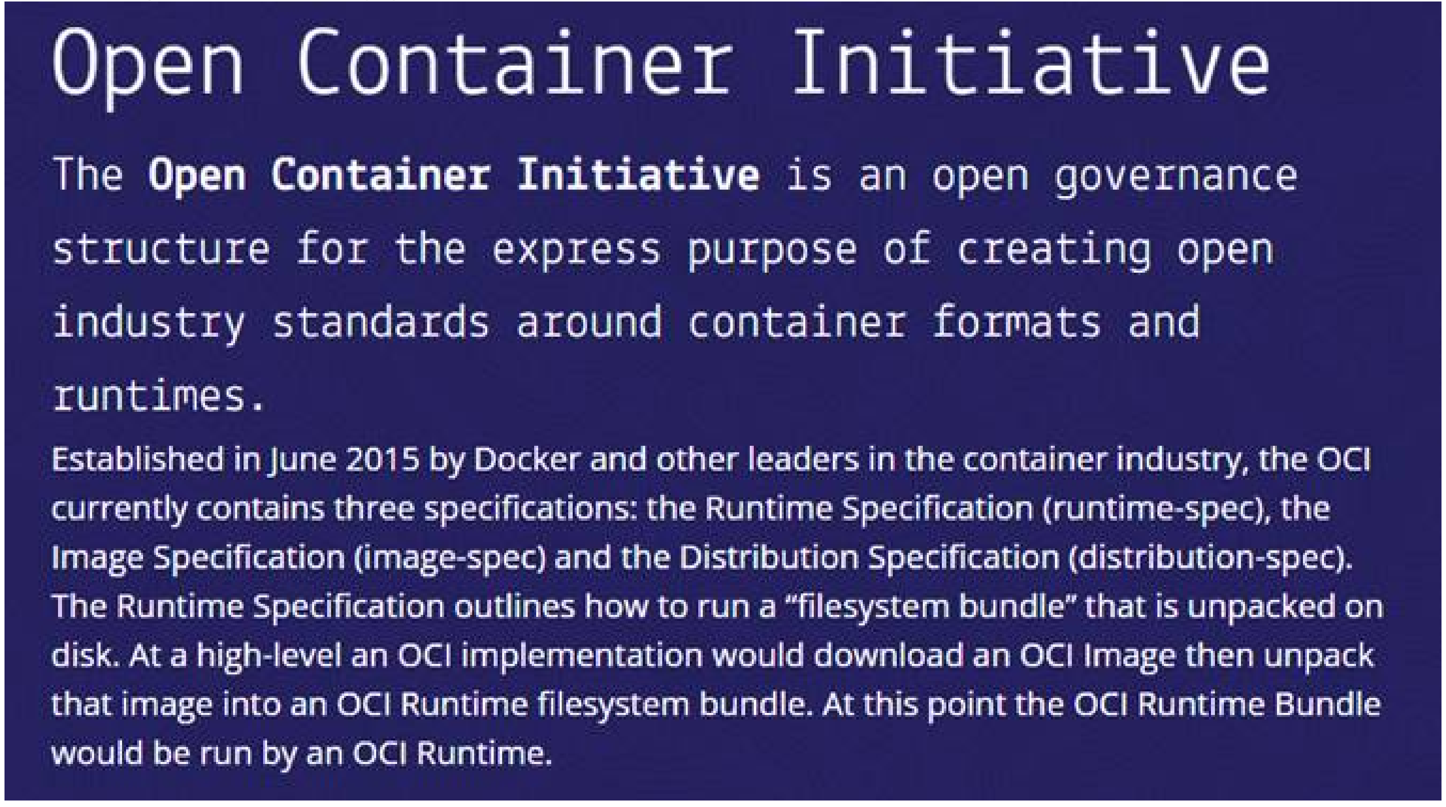
A control group (cgroup) is a Linux kernel feature that **limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, and so on) of a collection of processes.** Jul 21, 2021

NEW QUESTION 31
What is Open Container Initiative 'OCI'?

A. A protocol for communicating with the kubernetes api
B. The governing body of the Cloud Native Computing Foundation 'CNCF'
C. An open standard for managing service mesh in kubernetes
D. An organization that creates open standards for containers

Answer: D

Explanation:
https://opencontainers.org/
Text Description automatically generated



Open Container Initiative

The **Open Container Initiative** is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes.

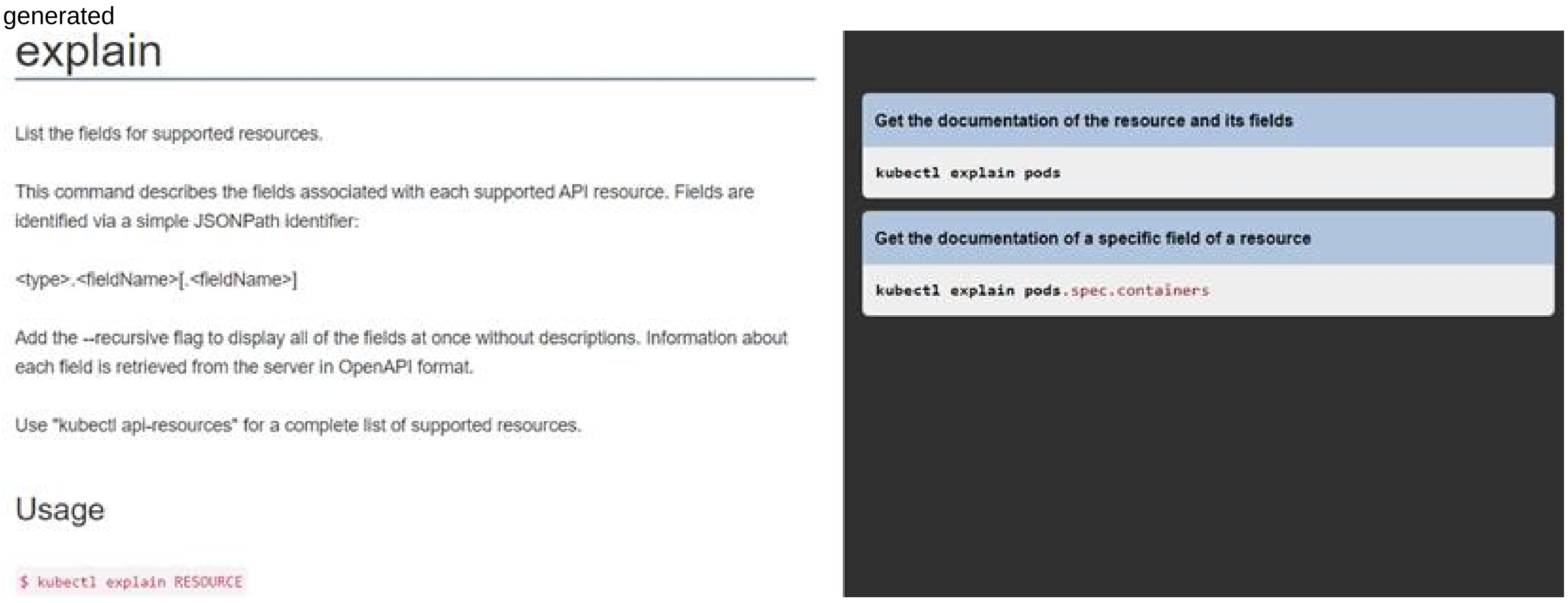
Established in June 2015 by Docker and other leaders in the container industry, the OCI currently contains three specifications: the Runtime Specification (runtime-spec), the Image Specification (image-spec) and the Distribution Specification (distribution-spec). The Runtime Specification outlines how to run a “filesystem bundle” that is unpacked on disk. At a high-level an OCI implementation would download an OCI Image then unpack that image into an OCI Runtime filesystem bundle. At this point the OCI Runtime Bundle would be run by an OCI Runtime.

NEW QUESTION 32
What command can you use to get documentation about a resource type from the command line?

A. kubectl api-resources
B. kubectl explain
C. kubectl get
D. kubeadm get-resource

Answer: B

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
```

Get the documentation of the resource and its fields

