+33 (0)1 48 10 09 46
formation@ambient-it.net 140 Avenue Jean Lolive 93500 Pantin - France

Updated on 20/01/2025

Sign up

Kubernetes Advanced Training: Production Administration

3 days (21 hours)

PRESENTATION

Our advanced Kubernetes training course gives you in-depth expertise in administering and securing Kubernetes clusters in production.

In this course, you'll learn how to monitor, administer, manage users and deploy this infrastructure in production.

You'll learn how to automate installation, manage scalability and security, set up RBAC, use advanced add-ons (Istio, Cilium), and adopt GitOps practices.

This course provides a solid foundation for the CKS and CKA certifications.

For a high score, we recommend that you continue with our CKA certification preparation and CKS certification preparation. Following these one-day courses, you can take these certifications free of charge.

This course will introduce you to the latest version of Kubernetes (at the time of writing: Kubernetes 1.32).

OBJECTIVES

- Understanding how to use Kubernetes
- Master advanced Kubernetes concepts for production cluster administration
- Configure security (RBAC, network policies, secret management)
- Use advanced tools such as Istio, Helm, and GitOps practices
- Supervise, monitor and troubleshoot Kubernetes clusters
- Prepare for CKA and CKS certifications.

- Master advanced network operations on Kubernetes

- Setting up security for a Kubernetes cluster

TARGET AUDIENCE

- System administrators
- Infrastructure architect
- DevOps

PREREQUISITES

- Preferably have taken our Kubernetes training course
- Basic knowledge of a Unix system and how containers work
- Test My Knowledge

RECOMMENDATIONS FOR PRE- AND POST-COURSE READING

- "Kubernetes: Up and Running" by Kelsey Hightower, Brendan Burns and Joe Beda
- "The Kubernetes Book" by Nigel Poulton
- "Cloud Native DevOps with Kubernetes" by John Arundel
- "Kubernetes Cookbook: Building Cloud Native Applications" by S. Goasguen and Michael Hausenblas
- The Kubernetes Github page

OUR ADVANCED KUBERNETES TRAINING PROGRAM

INTRODUCTION TO MICROSERVICES

- Monolithic applications vs. microservices
  - Characteristics of a Monolithic Application
  - Advantages of a Monolithic Application
  - Disadvantages of a Monolithic Application
  - Illustrative diagram of a Monolithic Application
- What is a microservice?
  - Key principles of microservices
  - Advantages of microservices
  - The challenges of microservices
  - Tools and Technologies for Micro-Services
  - Micro-services and Kubernetes
  - Comparison with a microservices architecture

KUBERNETES ADMINISTRATION IN PRODUCTION

- Kubeadm: A Kubernetes deployment tool
  - What is Kubeadm?
  - Other tools for deploying Kubernetes
- Advanced kubeadm configuration
- Practical work
- Deploying a high-availability Kubernetes cluster
- Automated setup of an On-Premise Kubernetes cluster
- Securing an On-Premise Kubernetes cluster for production
- Setting up high availability for the Control-Plane
- Automated Rolling Update of Control-Plane and Kubernetes nodes
- Virtuoso use of kubectl for CKAD
- Continuous integration in the cloud with kind
- Runtimes: crictl, Docker and Containerd

CONTROL PLANE COMPONENTS AND WORK NODES

- Introduction
- Control Plane components
- API Server (kube-apiserver)
- etcd
- Scheduler (kube-scheduler)
- Controller Manager (kube-controller-manager)
  - Cloud Controller Manager (cloud-controller-manager)
- Work node components
- How the reconciliation loop and Kubernetes Controller work
  - The reconciliation loop
  - How Kubernetes controllers work
- Internal operation of the API Server: Authentication, Authorization and Admission Control
  - Internal workings of the API Server
  - Admission controller management
  - Extending the API server lifecycle with Admission Webhooks
- Extending the API server lifecycle with MutatingAdmissionWebhook and ValidatingAdmissionWebhook
- Declarative configuration
- Implicit or dynamic grouping
- Kinematics of Pod creation from Deployment
- Kube-proxy, advanced operation of the virtual service network
- Service discovery with CoreDNS

ACCESS MANAGEMENT WITH RBAC AND USERS

- Introduction to RBAC
  - What is RBAC?
  - Why use RBAC?
- Basic RBAC concepts
  - Role, ClusterRole, RoleBinding and ClusterRoleBinding
- GrouApi, Resources and verbs
  - Key concepts of RBAC in Kubernetes
  - PLC groups (apiGroups)
  - Resources
  - Verbs
  - Relationships in RBAC

- User management and RBAC
  - Prerequisites and assumptions
  - RBAC API objects
    - Use cases
  - User creation and authentication with X.509 client certificates
- Authentication: certificates, tokens
- User and authorization management
  - Installing KREW
  - rakkess
  - kubect-who-can
  - rbac-lookup
  - RBAC Manager

