

Kubernetes Administration – Part 1

Course Outline

Overview

This course will teach you how to use the container management platform used by companies like Google to manage their application infrastructure.

This course is designed to work with a wide range of Linux distributions, so you will be able to apply these concepts regardless of your distro.

Kubernetes is quickly becoming the de-facto standard to operate containerized applications at scale in the data-center. This course covers the fundamentals needed to understand Kubernetes and get quickly up-to-speed, to start building distributed applications that will scale, be fault-tolerant and simple to manage. From understanding its origin, to its high-level architecture, powerful API and key primitives, this course takes you from nothing to being in a position to start building complex applications. Kubernetes builds on 15 years of Google's experience managing containerized applications. With a growing open-source community, it is poised to change the way we build and manage applications, as well as change the role of system administrators. This self-paced course will distill key principles, such as pods, deployments, replicaset, and services, and will give you enough information so that you can start using Kubernetes on your own.

You'll learn:

- Kubernetes architecture
- Deployment
- How to access the cluster
- Secrets and ConfigMaps
- And much more!

Prerequisites

To get the most out of this course, you should have basic Linux command line skills and at least some knowledge of Linux containers (e.g. Docker).

Duration

3 days

Course Outline

- Introduction to Container Orchestration
- Kubernetes Features
- Kubernetes Architecture
- Master Node Components
- Worker Node Components
- Container Runtime
- Worker Node Components: Dashboard
- Infrastructure for Kubernetes Installation
- kubernetes POD networking
- kubernetes On-Premise Bare Metal/VM Installation Lab (1 master node + 3 worker node)
- Accessing Kubernetes API server

Kubernetes Administration – Part 1

Course Outline

- Kubernetes Object Model
- Establish SSH tunnel to Master node
- Pods
- ReplicaSets
- Replicationcontroller
- Deployments
- DaemonSets
- Rolling out update process
- Namespaces
- Connecting Users to Pods (Labels and Selectors)
- Node Label and Node selector
- Jobs and cronjob
- Node and POD networking (routing and overlay network)
- ServiceType: ClusterIP and NodePort
- Servicetype: Host port and Host network
- Deploying an Application Using yaml file
- Setup resource quota for POD and Namespace
- Deploying an Application Using the Dashboard
- Package Management
- introduction to Kubernetes Volume Management
- Container Storage Interface
- Persistent volume and Persistent volume claim
- ConfigMaps
- ingress control (Name based virtual hosting and fanout)
- Deploy Kubernetes aware Network Load Balancer
- Configure Kubernetes aware NLB work with ingress controller

Kubernetes Administration – Part 1

Course Outline

Exam Details

Docker Certified Associate (DCA) exam

- 13 multiple choice and 42 discrete option multiple choice (DOMC) questions in 90 minutes
- Designed to validate professionals with a minimum of 6 to 12 months of Docker experience
- Remotely proctored on your Windows or Mac computer
- Available globally in English
- USD \$195 or Euro €175 purchased online
- Results delivered immediately

Certified Kubernetes Administrator (CKA)

A certified K8s administrator has demonstrated the ability to do basic installation as well as configuring and managing production-grade Kubernetes clusters. They will have an understanding of key concepts such as Kubernetes networking, storage, security, maintenance, logging and monitoring, application lifecycle, troubleshooting, API object primitives and the ability to establish basic use-cases for end users.

The online exam consists of a set of performance-based items (problems) to be solved in a command line and candidates have 2 hours to complete the tasks.

The Certification focuses on the skills required to be a successful Kubernetes Administrator in industry today. This includes these general domains and their weights on the exam:

Domain	Weight
Cluster Architecture, Installation & Configuration <ul style="list-style-type: none"> • <i>Manage role based access control (RBAC)</i> • <i>Use Kubeadm to install a basic cluster</i> • <i>Manage a highly-available Kubernetes cluster</i> • <i>Provision underlying infrastructure to deploy a Kubernetes cluster</i> • <i>Perform a version upgrade on a Kubernetes cluster using Kubeadm</i> • <i>Implement etcd backup and restore</i> 	25%
Workloads & Scheduling <ul style="list-style-type: none"> • <i>Understand deployments and how to perform rolling update and rollbacks</i> • <i>Use ConfigMaps and Secrets to configure applications</i> • <i>Know how to scale applications</i> • <i>Understand the primitives used to create robust, self-healing, application deployments</i> • <i>Understand how resource limits can affect Pod scheduling</i> • <i>Awareness of manifest management and common templating tools</i> 	15%
Services & Networking <ul style="list-style-type: none"> • <i>Understand host networking configuration on the cluster nodes</i> • <i>Understand connectivity between Pods</i> • <i>Understand ClusterIP, NodePort, LoadBalancer service types and endpoints</i> • <i>Know how to use Ingress controllers and Ingress resources</i> 	20%

