Vendor: Linux Foundation

Exam Code:KCNA

Exam Name: Kubernetes and Cloud Native Associate (KCNA)

Version: Demo

QUESTION 1

How can persistent volume be provisioned?

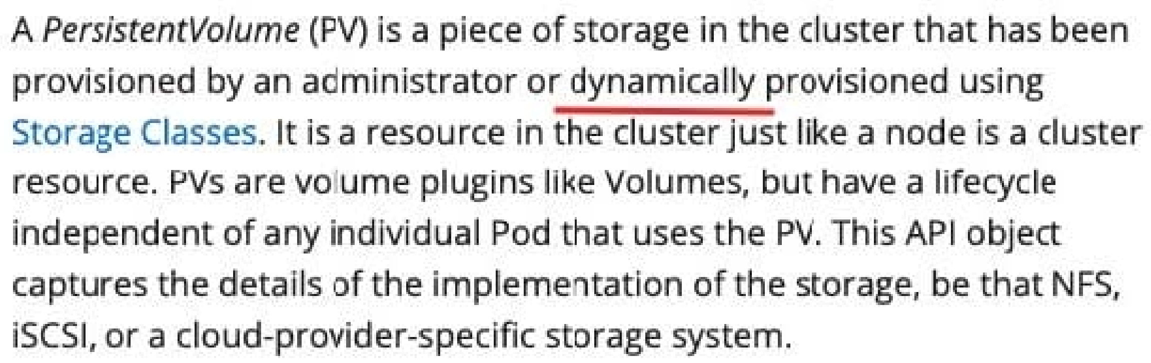
A. Automatically

B. Bootstrap

C. Dynamically

Correct Answer: C

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



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.

QUESTION 2

Which kubernetes resource type allows defining which pods are isolated when it comes to network-ing?

A. Network policy

B. Domain Name System \\'DNS\\'

C. Role Binding

D. Service

Correct Answer: A

Explanation: https://kubernetes.io/docs/concepts/services-networking/network- policies/#the-two-sorts-ofpod-isolation

The Two Sorts of Pod Isolation

There are two sorts of isolation for a pod: isolation for egress, and isolation for ingress. They concern what connections may be established. "Isolation" here is not absolute, rather it means "some restrictions apply". The alternative, "non-isolated for $direction", means that no restrictions apply in the stated direction. The two sorts of isolation (or not) are declared independently, and are both relevant for a connection from one pod to another.

By default, a pod is non-isolated for egress; all outbound connections are allowed. A pod is isolated for egress if there is any NetworkPolicy that both selects the pod and has "Egress" in its `policyTypes`; we say that such a policy applies to the pod for egress. When a pod is isolated for egress, the only allowed connections from the pod are those allowed by the `egress` list of some NetworkPolicy that applies to the pod for egress. The effects of those `egress` lists combine additively.

By default, a pod is non-isolated for ingress; all inbound connections are allowed. A pod is isolated for ingress if there is any NetworkPolicy that both selects the pod and has "Ingress" in its `policyTypes`; we say that such a policy applies to the pod for ingress. When a pod is isolated for ingress, the only allowed connections into the pod are those from the pod's node and those allowed by the `ingress` list of some NetworkPolicy that applies to the pod for ingress. The effects of those `ingress` lists combine additively.

QUESTION 3

What is not semantic versioning?

A. 1.0.0

B. 2022-05-04

C. 1.0.0-alpha

D. 1.0.0-beta.2

Correct Answer: B

Explanation: https://semver.org/RegEx SemVer at https://regex101.com/r/vkijKf/1/

QUESTION 4

What is a commonly used package manager for kubernetes applications?