```
kubectl explain pods
```

Get the documentation of a specific field of a resource

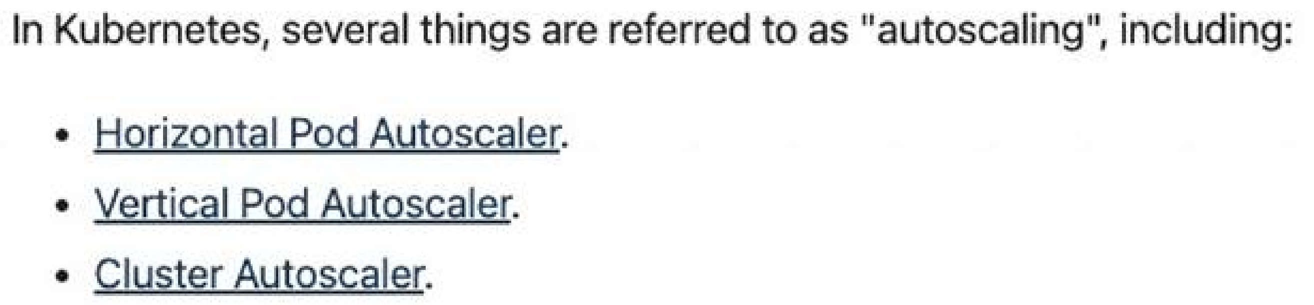
```
kubectl explain pods.spec.containers
```

NEW QUESTION 34
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 39
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 40
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

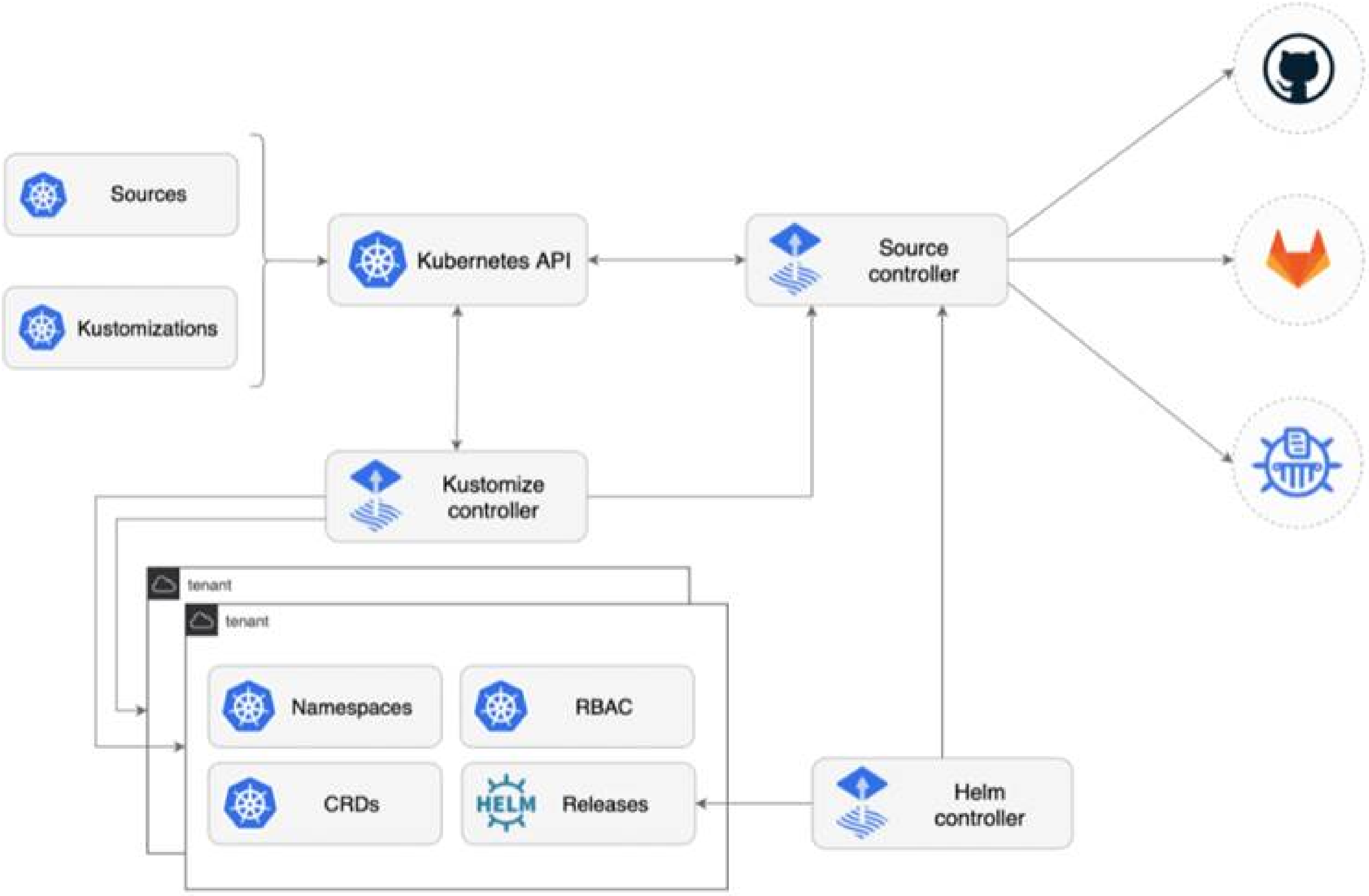
Explanation:
https://github.com/cncf/landscape#trail-map

NEW QUESTION 41
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 46
What are the two goals of Cloud-Native?

A. Rapid innovation and automation
B. Slow innovation and stable applications
C. Frequent deployments and well-defined organizational silos
D. Rapid innovation and reliability

Answer: D

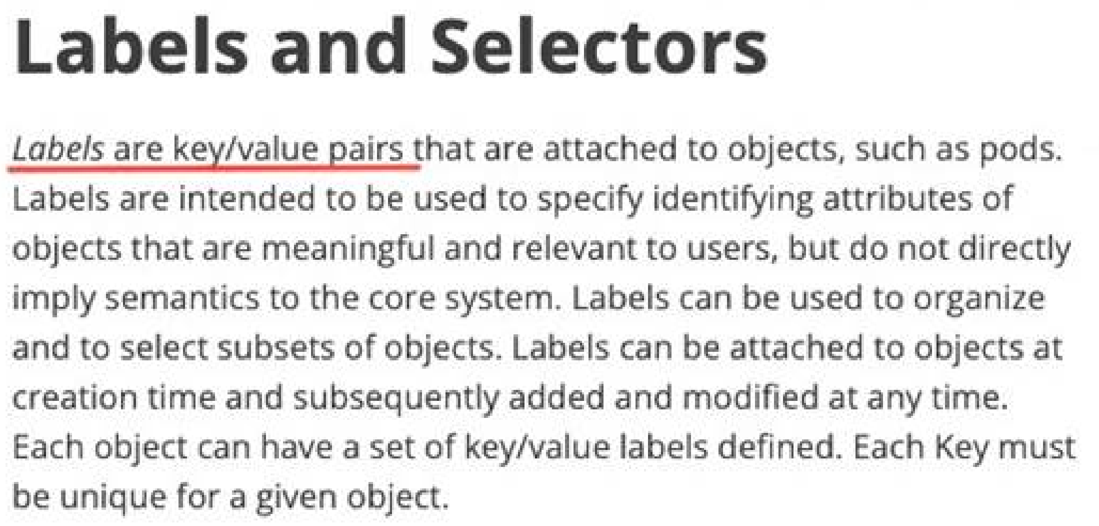
Explanation:
https://www.redhat.com/en/topics/cloud-native-apps

NEW QUESTION 50
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.

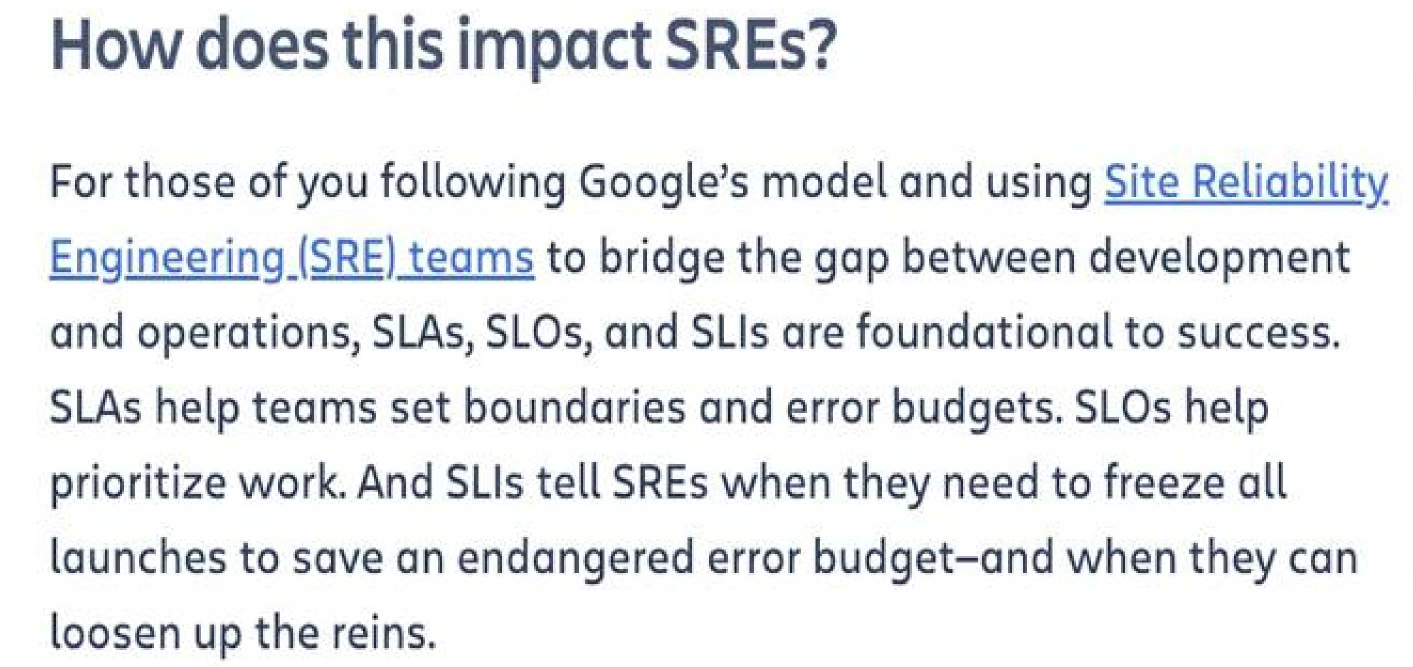
NEW QUESTION 53
Which role is responsible of creating service level indicator 'SLI', service level objective 'SLO', & Service Level Agreements 'SLA'

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?

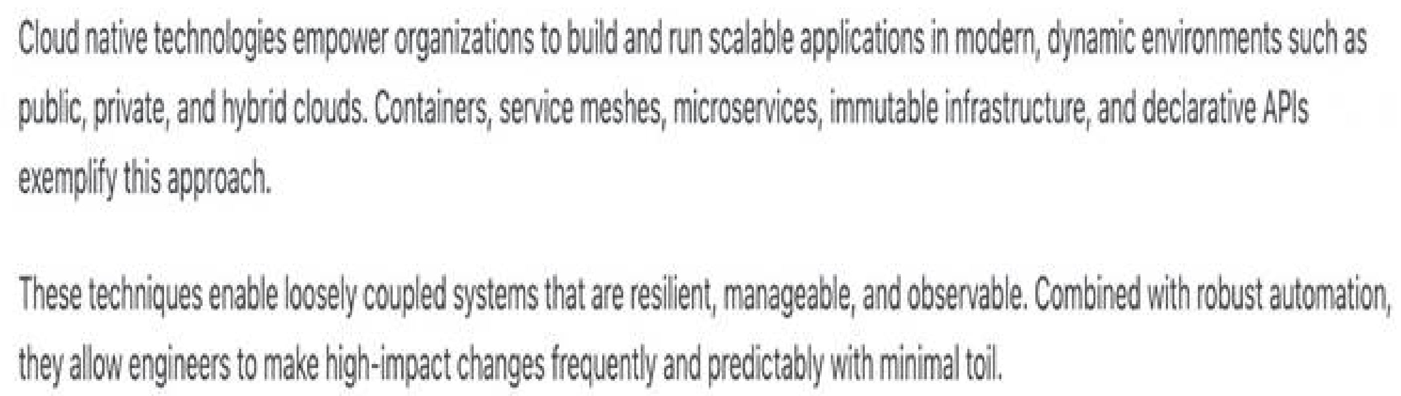
For those of you following Google's model and using Site Reliability Engineering (SRE) teams to bridge the gap between development and operations, SLAs, SLOs, and SLIs are foundational to success. SLAs help teams set boundaries and error budgets. SLOs help prioritize work. And SLIs tell SREs when they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

NEW QUESTION 56
What makes cloud native technology so important?

A. It makes data centric
B. It strengthens team
C. It removes roadblocks to innovation
D. It helps gather software requirements
E. It makes operational centric

Answer: C

Explanation:
https://github.com/cncf/foundation/blob/main/charter.md
Graphical user interface, text, application Description automatically generated



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.

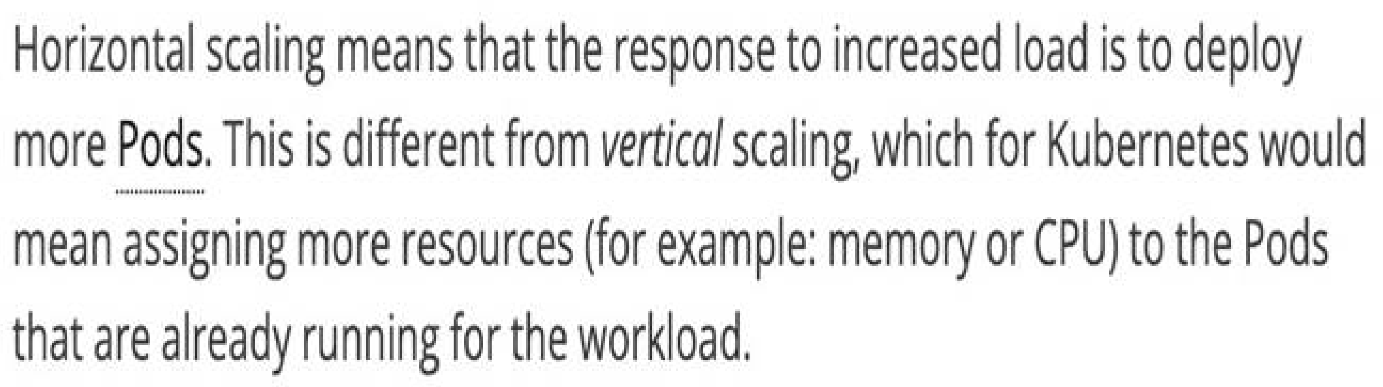
These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

NEW QUESTION 60
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 61
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

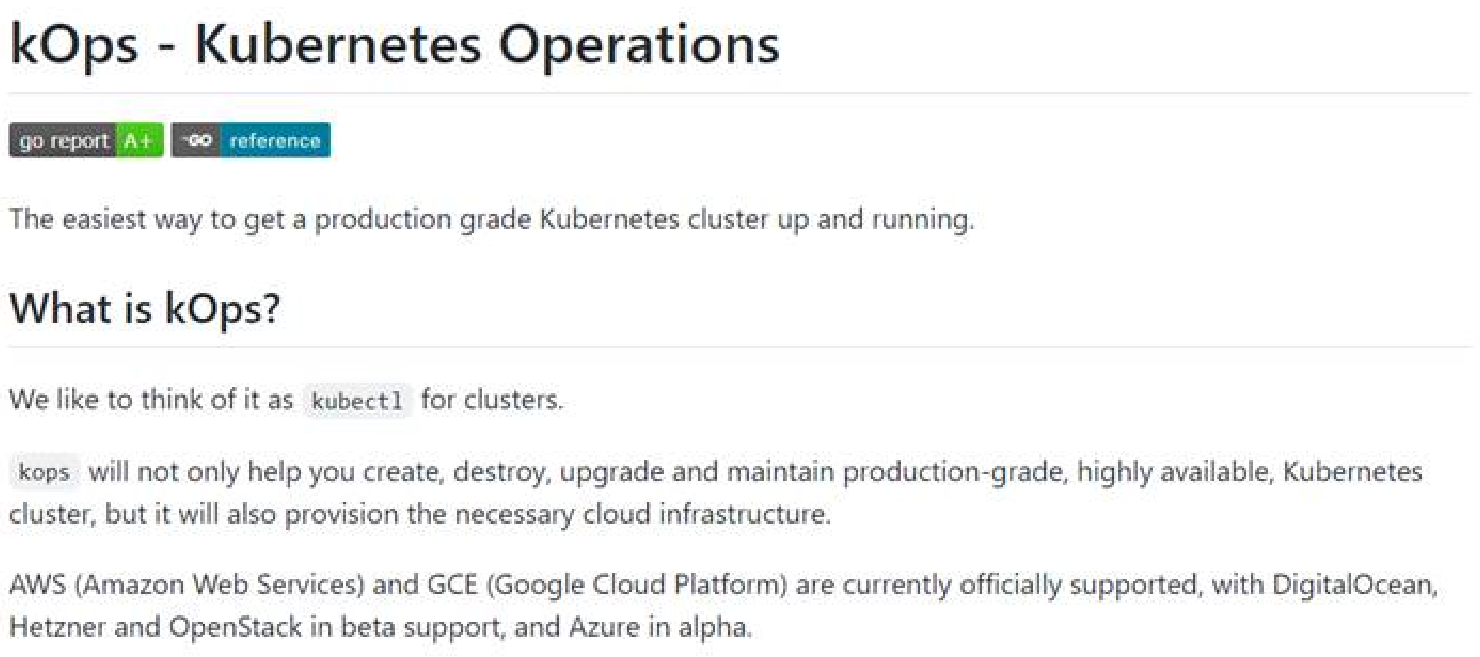
Answer: D

NEW QUESTION 65
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

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.

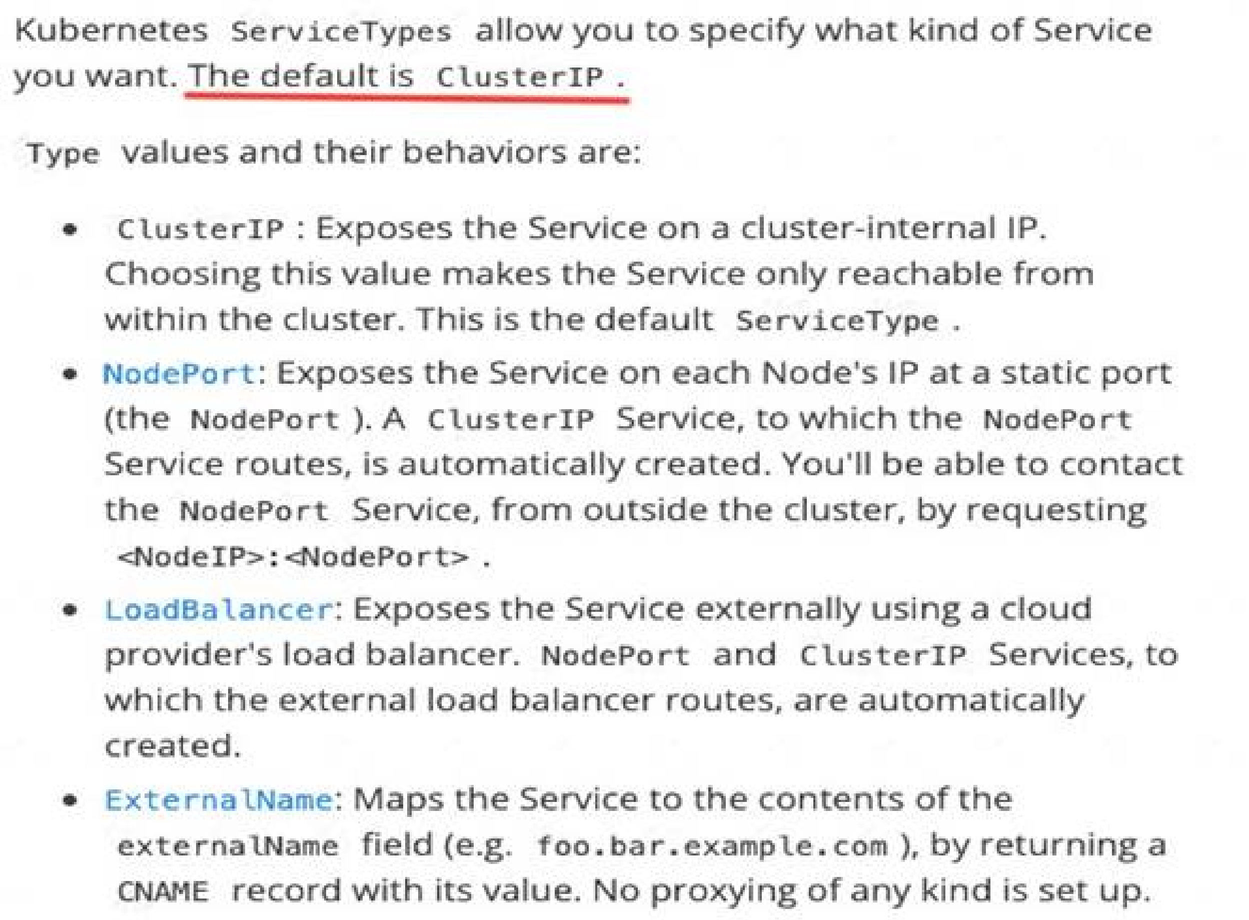
AWS (Amazon Web Services) and GCE (Google Cloud Platform) are currently officially supported, with DigitalOcean, Hetzner and OpenStack in beta support, and Azure in alpha.