LimitRange and ResourceQuota in Kubernetes

- Introduction to Namespaces in Kubernetes
  - Namespaces offer the following advantages
- LimitRange management
  - What is LimitRange?
  - LimitRange configuration
- ResourceQuota management
  - What is ResourceQuota?
- Limiting resources per user: Background and solutions
  - Using ResourceQuotaScopes with labels
- Scopes in Kubernetes
  - Scopes available
  - Using scopes
  - Using scopeSelector
- PriorityClass in Kubernetes
  - PriorityClass functionality
  - PriorityClass configuration
  - Using PriorityClass in Pods
  - Preemption with PriorityClass
- TP: LimitRange, resourcequota

NETWORK POLICIES IN KUBERNETES

- Introduction to Network Policies
  - What is a Network Policy?
  - Components of a Network Policy
  - Basic syntax of a Network Policy
- Practical work
  - TP1: Open port 80 on a pod
  - TP2: Restrict traffic between prods
  - TP3: Restricting traffic between pods and namespaces
  - TP4: Authorize outgoing traffic outside the cluster

INFRASTRUCTURE AS CODE, GITOPS

- Understanding IaC
  - Basic principles of IaC
  - IaC in Kubernetes
  - IaC tools for Kubernetes
  - Benefits of IaC in Kubernetes
- Understanding GitOps
  - GitOps fundamentals
  - How GitOps works
  - Popular GitOps tools
  - Benefits of GitOps
  - Example of a GitOps workflow with Kubernetes
  - Conclusion
- Overview of package managers for Kubernetes Helm, Kustomize
- What is Helm?
- What is Kustomize?
- Helm vs. Kustomize
- Automate deployments with Flux and ArgoCD
  - Flow
  - ArgoCD
  - Comparison

INGRESS CONTROLLERS AND NGINX INGRESS CONTROLLER

- What is an Ingres Controller?
  - Key features of Ingress Controllers
- NGINX Ingress Controller
  - Key features of the NGINX Ingress Controller
  - Installing and configuring the NGINX Ingress Controller
  - Installing the NGINX Ingress Controller
  - Installation via Helm
- TP: Ingress controller, Ingress

NETWORKS - MESH SERVICE

- Understanding ISTIO, Cilium and Ingress Controllers
- Choosing a secure, high-performance network Add-On
- Deploy ingress, gateways and routes for applications
- Manage network flows

SAFETY

- Securing the execution of Unix processes in Pods
- SecurityContext
  - Privileged mode
  - Linux Capabilities
  - Securing Unix processes
- Industrialize Pod security with PodSecurityPolicies
- Industrialize network security (L4) with NetworkPolicies
- Industrialize certificate management with Cert-Manager
- Discover OPA and Falco

QUALITY OF SERVICE

- Optimum use of material resources with Requests and Limits
- QoS classes
  - Guaranteed
  - Burstable
  - BestEffort
- Namespace resource allocation control with ResourceQuota
- Pod-based resource allocation control with LimitRanges

SCHEDULER OPTIMIZATION

- Planning control with Labels and Affinities
- NodeSelector, NodeAffinity, PodAffinity, PodAntiAffinity
- Taints and Tolerations

THE OPERATORS

- Presentation of Kubernetes extension methods: Operators
- Understanding the use of CRD resources
- Adding custom APIs to Kubernetes: CustomResourceDefinitions
- Deploy a monitoring stack with the Prometheus Kube state metrics operator

MONITORING

- Monitoring and logging objectives
- Automate monitoring with Prometheus operator
- Obtain and aggregate metrics for your cluster and applications
- Visualize and interact with your data with Grafana

PRODUCTION STORAGE MANAGEMENT

- Understanding hyperconverged and highly available storage
- Deploying ceph with rook operator
- Deploying NAS storage

OPERATORS, HELM & EFK (+1 DAY)

- Presentation of Kubernetes extension methods: Operators
- Adding custom APIs to Kubernetes: CustomResourceDefinitions
- Create operators with Operator-Framework and Operator-SDK
- Helm 2 and Helm 3
- Log management with the EFK stack (ElasticSearch, Fluentd, Kibana)

INTRODUCTION TO ISTIO & LINKERD (+1 day - on request only)

in a team)

- Service Mesh
- ISTIO
- LINKERD2 (Conduit)

Additional Cloud Modules

Preparing production (1 day)

- CI/CD pipeline: theory and implementation (GihubActions/ArgoCD)
- Mesh Services: how they work and a case study with Istio
- Ingress: operation and case study with nginx-controller

Public Cloud or Multi-Cloud container management services: the examples of Google Kubernetes Engine and Rancher (½ day)

Advanced deployment tools for Kubernetes (½ day)

- Operators
- Adding custom APIs to Kubernetes: CustomResourceDefinitions
- Create operators with Operator-Framework and Operator-SDK
- Helm: presentation and example with EFK log management (ElasticSearch, Fluentd, Kibana)

Support and advice on practical cases proposed by trainees (½ day to 1 day)

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning on entry to training complies with Qualiopi quality criteria. As soon as

On final registration, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.