BEYOND DEVELOPMENT: DEPLOYING CONTAINERS IN PRODUCTION**Introduction & Learning Objectives**

# Chapter Overview

### BEYOND DEVELOPMENT: DEPLOYING CONTAINERS IN PRODUCTIONKnowledge Check

Congratulations on completing Chapter 10 - Beyond Development: Deploying Containers in Production. Take this quiz to check your understanding of the concepts you've learned about so far.

BEYOND DEVELOPMENT: DEPLOYING CONTAINERS IN PRODUCTION Knowledge Check Congratulations on completing Chapter 10 - Beyond Development: Deploying Containers in Production. Take this quiz to check your understanding of the concepts you've learned about so far. Question 10.1 COE stands for: A. Container Orientation Engine B. Contianer Operations Engine C. Container Organizer Engine D. Container Orchestration Engine

ChatGPT

D. Container Orchestration Engine

COE stands for Container Orchestration Engine. Container orchestration engines are tools or platforms that automate the deployment, scaling, and management of containers in production environments. They handle tasks like container scheduling, load balancing, service discovery, and self-healing of containers.

Popular examples of container orchestration engines include Kubernetes, Docker Swarm, Apache Mesos, and Amazon ECS (Elastic Container Service). These tools help manage the complexity of deploying and running containerized applications at scale, making them an essential part of deploying containers in production.



Question 10.2 Which of the following organizations hosts Kubernetes project? A. Google B. Microsoft C. Cloud Native Computing Foundation (CNCF) D. Apache

ChatGPT

C. Cloud Native Computing Foundation (CNCF)

The Kubernetes project is hosted by the Cloud Native Computing Foundation (CNCF). CNCF is an open-source foundation that focuses on promoting the adoption and advancement of cloud-native computing technologies. Kubernetes was originally developed by Google and later donated to the CNCF in 2015. Since then, CNCF has been responsible for managing and maintaining the Kubernetes project, as well as fostering the growth of the Kubernetes ecosystem.

CNCF hosts a wide range of cloud-native projects and provides governance, marketing, and support for these projects. As one of the most prominent projects under CNCF, Kubernetes has become the de facto standard for container orchestration and is widely used in production environments for deploying and managing containerized applications.



Question 10.3 Docker Compose distributes application workload across multiple hosts. True or False? A. True B. False

ChatGPT

B. False

The statement is False.

Docker Compose is primarily designed for managing multi-container applications on a single host or a development environment. It does not have built-in capabilities for distributing application workloads across multiple hosts. Docker Compose is used for defining and running multiple containers that are closely related and need to work together on the same host.

For distributing application workloads across multiple hosts in a production environment, you would typically use a container orchestration platform like Kubernetes, Docker Swarm, Apache Mesos, or Amazon ECS. These container orchestration engines provide the necessary features for scaling and distributing containers across a cluster of nodes to ensure high availability, load balancing, and resource utilization.

In Kubernetes, for example, you can define Deployments or StatefulSets that automatically manage the desired number of replicas of your application across the cluster, ensuring that it is distributed and scheduled efficiently across multiple nodes. Similarly, Docker Swarm provides services and replicas that handle load balancing and scheduling of containers across the nodes in the swarm.

While Docker Compose is an excellent tool for local development and testing, it is not suitable for orchestrating containers across multiple hosts in a production environment. For that purpose, a container orchestration engine is the appropriate choice.



Question 10.4 Which of the following tools does not belong to COE family? A. Kubernetes B. Mesos Marathon C. Nomad D. Docker Compose

ChatGPT

D. Docker Compose

Docker Compose does not belong to the Container Orchestration Engine (COE) family. It is not a full-fledged container orchestration tool like Kubernetes, Mesos Marathon, or Nomad.

Docker Compose is a tool for defining and running multi-container Docker applications. It allows you to define a multi-service application using a simple YAML file and then use the **docker-compose** command to start and manage all the services as a single unit. It is primarily designed for local development and testing environments, as well as simple deployments on a single host.

On the other hand, Kubernetes, Mesos Marathon, and Nomad are container orchestration engines that are used for managing the deployment, scaling, and management of containerized applications in production environments. These tools offer advanced features such as service discovery, load balancing, self-healing, and auto-scaling, making them suitable for running containers at scale across a cluster of hosts.

In summary, Docker Compose is not a full-fledged COE like Kubernetes, Mesos Marathon, or Nomad. Instead, it is a tool for local development and simple container deployments on a single host.



Question 10.5 What happens when a node in Kubernetes cluster goes down? A. Kubernetes reschedules containers in some other available node B. Kubernetes waits for the failed node to come up C. Kubernetes cluster will stop working D. None of the above

ChatGPT

A. Kubernetes reschedules containers in some other available node

When a node in a Kubernetes cluster goes down, Kubernetes automatically detects the node failure and takes action to ensure the high availability of the applications running on that node. The Kubernetes control plane, which includes the master components, continuously monitors the health of the cluster and the nodes.

When a node fails, Kubernetes performs the following actions:

1. **Detect Node Failure:** Kubernetes detects that the node is not responding or has become unavailable.
2. **Mark Node as Unreachable:** The Kubernetes control plane marks the node as "NotReady" to prevent new pods from being scheduled on the failed node.
3. **Reschedule Pods:** Kubernetes reschedules the pods that were running on the failed node to other available nodes in the cluster. It tries to ensure that the desired number of replicas for each pod is maintained across the cluster.
4. **Node Recovery:** If the failed node becomes available again, Kubernetes automatically detects its availability and marks it as "Ready" again. If any pods were rescheduled to other nodes, Kubernetes may reschedule them back to the recovered node if it has sufficient capacity.

By rescheduling pods to other available nodes, Kubernetes ensures that the application workloads remain operational and distributed across the cluster, even in the presence of node failures. This automatic handling of node failures is one of the key features that make Kubernetes suitable for running containerized applications in production environments with high availability requirements.



Question 10.6 Kubernetes is created based on \_\_\_\_\_\_\_\_\_. Fill in the blank. A. Docker Swarm B. OpenShift Origin C. Borg D. Mesos Marathon

ChatGPT

C. Borg

Kubernetes is created based on Borg. Borg is the internal cluster management system developed at Google, which has been used to run their production workloads for many years. Kubernetes was inspired by and designed based on the lessons learned from Borg, and it aims to bring similar cluster management capabilities to a wider audience.

Google used Borg to manage the deployment and scaling of their applications across thousands of nodes in their data centers. Kubernetes takes similar concepts and principles from Borg and provides an open-source, portable, and extensible platform for automating the deployment, scaling, and management of containerized applications. It allows users to create, deploy, and manage applications in a highly available and scalable manner, much like Google does with Borg internally.