QA Consulting

Kubernetes Introduction

TRAINING MATERIALS - MODULE HANDOUT

Contacts

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

Kubernetes (k8s) is and open-source solution for automating deployments of containerized applications. k8s has any solutions for managing and scaling very large quantities of containers by grouping them into manageable units. Because Kubernetes is open-source, it can be deployed as a on-premises, hybrid or public cloud solution. Cloud Services such as Google Cloud Platform and Azure offer managed Kubernetes services, which completely configure and deploy a Kubernetes cluster for you.

Kubernetes API

You can use Kubernetes API objects to describe the desired state of your cluster. A desired state would be whichever applications an workloads that you would like to have running, for example a web server, five different API servers and a database. The desired state that you would like can be created using the Kubernetes API, this is usually through a CLI such as **kubectl**. Once you have configured the desired state for your cluster, Kubernetes will automatically configure the containers in the cluster to that desired state. This includes tasks such as starting, stopping containers and scaling them.

Concepts

Objects

Kubernetes uses abstractions to represent the state of the system which is the deployed, containerized applications and workloads that are running in your cluster. Some examples of objects are:

- Service
- Pod
- Volume

Controllers

Controllers allow for additional features by building on top of the basic Kubernetes objects. For example, a Deployment controller will allow to describe a desired state such as three web servers, an API server and a database. Some examples of Controllers are:

- Deployment
- ReplicaSet
- Job

Kubernetes Master

The master in Kubernetes is what is responsible for maintaining the cluster in a desired state. When we communicate with the Kubernetes API, we are connecting to the master. For example when we use the **kubectl** tool, we are connecting to the Kubernetes API on master. So when we refer to master we are talking about a set of processes that managing the state of the cluster. These processes usually run on just one node in the cluster - the master. Some configurations may have replicated master nodes to account for availability and redundancy

Nodes

The nodes in a cluster are the actual machines such as Virtual Machines, Physical Servers etc that are running your applications. These nodes are controlled by the master node. Every node has an agent on it known as **kubelet**, which is a service that runs on the node to make sure that your applications and workloads are running and healthy.

Pod

The Pod is the smallest unit or Object in Kubernetes and represents process. The most common model is to have one container per Pod. A less popular model is a sidecar model, where two or more containers are tightly coupled. Most people avoid a sidecar model because it is not usually needed and couples processes unnecessarily.