NEW QUESTION 69
What is the default service type in Kubernetes?

A. CusterIP
B. NodePort
C. serviceType
D. loadBalancer

Answer: A

Explanation:
https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types Graphical user interface, text, application, email Description automatically generated



Kubernetes `ServiceTypes` allow you to specify what kind of Service you want. The default is `ClusterIP` .

`Type` values and their behaviors are:

- `ClusterIP` : Exposes the Service on a cluster-internal IP. Choosing this value makes the Service only reachable from within the cluster. This is the default `ServiceType` .
- `NodePort`: Exposes the Service on each Node's IP at a static port (the `NodePort`). A `ClusterIP` Service, to which the `NodePort` Service routes, is automatically created. You'll be able to contact the `NodePort` Service, from outside the cluster, by requesting `<NodeIP>:<NodePort>` .
- `LoadBalancer`: Exposes the Service externally using a cloud provider's load balancer. `NodePort` and `ClusterIP` Services, to which the external load balancer routes, are automatically created.
- `ExternalName`: Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a `CNAME` record with its value. No proxying of any kind is set up.

NEW QUESTION 71
Which of the following is not a stop on the cloud native trailmap?

A. Microservices
B. CI/CD
C. Containerization
D. Software distribution

Answer: A

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

NEW QUESTION 75
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

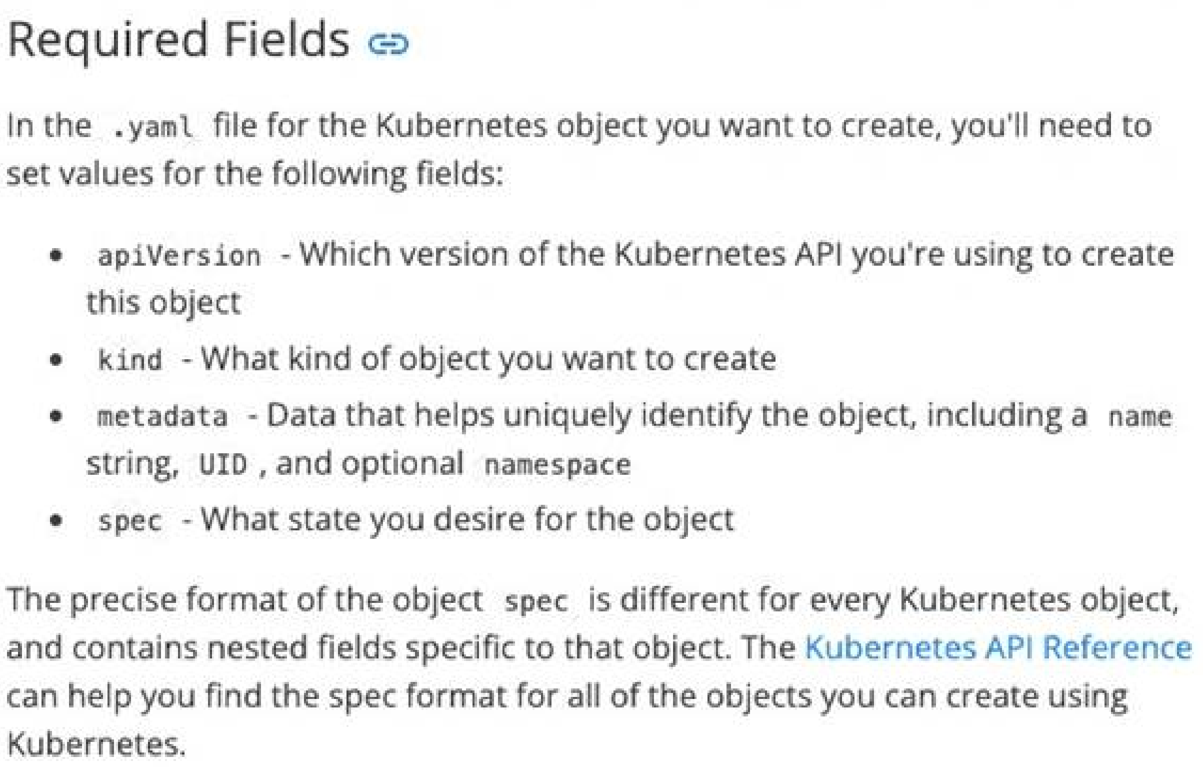
Explanation:
Scheduling takes resource requirements into account in the form of resource requests.

NEW QUESTION 76
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 78
How can you achieve cost optimization in the cloud environment?

A. Use On Demand instances
B. Use Spot Instances
C. Use Reserved Instances
D. Use Bare Metal

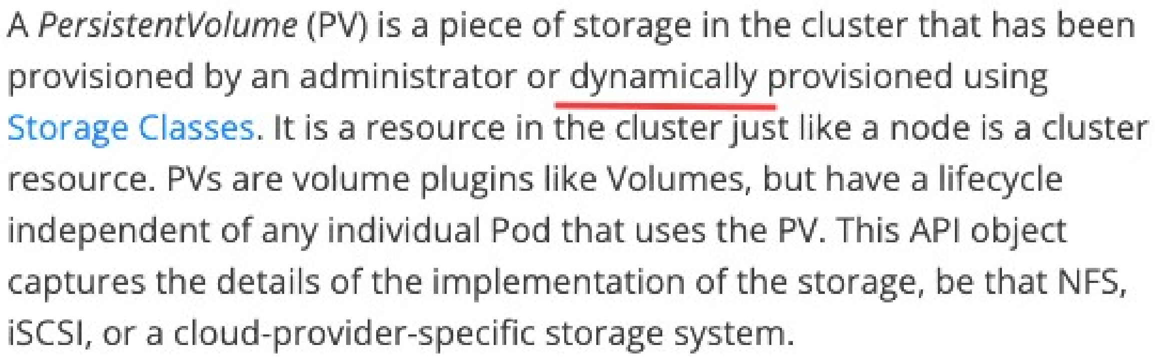
Answer: C

NEW QUESTION 81
How can persistent volume be provisioned?

A. Automatically
B. Bootstrap
C. Dynamically

Answer: C

Explanation:
https://kubernetes.io/docs/concepts/storage/persistent-volumes/ Text Description automatically generated



A *PersistentVolume* (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes, but have a lifecycle independent of any individual Pod that uses the PV. This API object captures the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system.

NEW QUESTION 83
Observability and monitoring are not the same?

A. True
B. False

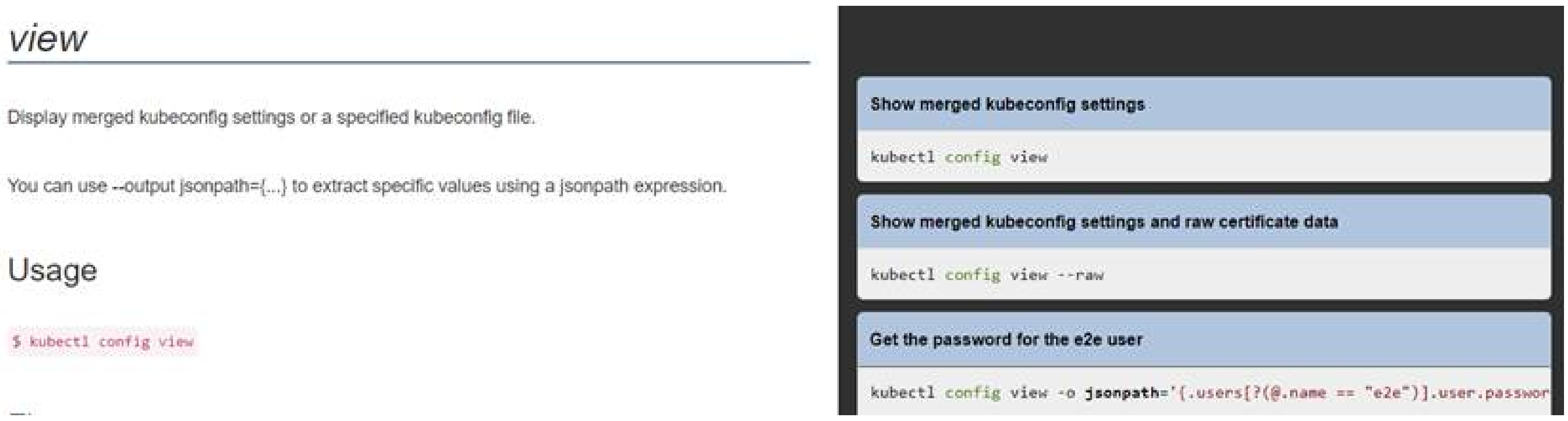
Answer: A

NEW QUESTION 86
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
```

