Kubernetes Administrator

Duration 3 day(s) (KUBERNETES-ADM)

Manage Kubernetes platforms

Description

Deploying applications inside software containers are becoming more common, especially in technical architectures advocating the organization of an application as a set of services collaborating with each other. In order to better address the problem of load distribution and fault tolerance of applications deployed inside software containers, the use of an orchestration tool is strongly recommended. Using an existing solution avoids reinventing the wheel and taking advantage of the increased productivity and responsiveness associated with the use software containers. Kubernetes is one of the most cited products and used in this field, its reputation and maturity are based on the experience and know-how of Google in terms of containers. This training covers the operations aspect of the installation and the maintenance of a containerized platform based on Kubernetes.

Goals

- Understand the architecture and internal workings of a Kubernetes platform
- Know how to install and operate a Kubernetes instance
- Know how to choose different tools related to the maintenance of an Kubernetes instance to collect, monitor, and process logs and network metrics
- Know how to configure and monitor an instance of Kubernetes

Public

- IT Engineers with Linux knowledge

Prerequisites

- Knowledge of containerization technologies like Docker
- Knowledge of Kubernetes, including the different types of loads that can be deployed (Pods, ReplicaSets, Deployments, Services, Ingress, DaemonSets, StatefulSets). Ideally have completed our [Kubernetes Application Developer] training (https://training.zenika.com/en/training/kubernetes/desc or equivalent training or certified [Certified Kubernetes Application Developer] (https://www.cncf.io/certification/CKAD/).
- Good knowledge of linux systems as well as basic knowledge about the network and fundamental principles of Linux security

Structure

50% Theory, 50% Practice

Program

Architecture

- Introduction
- Architecture
- Control Plane
- Workers

Installation

- Installation steps
- Distributions
- Network solutions
- Container Runtimes
- DNS
- Storage

Configuration and operational maintenance

- Logs centralization
- Metrics centralization
- Backup / Restore
- Tools for capacity planning (node resources consumption, namespaces, requests, limits)
- Garbage Collection
- Connectivity of a Kubernetes cluster with the rest of your infrastructure
- LoadBalancers
- Ingress
- Schedulers
- Troubleshooting the cluster

Cluster Upgrade

- Kubernetes Release Cycle
- Upgrade process
- Upgrade with kubeadm
- Upgrade with kops
- Upgrade with kubespray
- Upgrade with gke
- Upgrade with eks

Day to day actions, make your Kubernetes users happy and aware

- Namespaces and Instances
- NetworkPolicies
- Common Admission Controllers
- LimitRanges
- Quotas
- Security contexts
- PodSecurityPolicy
- Authentication/Authorization/RBAC
- Persistent Volumes

Extensibility: Operators, CRD and API Servers

- Controllers

- Dynamic Admission Controllers
- Custom Resource Definition
- Operators
- API Servers/Aggregation layer

Federation, Service Mesh, Security

- Federation
  - Principles
  - History
- Service Mesh
- Security