

DUMPS ARENA

Kubernetes and Cloud Native Associate (KCNA)

Linux Foundation KCNA

Version Demo

Total Demo Questions: 9

Total Premium Questions: 126

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

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

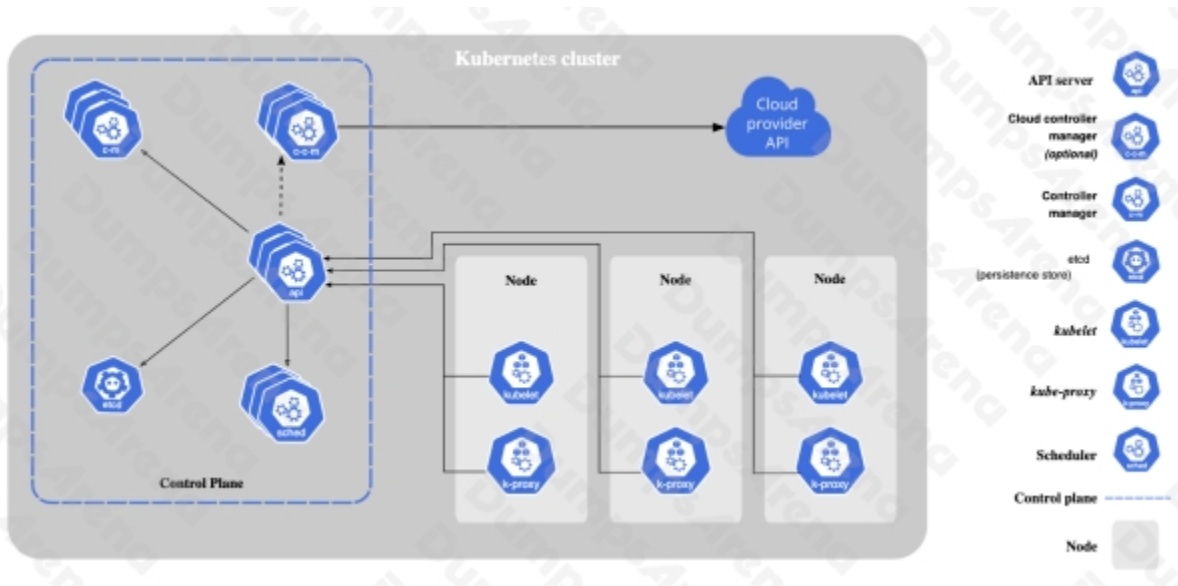
ANSWER: A B C E**QUESTION NO: 2**

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/>



QUESTION NO: 3

What are container runtimes with Kubernetes?

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

ANSWER: A C

Explanation:

Container Runtimes

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

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

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

See [CRI version support](#) for more information.

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

- [containerd](#)
- [CRI-O](#)
- [Docker Engine](#)
- [Mirantis Container Runtime](#)

Note:

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

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

QUESTION NO: 4

Various Container Orchestrator Systems (COS)?

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

ANSWER: A C D

QUESTION NO: 5

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>

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)

QUESTION NO: 6

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>

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`

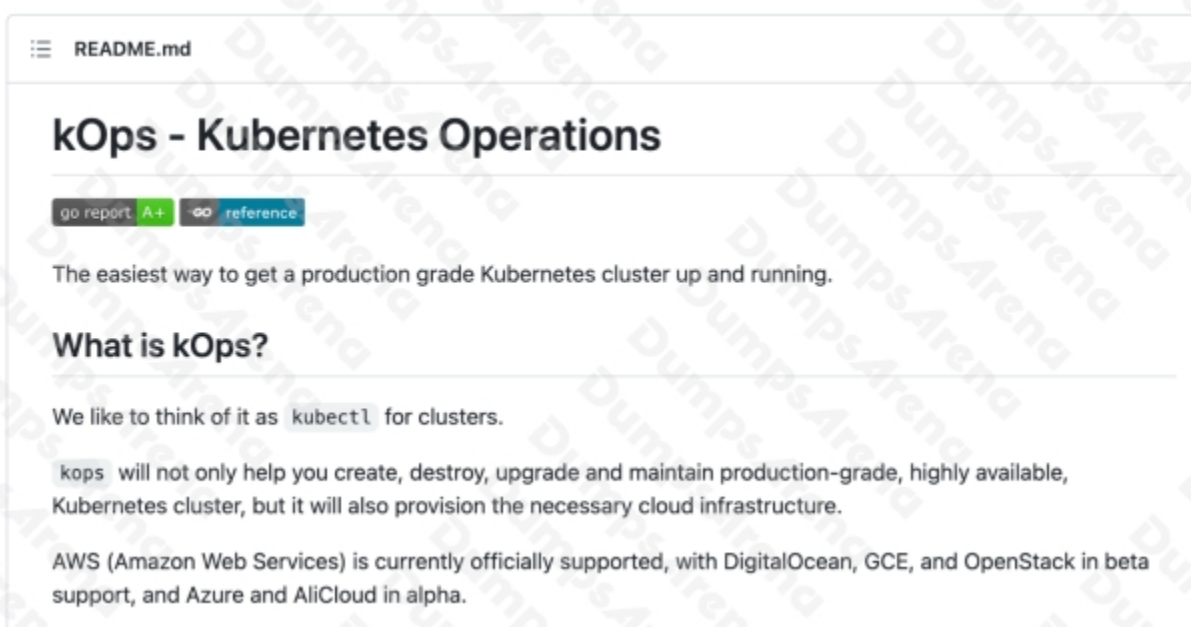
Return snapshot of previous terminated ruby container logs from pod web-1
`kubectl logs -p -c ruby web-1`

Begin streaming the logs of the ruby container in pod web-1
`kubectl logs -f -c ruby web-1`

QUESTION NO: 7

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>**QUESTION NO: 8**

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: A B C**QUESTION NO: 9**

What is the default service type in Kubernetes?

- A. ClusterIP
- B. NodePort
- C. serviceType

D. loadBalancer

ANSWER: A

Explanation:

<https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

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.