Show merged kubeconfig settings

```
kubectl config view
```

Show merged kubeconfig settings and raw certificate data

```
kubectl config view --raw
```

Get the password for the e2e user

```
kubectl config view -o jsonpath='{.users[?(@.name == "e2e")].user.passwor
```

NEW QUESTION 89
What is autoscaling?

A. Automatically measuring resource usage
B. Automatically assigning workloads to nodes in a cluster
C. Automatically repairing broken application instances
D. Automatically adding or removing compute resources as needed

Answer: D

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

NEW QUESTION 90
Continuous delivery is .

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

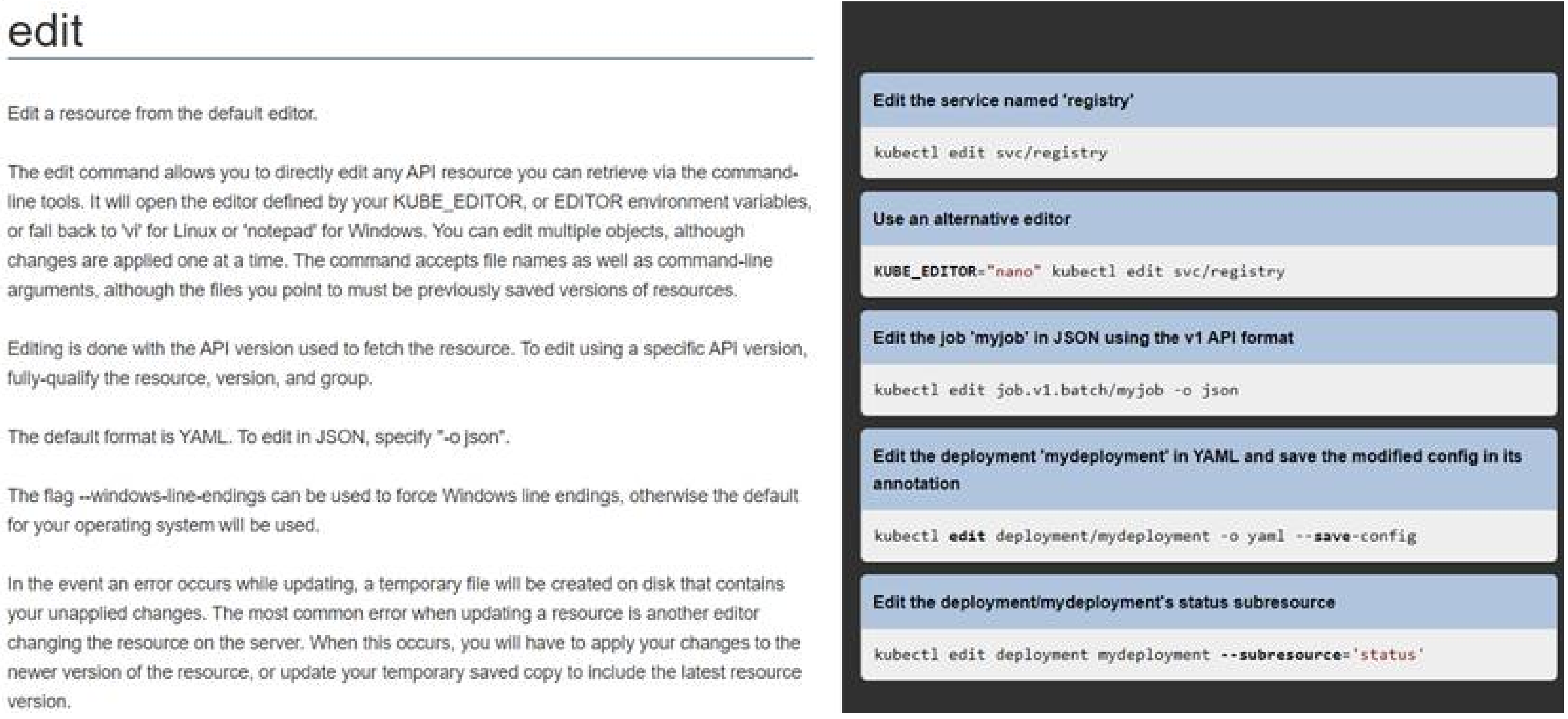
Answer: C

NEW QUESTION 91
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.

