KCNA Q&As

Kubernetes and Cloud Native Associate (KCNA)

Pass Linux Foundation KCNA Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/kcna.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

To run a startup task before a Pod\\'s container starts up. What Kubernetes feature can help you accomplish this?

A. Init container

B. Sidecar container

C. Startup probe

D. DaemonSet

Correct Answer: A

Explanation: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/

A Pod can have multiple containers running apps within it, but it can also have one or more init containers, which are run before the app containers are started.

Init containers are exactly like regular containers, except:

- Init containers always run to completion.
- Each init container must complete successfully before the next one starts.

If a Pod's init container fails, the kubelet repeatedly restarts that init container until it succeeds. However, if the Pod has a `restartPolicy` of Never, and an init container fails during startup of that Pod, Kubernetes treats the overall Pod as failed.

To specify an init container for a Pod, add the `initContainers` field into the Pod specification, as an array of `container` items (similar to the app `containers` field and its contents). See Container in the API reference for more details.

QUESTION 2

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

Correct Answer: D

Explanation: https://www.aquasec.com/cloud-native-academy/container- security/container-runtime/

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.

QUESTION 3

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 4

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

A. POD

B. Deployment

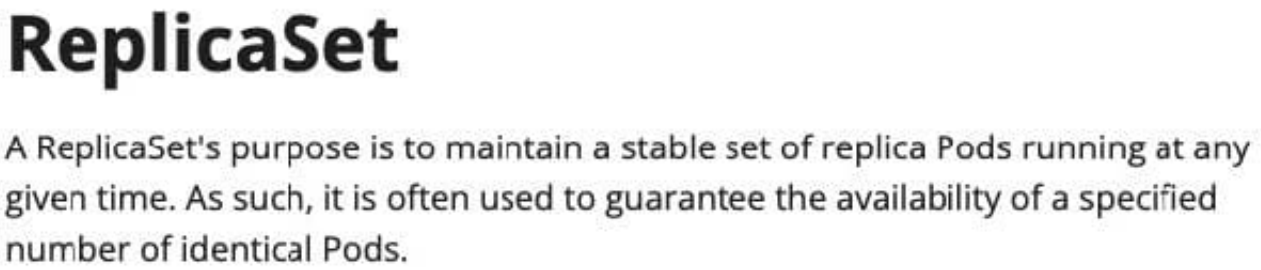
C. Horizontal pod autoscaler

D. Api Scheduler

E. Replicasets

Correct Answer: E

Explanation: https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/



QUESTION 5

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

Correct Answer: D

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