



DEVOPS EASY LEARNING

S5GROUP5
BrainsCells
(K8s Pod Object)

Questions

Exercise 1: Kubernetes Object and Concept Differentiation

Instructions:

For each scenario or requirement provided, choose the appropriate Kubernetes object(s) and concept(s) from the list below. Some options may be used more than once, while others may not be used at all.

Kubernetes Objects:

- Deployment
- StatefulSet
- Service Account
- Pod Disruption Budget
- Pod Affinity and Pod Anti-Affinity
- Node Selector
- Horizontal Pod Autoscaler



Scenario

Scenario: You have a web application with multiple containers that need to communicate with each other. You want to control the network access and permissions for these containers.

Requirement: Manage access control and authentication for the containers within the cluster.

Kubernetes Objects: ???

Scenario: You are performing maintenance or upgrades on your cluster and want to ensure that a minimum number of pods are available during the process.

Requirement: Define a minimum number of available replicas during maintenance or upgrades.

Kubernetes Objects: ???

Scenario: You have a multi-tier application where certain pods need to be co-located or placed on different nodes based on specific rules.

Requirement: Specify affinity or anti-affinity rules to control pod placement.

Kubernetes Objects: ???

Scenario: You want to restrict the deployment of pods to nodes that meet specific criteria, such as having certain labels.

Requirement: Define constraints for pod scheduling based on node attributes.

Kubernetes Objects: ???



Scenario

Scenario: You have a stateful application that requires stable network identities and persistent storage. The application consists of multiple replicas that need to be managed individually.

Requirement: Ensure that each replica has a unique network identity and stable storage, and manage them as individual entities.

Kubernetes Objects: ???

Scenario: You want to automatically scale your application based on CPU utilization to handle varying loads.

Requirement: Scale the number of pods dynamically based on CPU resource utilization.

Kubernetes Objects: ???

Scenario: You have a web application with multiple containers that need to communicate with each other. You want to control the network access and permissions for these containers.

Requirement: Manage access control and authentication for the containers within the cluster.

Kubernetes Objects: ???

Scenario: You are performing maintenance or upgrades on your cluster and want to ensure that a minimum number of pods are available during the process.

Requirement: Define a minimum number of available replicas during maintenance or upgrades.

Kubernetes Objects: ???



Scenario

Scenario: You have a multi-tier application where certain pods need to be co-located or placed on different nodes based on specific rules.

Requirement: Specify affinity or anti-affinity rules to control pod placement.

Kubernetes Objects: ???

Scenario: You want to restrict the deployment of pods to nodes that meet specific criteria, such as having certain labels.

Requirement: Define constraints for pod scheduling based on node attributes.

Kubernetes Objects: ???



Questions

Exercise 2: Kubernetes Object Management

In this exercise, you will be working with various Kubernetes objects to deploy and manage a microservice-based application. Each task will focus on a specific Kubernetes object, You will need to identify any differences between similar objects and explain their use cases. Let's get started!

1. Explain the differences between a Deployment and a StatefulSet in Kubernetes. When would you choose one over the other?
2. Explain the purpose of a Service Account in Kubernetes and why it is important for securing cluster access.
3. Describe the significance of a Pod Disruption Budget and how it helps ensure the availability of applications during cluster maintenance or node failures.
4. Explain the differences between Pod Affinity and Pod Anti-Affinity and their use cases in Kubernetes.
5. Explain the purpose of a Node Selector and how it can be used to control pod placement within a Kubernetes cluster.
6. Describe the role of a Horizontal Pod Autoscaler in Kubernetes and its benefits for automatically adjusting the number of pods based on resource usage.



Question

- 1.Explain the differences between Pod Affinity and Pod Anti-Affinity and their use cases in Kubernetes.
- 2.Explain the purpose of a Node Selector and how it can be used to control pod placement within a Kubernetes cluster.
- 3.Describe the role of a Horizontal Pod Autoscaler in Kubernetes and its benefits for automatically adjusting the number of pods based on resource usage.



GOOD JOB