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

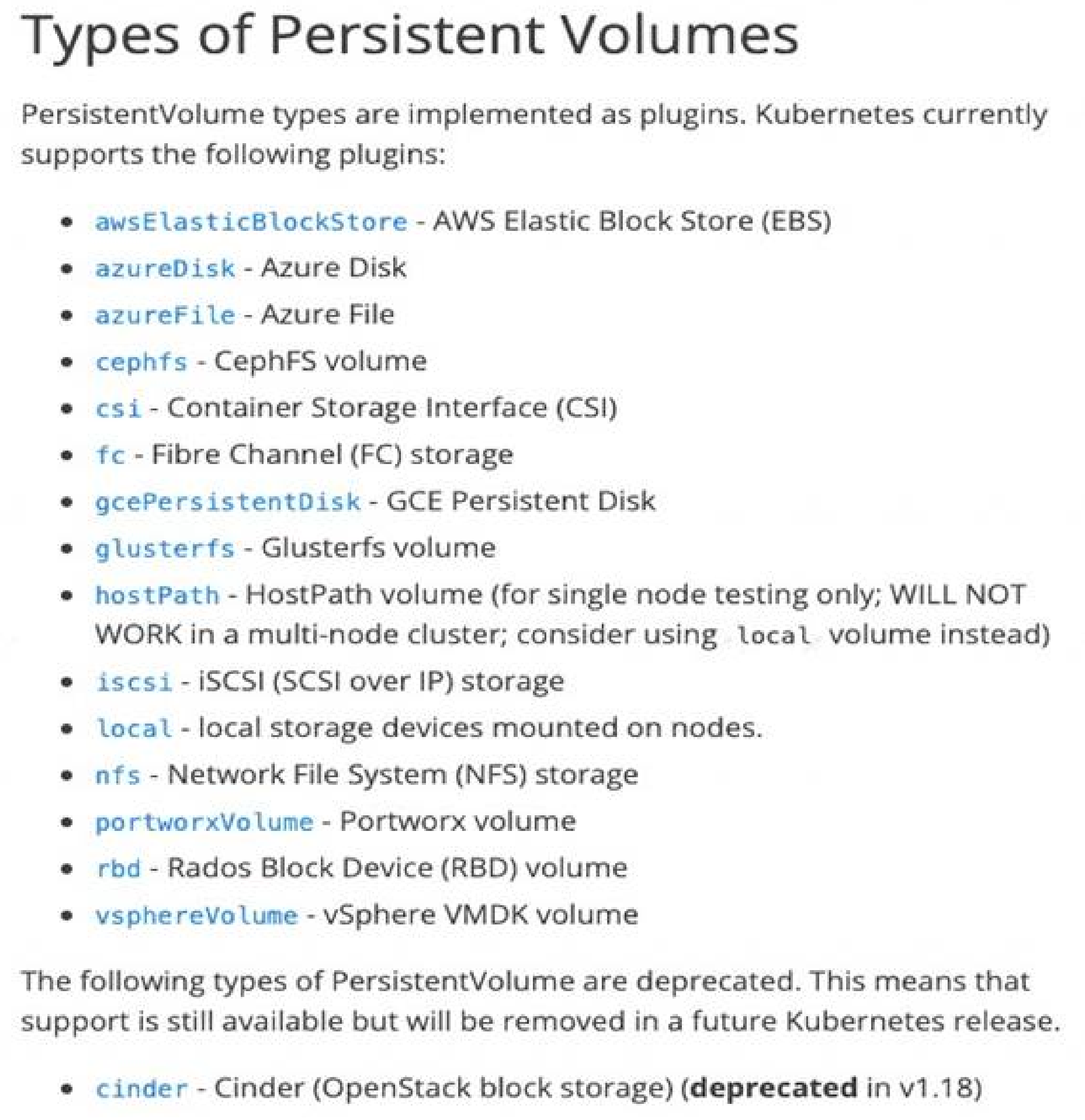
```
kubectl edit deployment mydeployment --subresource='status'
```

NEW QUESTION 93
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.

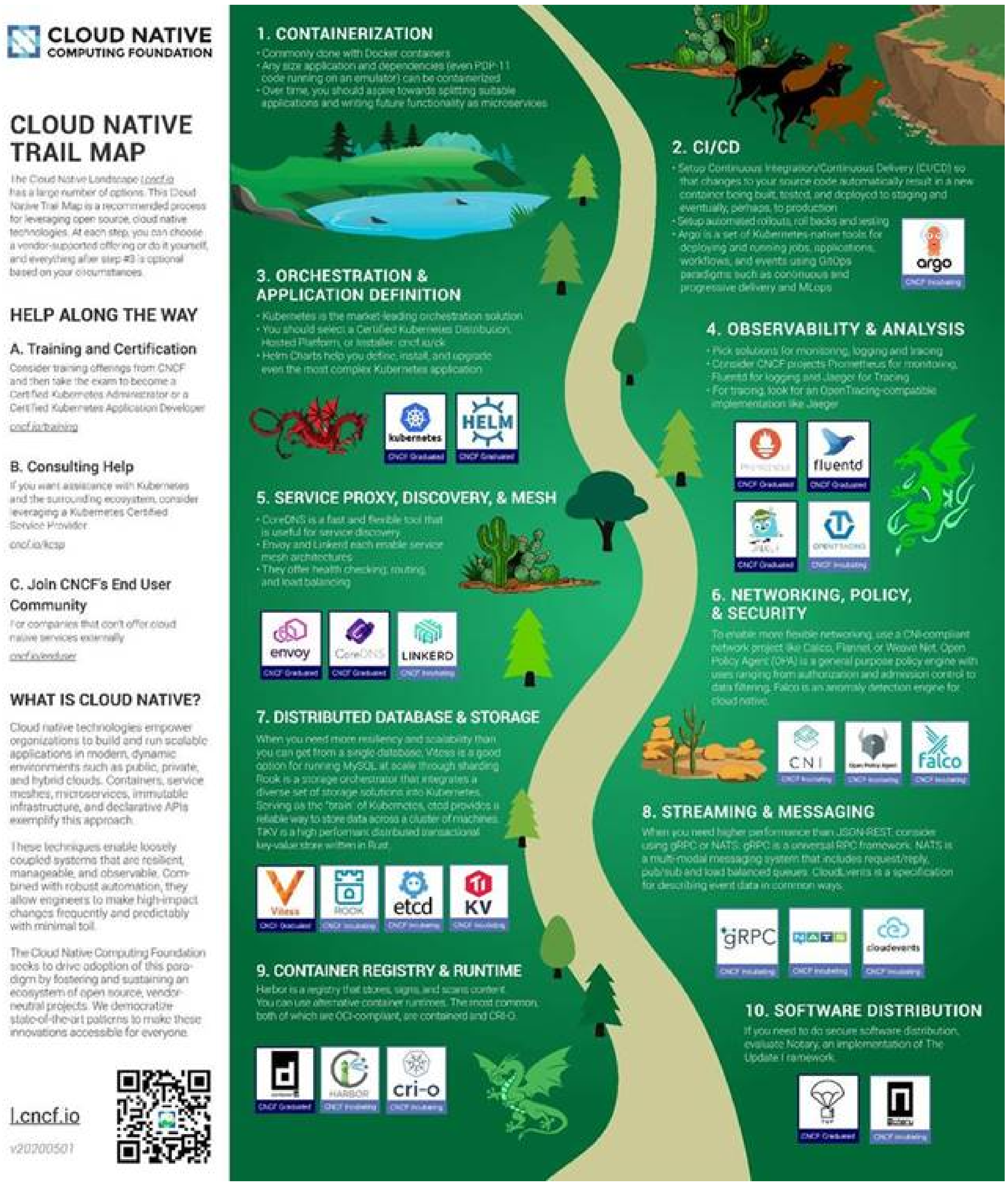
- `cinder` - Cinder (OpenStack block storage) (**deprecated** in v1.18)

NEW QUESTION 96
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 COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer

cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. 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

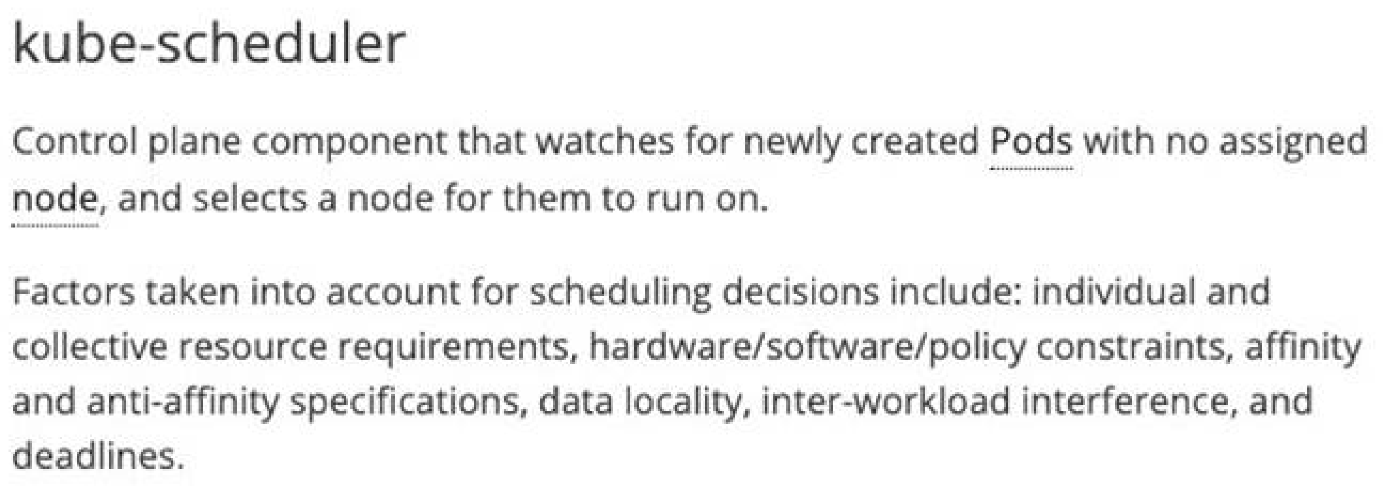
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

NEW QUESTION 98
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.

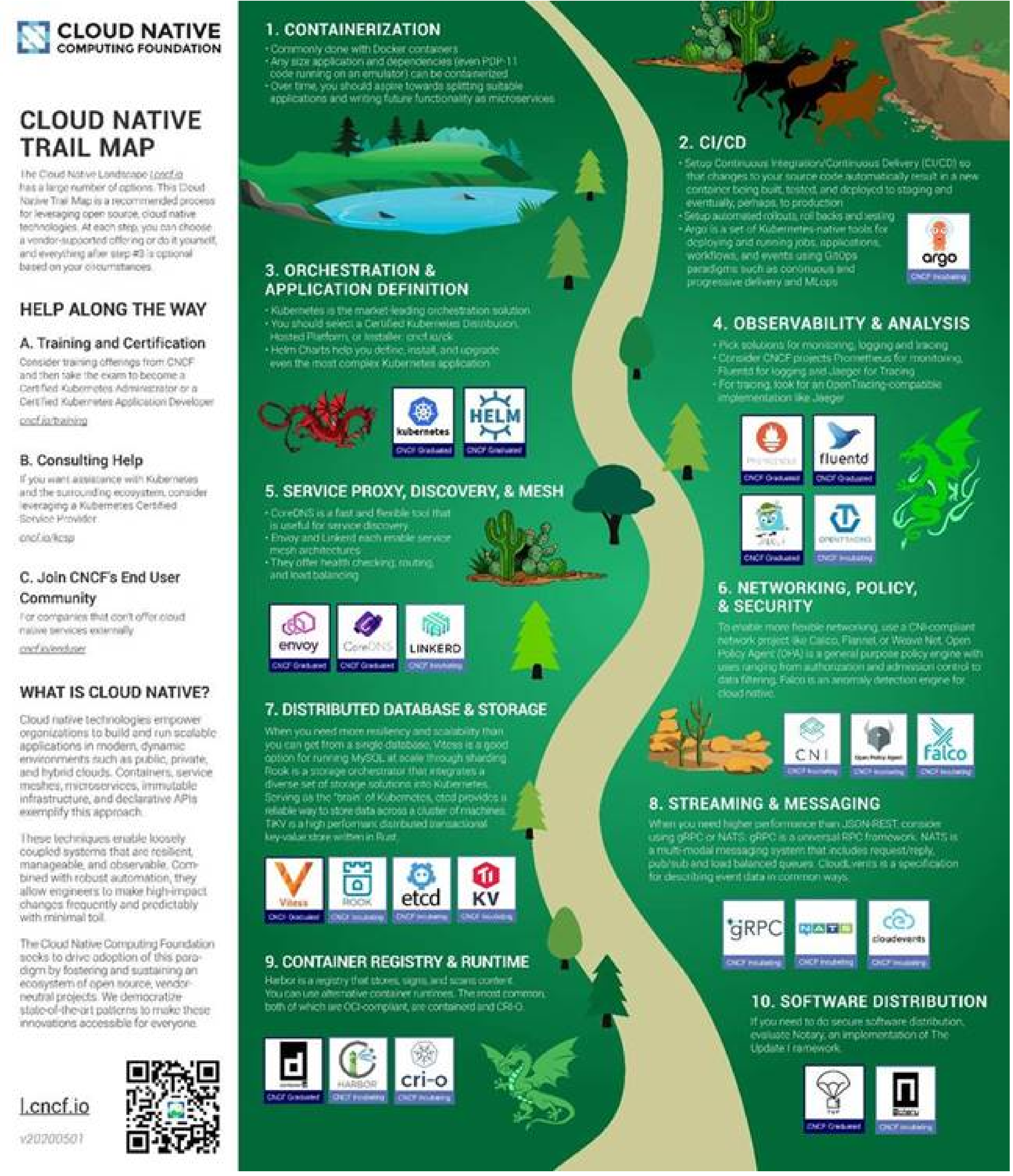
Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

NEW QUESTION 99
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
COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification
Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer
cncf.io/training

B. Consulting Help
If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider
cncf.io/kcsp

C. Join CNCF's End User Community
For companies that don't offer cloud native services externally
cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501

1. CONTAINERIZATION
- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD
- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

3. ORCHESTRATION & APPLICATION DEFINITION
- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS
- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH
- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY
To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE
When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING
When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. 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.

