



Introduction

In this lesson, we'll look at a quick introduction to Kubernetes.

We'll cover the following ^

- Definition
- Licence and community
- Kubernetes versions
- Kubernetes concepts
 - Nodes
 - Pod
 - Replica set
 - Deployment
 - Services
 - Declarative
- Features
- Chapter walkthrough



kubernetes



Definition

Kubernetes (<https://kubernetes.io/>) continuously gaining importance as a runtime environment for the development and operation of microservices.

Licence and community

Kubernetes is an open source project and under the Apache license. It is managed by the Linux foundation and was originally created at Google. An extensive ecosystem has emerged around Kubernetes, offering various extensions.

Kubernetes versions

There are different Kubernetes variants.

Minikube (<https://github.com/kubernetes/minikube>) is a version of Kubernetes for installing a test and development system on a laptop.

There are more versions (<https://kubernetes.io/docs/getting-started-guides/>) that can be installed on servers or used as cloud offerings:

- kops (<https://github.com/kubernetes/kops>) is a tool which enables the installation of a Kubernetes cluster in different types of environments like AWS (Amazon Web Services).
- Amazon Elastic Container Service for Kubernetes (Amazon EKS) (<https://aws.amazon.com/eks/>) provides Kubernetes clusters on AWS.
- Google Cloud also supports Kubernetes with the Google Container Engine (<https://cloud.google.com/container-engine>).
- Microsoft Azure provides the Azure Container Service

<https://azure.microsoft.com/en-gb/services/container-service/>



- IBM Bluemix provides Kubernetes with the IBM Bluemix Container Service
(https://console.ng.bluemix.net/docs/containers/container_index.html).

Kubernetes concepts

In addition to Docker concepts, Kubernetes introduces several additional concepts.

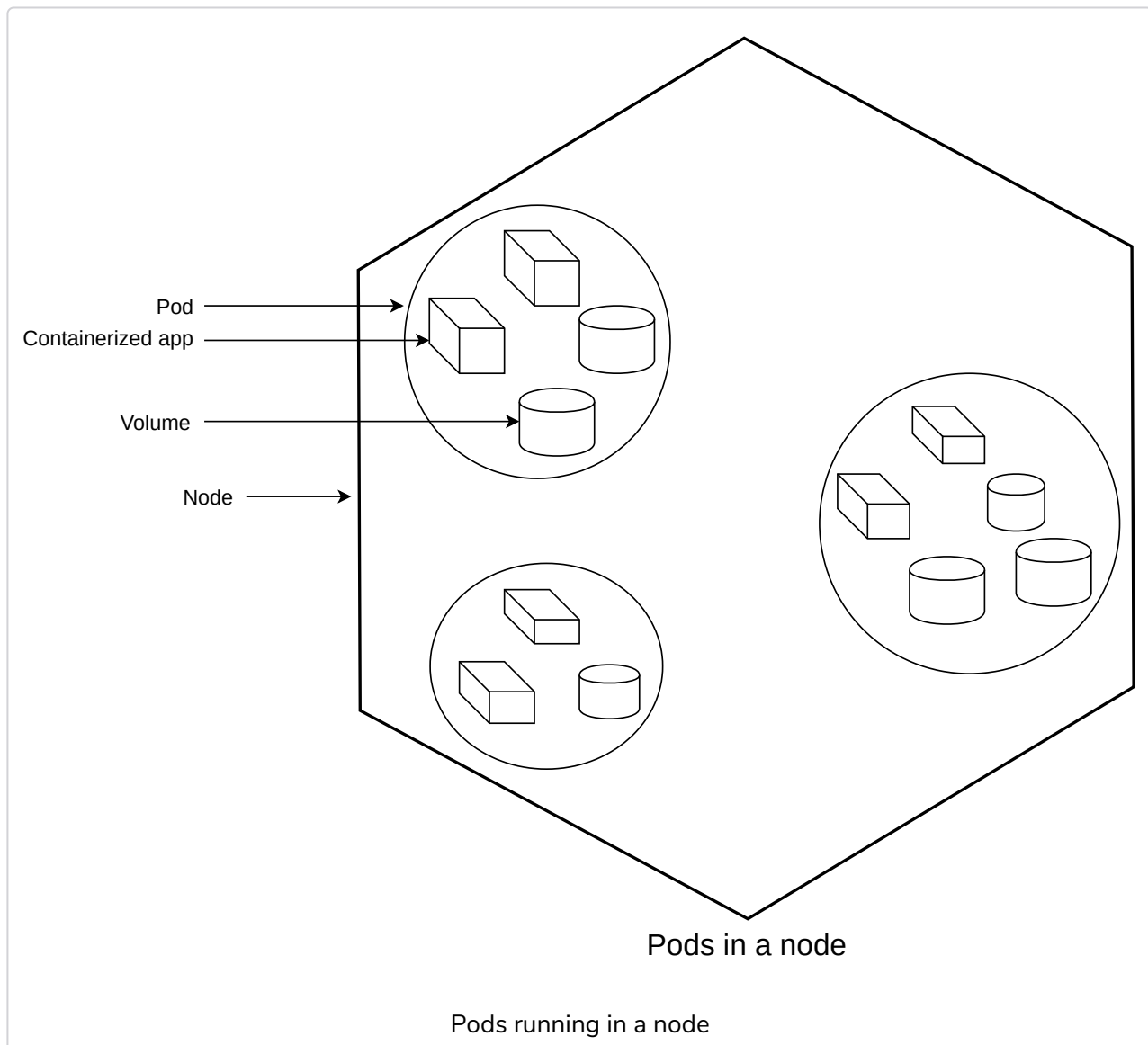
Nodes

Nodes are the servers on which Kubernetes runs. They are organized in a cluster.

