

Latest Version: 6.0

Question: 1

Which is not a service type in Kubernetes?

- A. ClusterIP
- B. NodePort
- C. Ingress
- D. LoadBalancer
- E. ExternalName

Answer: C

Explanation:

<https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>

without a Service. Services allow your applications to receive traffic. Services can be exposed in different ways by specifying a `type` in the `ServiceSpec`:

- *ClusterIP* (default) - Exposes the Service on an internal IP in the cluster. This type makes the Service only reachable from within the cluster.
- *NodePort* - Exposes the Service on the same port of each selected Node in the cluster using NAT. Makes a Service accessible from outside the cluster using `<NodeIP>:<NodePort>` . Superset of ClusterIP.
- *LoadBalancer* - Creates an external load balancer in the current cloud (if supported) and assigns a fixed, external IP to the Service. Superset of NodePort.
- *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. This type requires v1.7 or higher of `kube-dns` , or CoreDNS version 0.0.8 or higher.

More information about the different types of Services can be found in the [Using Source IP](#) tutorial. Also see [Connecting Applications with Services](#).

Question: 2

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.

Question: 3

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: 4

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/>

Question: 5

Which of the following is an advantage a cloud-native microservices application has over monolithic applications?

- A. Cloud-native microservices applications tend to be faster and more responsive than monolithic applications.
- B. Cloud-native microservice applications tend to be easier to troubleshoot.

C. Cloud-native microservice applications tend to be easier to scale and perform updates on.

Answer: C

Explanation:

Cloud-native applications tend to be microservice base, they have individual services that can be independently scaled, updated and rolled back. This makes scaling and update operations simpler and less risky.