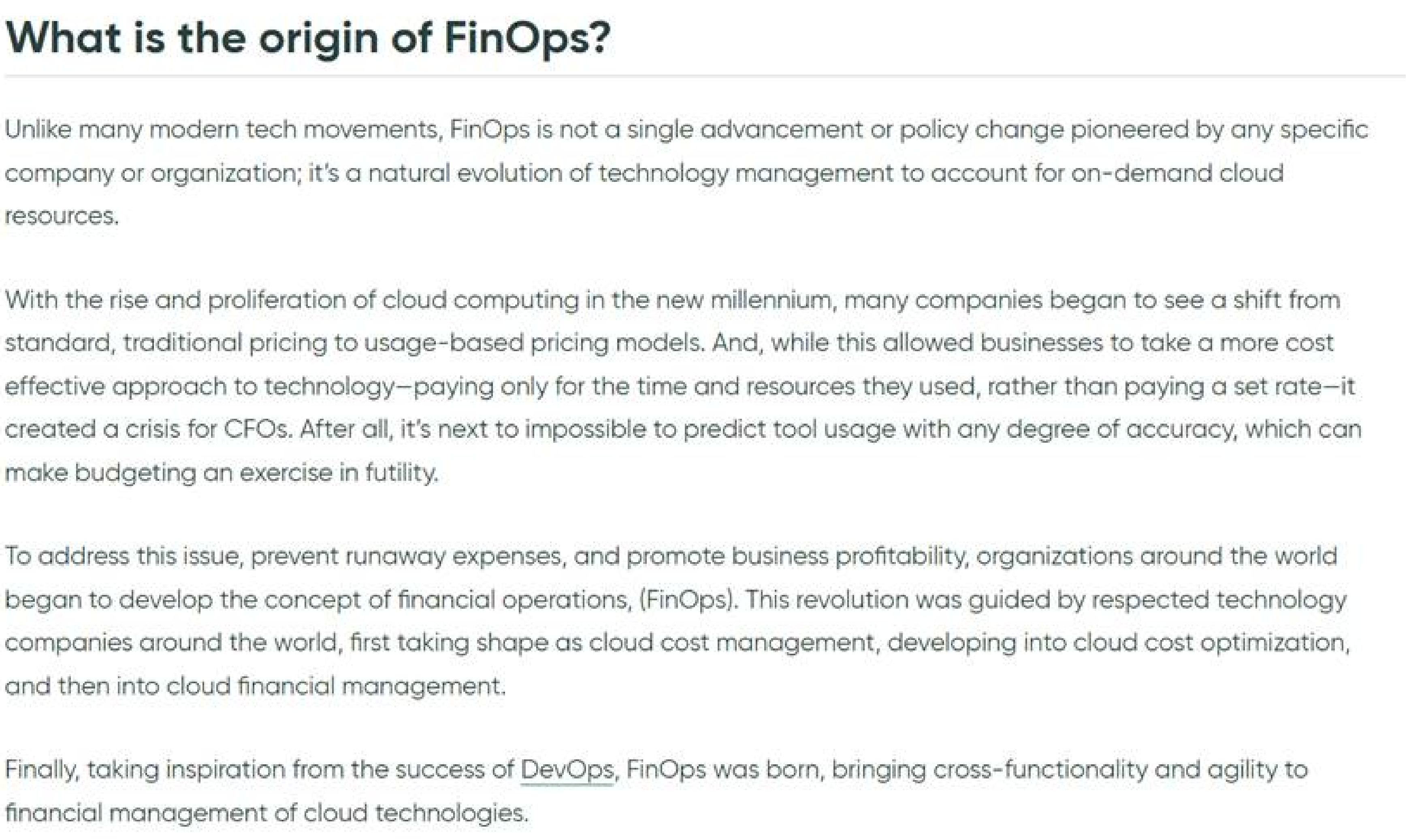
10. SOFTWARE DISTRIBUTION
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

NEW QUESTION 104
What is FinOps?

A. The first step in any cloud transformation
B. Stage beyond DevOps or DevSecOps, where organization transition to serverless tech-nologies
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.

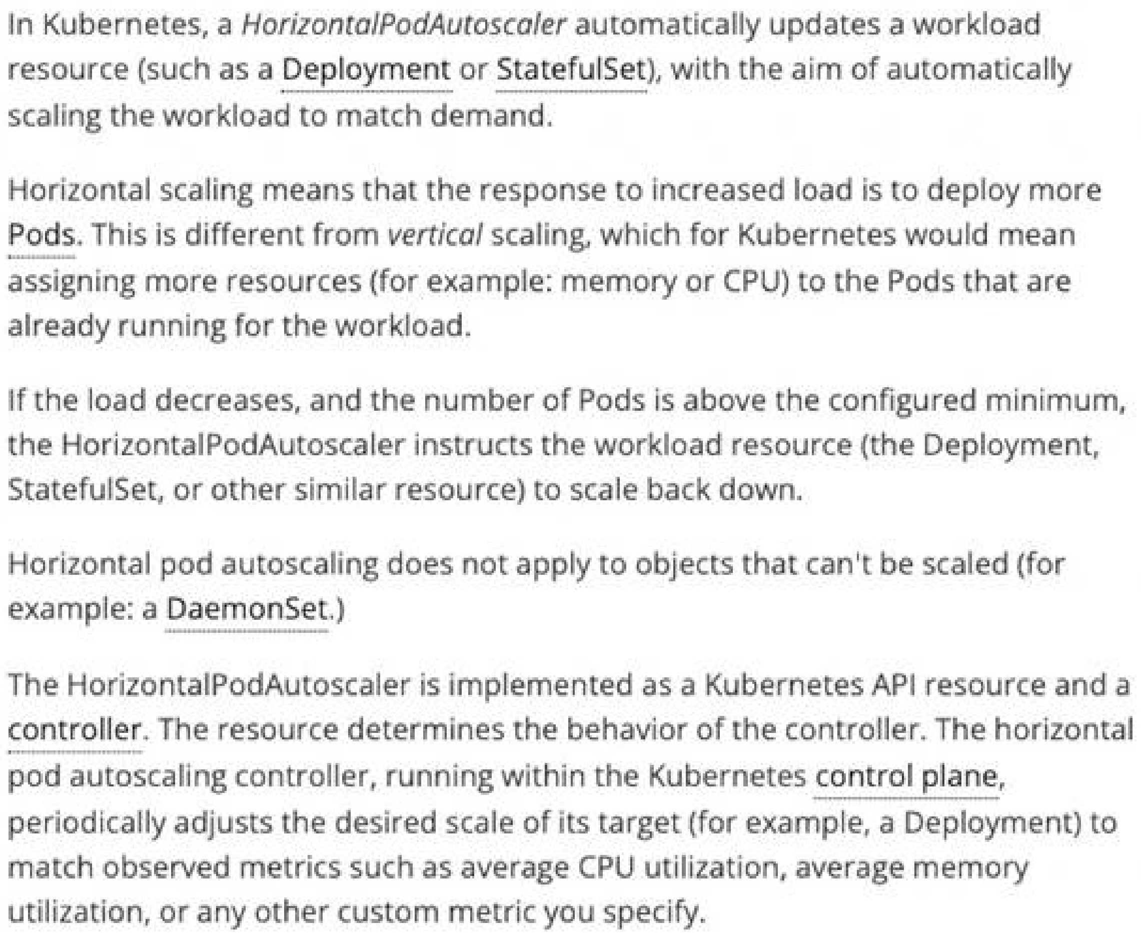
Finally, taking inspiration from the success of DevOps, FinOps was born, bringing cross-functionality and agility to financial management of cloud technologies.

NEW QUESTION 107
What is horizontal scaling?

A. Creating a Deployment
B. Adding resources to existing apps and servers
C. Moving workloads from one server to another
D. Adding additional replicas of apps and servers

Answer: D

Explanation:
https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/ Text, letter Description automatically generated



In Kubernetes, a *HorizontalPodAutoscaler* automatically updates a workload resource (such as a Deployment or StatefulSet), with the aim of automatically scaling the workload to match demand.

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.

If the load decreases, and the number of Pods is above the configured minimum, the HorizontalPodAutoscaler instructs the workload resource (the Deployment, StatefulSet, or other similar resource) to scale back down.

Horizontal pod autoscaling does not apply to objects that can't be scaled (for example: a DaemonSet.)

The HorizontalPodAutoscaler is implemented as a Kubernetes API resource and a controller. The resource determines the behavior of the controller. The horizontal pod autoscaling controller, running within the Kubernetes control plane, periodically adjusts the desired scale of its target (for example, a Deployment) to match observed metrics such as average CPU utilization, average memory utilization, or any other custom metric you specify.

NEW QUESTION 108
What is a commonly used package manager for kubernetes applications?