A. npm

B. apt

C. helm

D. kubernetes manifest

Correct Answer: C

Explanation: https://helm.sh/

QUESTION 5

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

A. kubectl logs -p -c nginx web

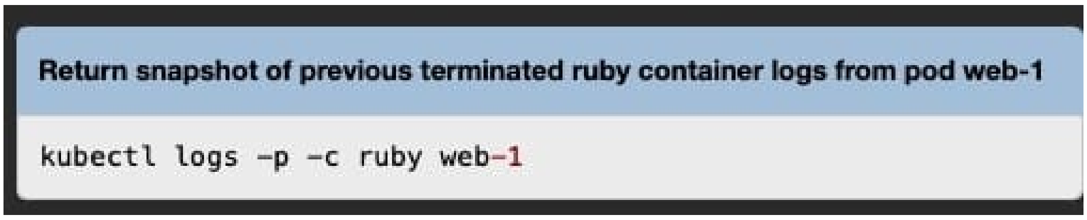
B. kubectl logs nginx

C. kubectl logs -p -c web nginx

D. kubectl logs -f -c nginx web

Correct Answer: A

Explanation: https://kubernetes.io/docs/reference/generated/kubectl/kubectl- commands#logs


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

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

QUESTION 6

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

Correct Answer: D

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

QUESTION 7

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

Correct Answer: B

Explanation: https://www.splunk.com/en_us/data-insider/what-is-distributed-tracing.html

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.

Text, letter

QUESTION 8

What is the command used to scale the application?