Kubernetes Administration – Part 1

Course Outline

<ul style="list-style-type: none"> • <i>Know how to configure and use CoreDNS</i> • <i>Choose an appropriate container network interface plugin</i> 	
Storage <ul style="list-style-type: none"> • <i>Understand storage classes, persistent volumes</i> • <i>Understand volume mode, access modes and reclaim policies for volumes</i> • <i>Understand persistent volume claims primitive</i> • <i>Know how to configure applications with persistent storage</i> 	10%
Troubleshooting <ul style="list-style-type: none"> • <i>Evaluate cluster and node logging</i> • <i>Understand how to monitor applications</i> • <i>Manage container stdout & stderr logs</i> • <i>Troubleshoot application failure</i> • <i>Troubleshoot cluster component failure</i> • <i>Troubleshoot networking</i> 	30%

The cost is USD \$300 and includes one free retake.

Are you ready to take Multi-Cloud & Containers training? Check your knowledge with this short quiz.

If you scored **less than 80%**, we highly recommend taking [Kubernetes Administration – Part 1](#) course. If you scored **80% or greater**, you should be ready for the [Kubernetes Administration – Part 2](#) course and other advanced containers courses.

Let's begin!

1) What is a cluster?

- a) A single system made up of servers and other resources
- b) A computer program or device that provides services to other computers
- c) A specific set of ordered operations
- d) A software package that performs a specific function for an end user

2) In the context of Kubernetes, what is a container?

- a) An application environment installed on software that imitates dedicated hardware
- b) A package that relies on virtual isolation technology to deploy and run applications that access a shared OS kernel
- c) A large metal box that is used to hold or transport something
- d) A unit that stores data on a computer's memory drive

3) What is a pod in Kubernetes and what does it do?

- a) A collection of physical IT components that supports a group of containers
- b) A collection of logic circuitry that evaluates a computer's operational code
- c) Where you find peas
- d) One or more containers grouped together to share resources and run as a unit

4) What is the function of a node?

- a) To communicate with hosts on a network
- b) To store data collected by the OS
- c) To run pods according to master components
- d) To channel incoming data from multiple input ports to a specific port

5) What is a kubelet?

- a) A node agent that manages pods and their containers
- b) Software that is used to run containers outside of pods
- c) A smaller version of Kubernetes
- d) A device that collects data in an organized manner for easy access

6) What is the purpose of a ReplicaSet?

- a) To prevent clones from invading other clusters
- b) To monitor and respond to environmental latency
- c) To duplicate pods
- d) To create and maintain volumes

7) Which description best fits a DaemonSet?

- a) A way to run a copy of a pod on all or some nodes
- b) A method to manage clusters of pods
- c) A channel through which to pass information between two or more nodes
- d) A system to connect nodes and pods

8) What is the function of labels?

- a) To classify functions
- b) To tag containers and link them together in groups
- c) To assign functions to pods
- d) To be ignored by millennials

9) Which is the intended use for etcd?

- a) To store all the cluster data, maintain its state and provide access to critical data
- b) To link a unique identifier to a value
- c) To encrypt cluster data and send it to a secrets manager
- d) To authenticate cluster data

10) What is a Kubernetes volume?

- a) The software within an OS that controls capacity allocation for nodes
- b) A directory for the data accessible to containers in a pod
- c) Layering software that puts apps into compartments for easier deployment
- d) Code that enables two software programs to communicate

Answer:

- 1) a
- 2) b
- 3) d
- 4) c
- 5) a
- 6) c
- 7) a
- 8) b
- 9) a
- 10) b

Score	Recommended Course
< 80%	Kubernetes Administration – Part 1
≥ 80%	Kubernetes Administration – Part 2 & Other advanced courses

*Check out our full schedule and course outline of **Multi-cloud & Containers Training** at [HERE](#)*