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

Answer: C

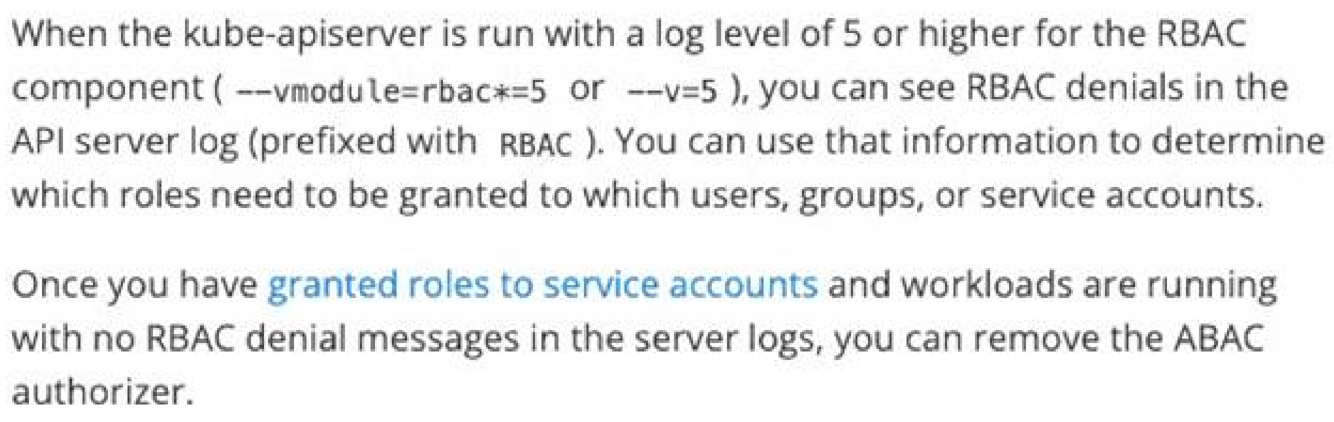
Explanation:
https://helm.sh/

NEW QUESTION 111
Which of the following best describes the way K8S Role-based access control (RBAC) works?

A. K8S does not do RBAC or Cluster role
B. RBAC lists which operations are denied to users
C. States which users can perform which actions against the resources.

Answer: C

Explanation:
https://kubernetes.io/docs/reference/access-authn-authz/rbac/ Text Description automatically generated



When the kube-apiserver is run with a log level of 5 or higher for the RBAC component (`--vmodule=rbac*=5` or `--v=5`), you can see RBAC denials in the API server log (prefixed with `RBAC`). You can use that information to determine which roles need to be granted to which users, groups, or service accounts.

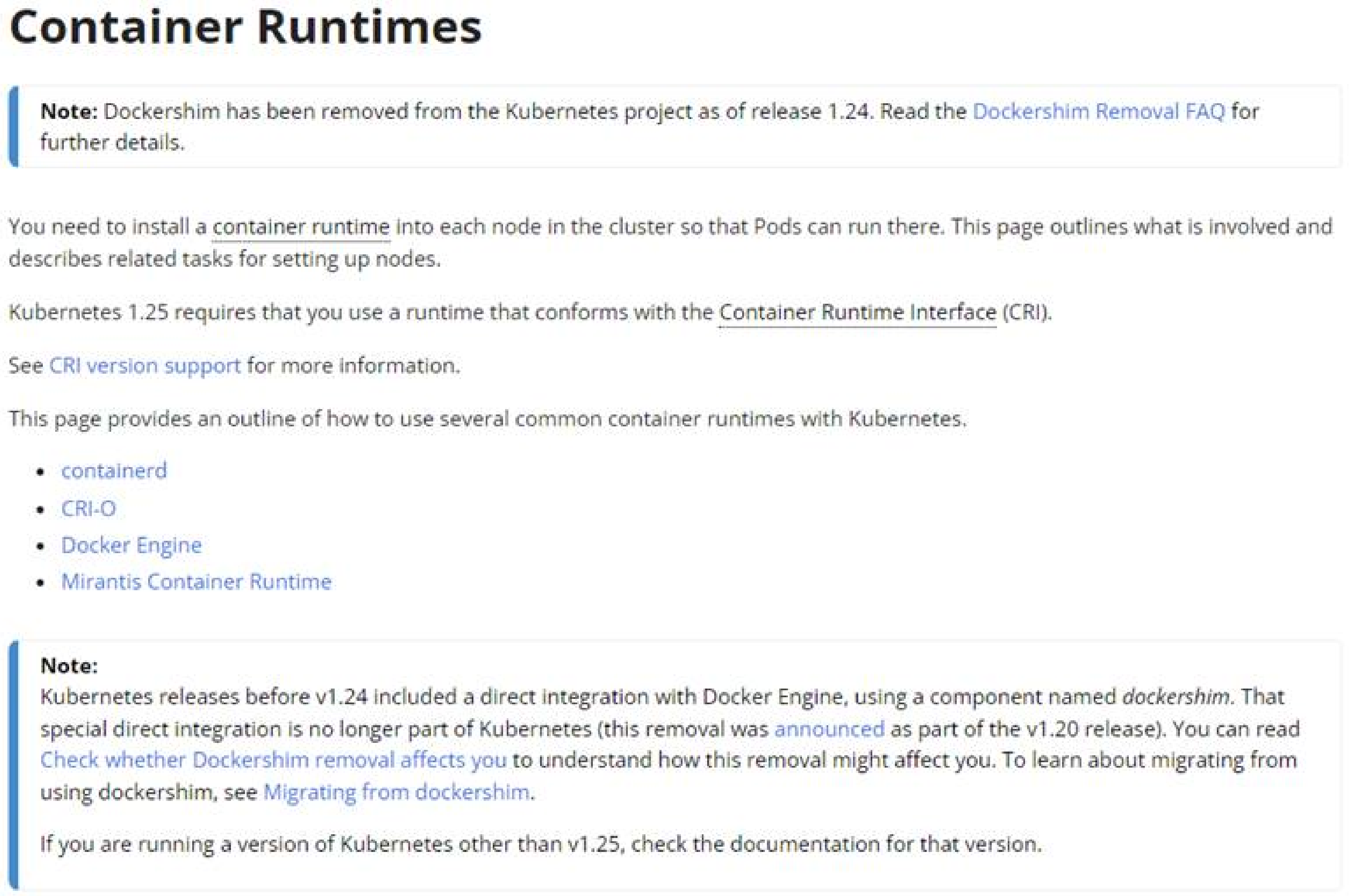
Once you have granted roles to service accounts and workloads are running with no RBAC denial messages in the server logs, you can remove the ABAC authorizer.

NEW QUESTION 114
What are container runtimes with Kubernetes?

A. CRI-O
B. lxd
C. containerd
D. Dockershim

Answer: AC

Explanation:
https://kubernetes.io/docs/setup/production-environment/container-runtimes/ Graphical user interface, text, application, email Description automatically generated



Container Runtimes

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

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

Kubernetes 1.25 requires that you use a runtime that conforms with the Container Runtime Interface (CRI).

See CRI version support for more information.

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

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

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

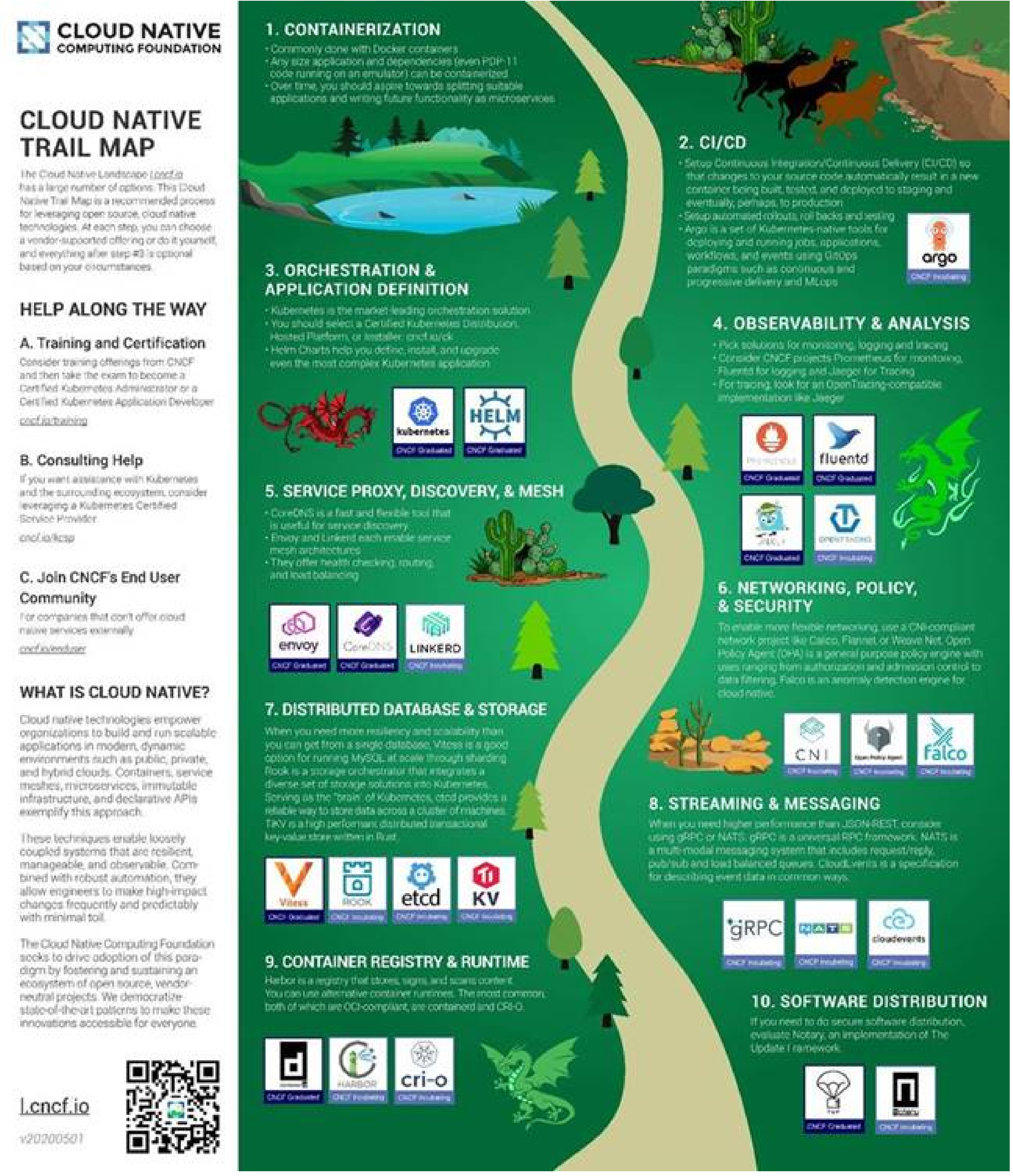
If you are running a version of Kubernetes other than v1.25, check the documentation for that version.

NEW QUESTION 116
Fluentd is the leading project in the CNCF space for logging?