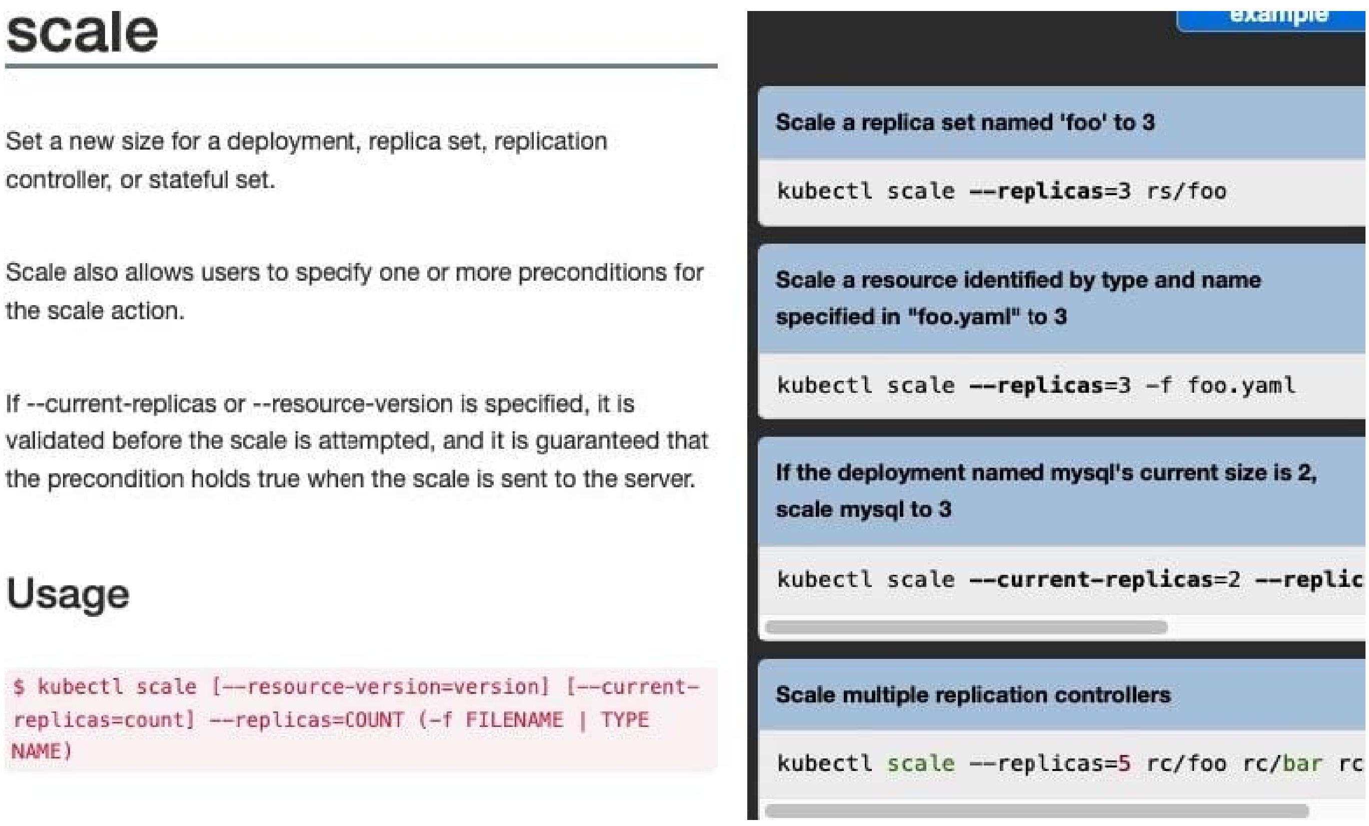
A. kubectl run

B. kubectl explain

C. kubectl scale

Correct Answer: C

Explanation: https://kubernetes.io/docs/reference/generated/kubectl/kubectl- commands#scale



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 --replic
```

Scale multiple replication controllers

```
kubectl scale --replicas=5 rc/foo rc/bar rc
```

QUESTION 9

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

A. start container

B. CustomResourceDefinitions

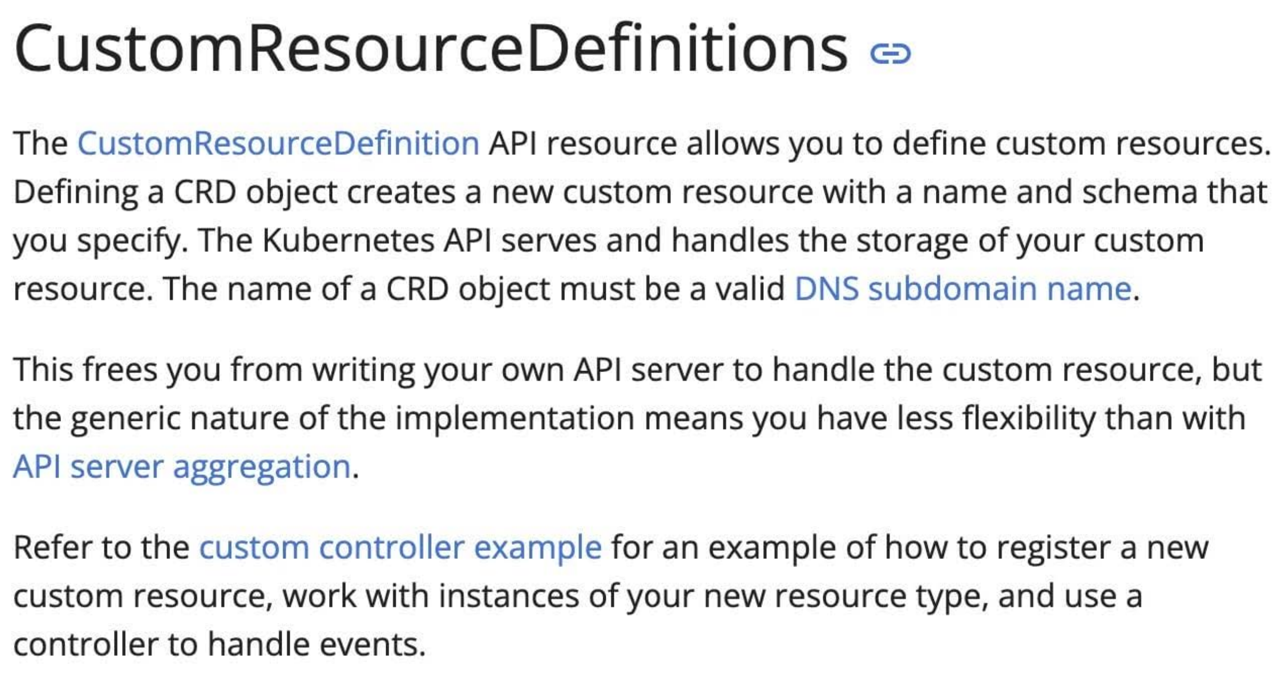
C. init container

D. Flux

E. CRI-O

Correct Answer: B

Explanation: https://kubernetes.io/docs/concepts/extend-kubernetes/api- extension/custom-resources/



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.

QUESTION 10

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

Correct Answer: ABC

QUESTION 11

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

Correct Answer: D

QUESTION 12

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

Correct Answer: D