FORMATION KUBERNETES AVANCE POUR LA PRODUCTION (3 jour + 1 jour en option)

PRÉSENTATION

Le logiciel libre Kubernetes (communément appelé « K8s ») est désormais le standard en terme d'orchestration de conteneurs. Cet outil vous permettra d'entrer dans l'ère "Cloud Native" et d'exposer à grande échelle vos applications de manière sûre, reproductible et flexible. Vous apprendrez également à faire évoluer vos applications vers le standard micro-service, modulaire et scalable. Plébiscité par les géants de la Silicon Valley, K8s est géré par une gouvernance responsable liée à Cloud Native Computing Foundation (une entité de la Fondation Linux). Kubernetes fournit une « plateforme pour automatiser le déploiement, la mise à l'échelle et la mise en production de conteneurs d'applications sur des grappes de serveurs ». Il supporte de multiples moteurs d'exécution de conteneurs dont Docker, Rocket et Singularity.

Cette formation s'adresse aux experts souhaitant mettre en oeuvre et maîtriser des clusters Kubernetes de production, ainsi qu'à à toute personne désirant comprendre en détail comment Kubernetes est architecturé, installé et maintenu. Le déploiement d'applications hautement sécurisées sur Kubernetes est également décrit, cette formation s'adresse aussi bien aux ingénieurs systèmes souhaitant mettre en place des clusters Kubernetes sûrs, performants et hautement disponibles qu'aux Devops et développeurs désirant déployer rapidement et simplement leurs applications.

Cette formation vous présentera la toute dernière version de Kubernetes (à la date de rédaction de l'article : Kubernetes 1.17).

OBJECTIFS

- Comprendre comment utiliser Kubernetes.
- Comprendre l'architecture interne de Kubernetes
- Appréhender les principaux composants avancés de Kubernetes
- Savoir installer et gérer Kubernetes en production.
- Sécuriser le cluster Kubernetes et les pods applicatifs
- Maîtriser le fonctionnement des réseaux virtuels Kubernetes
- Optimiser le monitoring du cluster Kubernetes
- Étendre et customiser les rouages de Kubernetes

PUBLIC VISÉ

Développeurs, Architectes, Administrateurs systèmes, DevOps

PRÉ-REQUIS

Connaissances de base d'un système **Unix**, du fonctionnement de **Kubernetes** et des **Linux Containers**.

PROGRAMME

ADMINISTRATION DE KUBERNETES EN PRODUCTION

- Configuration avancée de **kubeadm**
- **Mise en place automatisée** d'un cluster Kubernetes On-Premise
- **Sécurisation** d'un cluster Kubernetes On-Premise pour la production
- Mise en place de la **haute disponibilité** pour le Control-Plane
- Mise à jour automatisée en mode **Rolling Update** du Control-Plane et des noeuds Kubernetes
- Virtuosité dans l'utilisation de **kubectl** pour la **CKAD**
- Intégration continue dans le Cloud avec kind
- **CRI**: crictl, Docker et Containerd

ARCHITECTURE DE KUBERNETES

- Les composants du **Control Plane** et des noeuds de travail
- Philosophie Unix des composants
- Fonctionnement de la boucle de réconciliation et du **Controller** Kubernetes
- Fonctionnement de **etcd** en mode haute-disponibilité
- Fonctionnement interne de l'**API server**: authentification, autorisation et **Admission Control**
- Gestion des contrôleurs d'admission
- Extension du cycle de vie du serveur d'API avec les **MutatingAdmissionWebhook** et les **ValidatingAdmissionWebhook**
- Description de l'algorithme du **Scheduler** Kubernetes, prédicats et priorités
- Configuration déclarative
- Groupement implicite ou dynamique
- Interactions pilotées par les API
- Cinématique de création d'un Pod à partir d'un Deployment
- **Kube-proxy**: fonctionnement avancé du réseau virtuel des Services
- Service discovery avec **CoreDNS**
- Description de la structure interne d'un Pod et du conteneur d'infrastructure

GESTION DES UTILISATEURS ET DROITS D'ACCÈS

- Authentification: certificats, **tokens**, et **Dex**

- Paramétrage du fichier Kubeconfig avec les **Configuration Contexts**
- Gestion des ServiceAccounts
- Sécuriser le pilotage du cluster avec les autorisations **RBAC**
- **Role** et **ClusterRole, RoleBinding** et **ClusterRoleBinding**
- Création de RBAC simples et génériques pour piloter un cluster de production

SECURITE

- Sécuriser l'exécution des processus Unix dans les Pods
- **SecurityContext:** Mode privileged, Linux Capabilities, sécurisation des processus Unix
- Industrialiser la sécurité des Pods avec les **PodSecurityPolicies**
- Choix d'un plug-in réseau CNI sécurisé et performant
- Industrialiser la sécurité du réseau (L4) avec les **NetworkPolicies** (ingress et egress)

QUALITE DE SERVICE

- Utilisation optimale des ressources matérielles grâce aux **Requests** et **Limits**
- Classes de QoS: **Guaranteed**, **Burstable** et **BestEffort**
- Contrôle d'allocation des ressources par Namespace avec les **ResourceQuota**
- Contrôle d'allocation des ressources par Pod avec les **LimitRange**

OPTIMISATION DU SCHEDULER

- Contrôle de la planification avec les Labels et les Affinités
- **NodeSelector, NodeAffinity, PodAffinity, PodAntiAffinity**
- **Taints and Tolerations**

MONITORING

- Objectifs de surveillance et de journalisation
- Automatiser le monitoring avec l'opérateur **Prometheus**
- Obtenir et agréger les métriques de votre cluster et de vos applications
- **AlertManager**: gestion et routage des alertes
- Visualiser et interagir avec vos données avec **Grafana**

EN OPTION (1 jour)

- Présentation des méthodes d'extension de Kubernetes: les **Operators**
- Ajouter des API customisées à Kubernetes: les **CustomResourceDefinitions**
- Créer ses opérateurs avec l'**Operator-Framework** et l'**Operator-SDK**
- **Helm** 2 et Helm 3
- Gestion des logs avec la pile EFK (**ElasticSearch, Fluentd, Kibana**)