A. TRUE
B. FALSE

Answer: A

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



CLOUD NATIVE
COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification
Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer
cncf.io/training

B. Consulting Help
If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider
cncf.io/kcsp

C. Join CNCF's End User Community
For companies that don't offer cloud native services externally
cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io
v20200501

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. 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

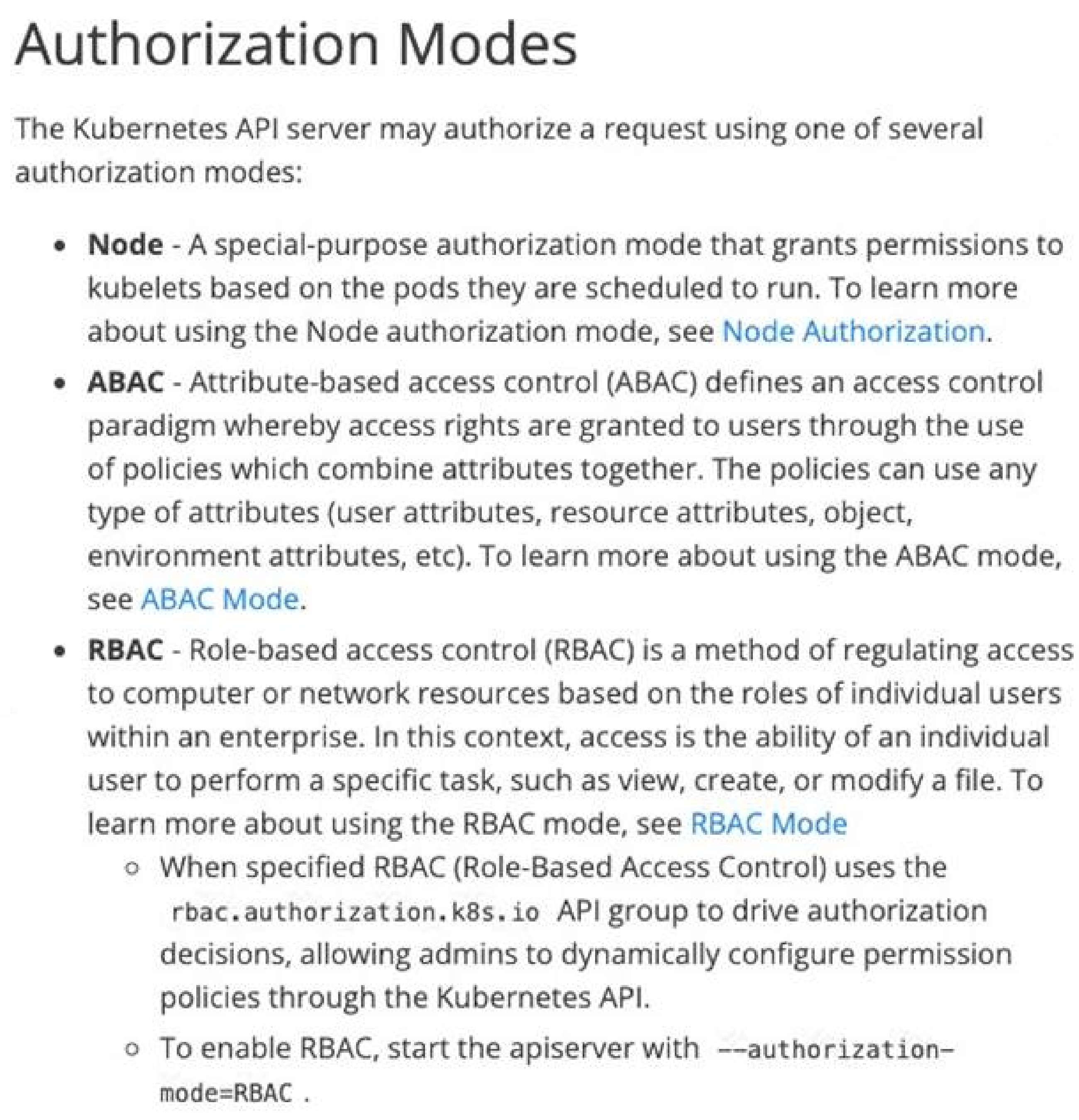
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

NEW QUESTION 121
Which access control component of Kubernetes is responsible for authorization and decides what requestor is allowed to do?

A. Service Account
B. Role-based access control 'RBAC'
C. Deployment

Answer: B

Explanation:
https://kubernetes.io/docs/reference/access-authn-authz/authorization/ Text, letter Description automatically generated



Authorization Modes

The Kubernetes API server may authorize a request using one of several authorization modes:

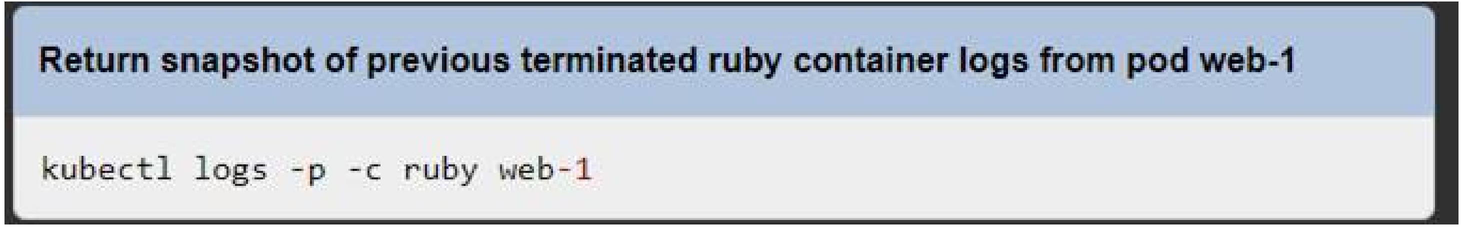
- **Node** - A special-purpose authorization mode that grants permissions to kubelets based on the pods they are scheduled to run. To learn more about using the Node authorization mode, see Node Authorization.
- **ABAC** - Attribute-based access control (ABAC) defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, object, environment attributes, etc). To learn more about using the ABAC mode, see ABAC Mode.
- **RBAC** - Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file. To learn more about using the RBAC mode, see RBAC Mode
  - When specified RBAC (Role-Based Access Control) uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing admins to dynamically configure permission policies through the Kubernetes API.
  - To enable RBAC, start the apiserver with `--authorization-mode=RBAC` .

NEW QUESTION 124
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

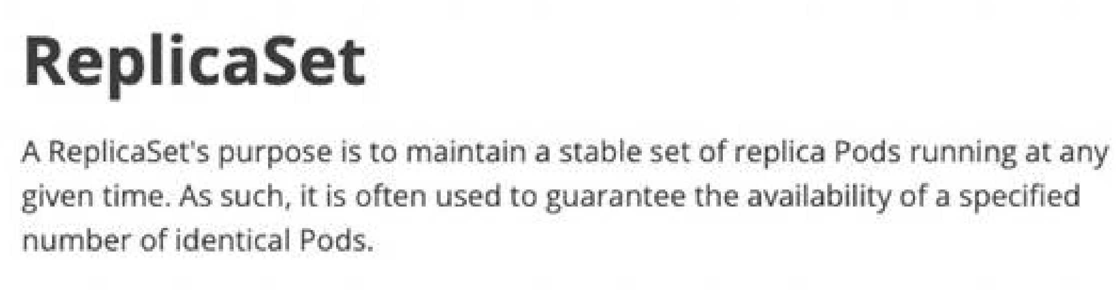
```
kubectl logs -p -c ruby web-1
```

NEW QUESTION 127
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 132
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

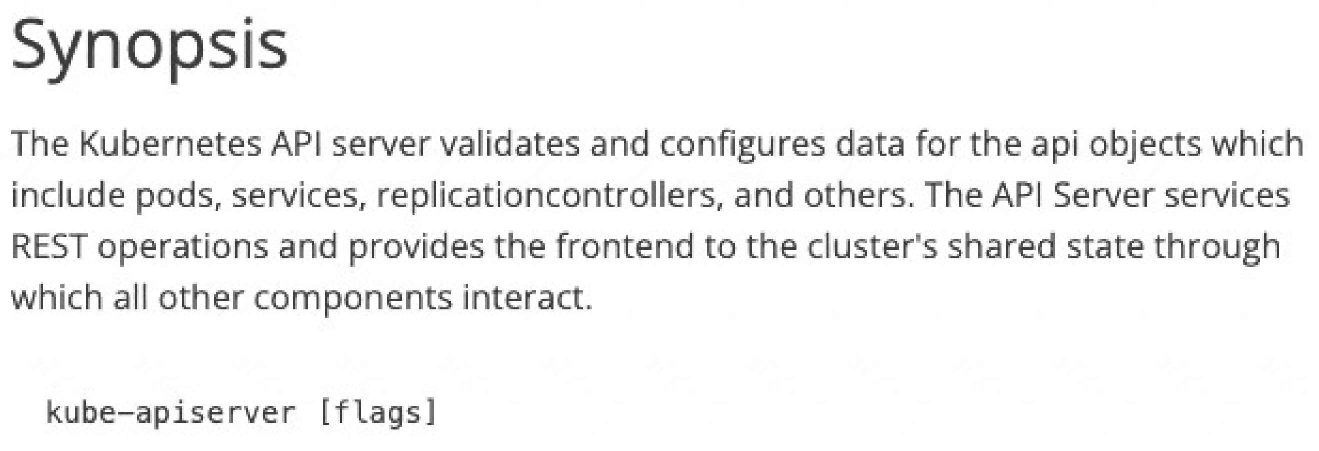
Answer: E

NEW QUESTION 136
Which component of the kubernetes control-plane (master) are all requests to deploy and manage objects posted to?

A. ETCD
B. Controller Manager
C. Kube-proxy
D. API Server
E. Kubelet

Answer: D

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



Synopsis

The Kubernetes API server validates and configures data for the api objects which include pods, services, replicationcontrollers, and others. The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.

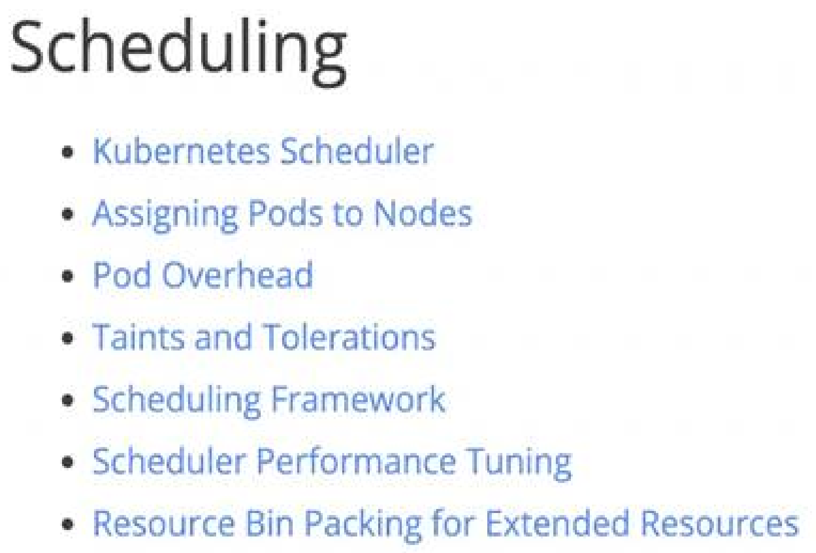
```
kube-apiserver [flags]
```

NEW QUESTION 138
What is scheduling in Kubernetes

A. Determining when to execute a cron-job
B. Assigning pods to nodes
C. Joining a new nodes to the clusters
D. Setting a time for automated tasks

Answer: B

Explanation:
https://kubernetes.io/docs/concepts/scheduling-eviction/
Graphical user interface, application Description automatically generated



Scheduling

- Kubernetes Scheduler
- Assigning Pods to Nodes
- Pod Overhead
- Taints and Tolerations
- Scheduling Framework
- Scheduler Performance Tuning
- Resource Bin Packing for Extended Resources

NEW QUESTION 140

......