Pod

A **Pod** is multiple Docker containers that together provide a service. For example, this can be a container with a microservice and a container for log processing. The Docker containers in a pod can share Docker volumes and efficiently exchange data.

Containers that belong to one pod run on one node. To scale the system to more nodes, more instances of the pod need to be started and distributed on the nodes.



Replica set

A **replica set** ensures that a certain number of instances of a pod runs. This allows the load to be distributed to the pods. In addition, the system is fail-safe. If a pod fails, a new pod is automatically started.

Deployment

A **deployment** generates a replica set and provides the required Docker images.

Services



Services make pods accessible. The services are registered under one name in the DNS and have a fixed IP address under which they can be contacted throughout the cluster. In addition, services enable routing for requests from the outside.

Declarative

Kubernetes is **declarative** meaning the configuration defines a desired state. Kubernetes makes sure that the system fits the desired state.

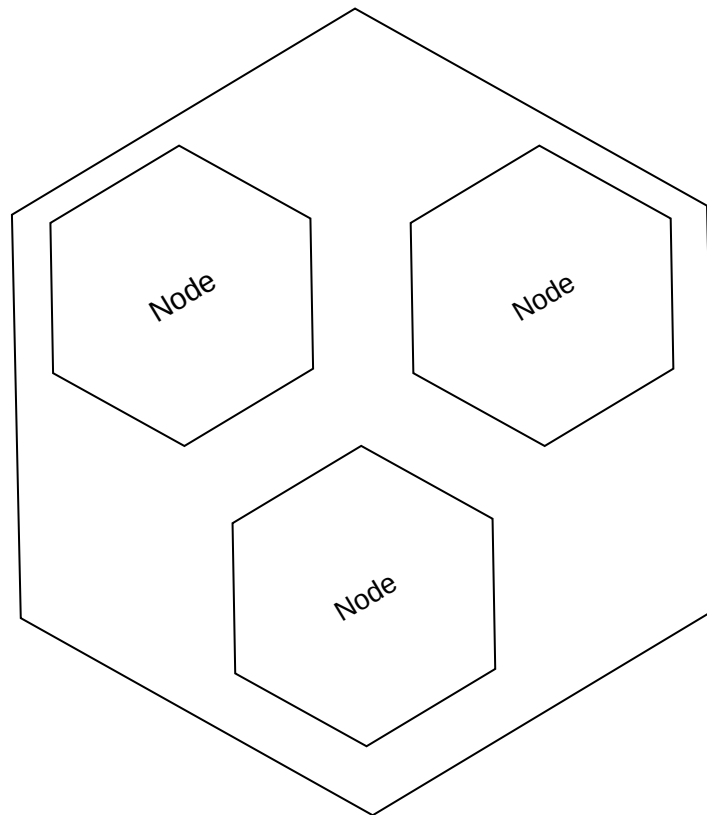
A replica set actually defines the number of pods that should be running, and it is up to Kubernetes to make the actual number of running pods match that number.

If the configuration of the replica set changes, the number of running pods is adjusted accordingly. And if some of the pods crash, enough pods are started to match the declaration in the replica set.

Features

As mentioned above, Kubernetes offers a platform based on Docker that supports important features:

- Kubernetes runs Docker containers in a **cluster** of nodes. Thus, Docker containers can use all resources the in the cluster.



Cluster

- In the event of a failure, Docker containers can be restarted. This is possible even if the original node on which the container was running is no longer available. In this way, the system achieves **fail-safety**.
- Kubernetes also supports **load balancing** and can distribute the load between multiple nodes.
- Finally, Kubernetes supports **service discovery**. Microservices that run in Docker containers can easily find each other and communicate with each other via Kubernetes.
- Since Kubernetes works on the level of Docker containers, the microservices have **no code dependencies** to Kubernetes. This is not only elegant, but means that a Kubernetes system supports virtually

all programming languages and frameworks for the implementation of microservices.



Chapter walkthrough

This chapter describes Kubernetes, a runtime environment for Docker containers. The reader gets to know the following points in this chapter:

- Kubernetes can run Docker containers in a cluster and comprises a complete infrastructure for microservices.
- Kubernetes does not introduce code dependencies into the example.
- MOMs or other tools can be run on Kubernetes.

QUIZ



Z

1 What is a pod in and what is its function?

☐ A) It is a specialized Kubernetes Docker container.

☐ B) Multiple Docker containers which provide a service together.

☐ C) A Kubernetes instance is called a pod.




Submit Answer

<

Question 1 of 4
0 attempted

>

Reset Quiz 

In the next lesson, we'll look at a Kubernetes example.

< Back

Variations

Next >

The Example with Kubernetes

☒

 Mark as Completed



 Report an Issue