

Formation **High Five**



Pack **DEVOPS** (3 jours)

Publics concernés : Développeurs, Leads techniques, Architectes.





Objectifs

Cette formation de trois jours au format **workshop** vous permet de maîtriser les **outils de développement** liés à la philosophie **DevOps** en les appliquant à un environnement conteneurisé sous **Docker** et orchestré par **Kubernetes** : **Gitlab, Helm, Skaffold, Jaeger/opentelemetry**, etc.

- Vous apprendrez à configurer les outils sur votre **poste de travail** local et à gérer l'ensemble des **configurations** d'un projet ;
- Vous apprendrez tout le nécessaire pour tirer partie des conteneurs et de Kubernetes dans le développement des applications ;
- Vous travaillerez sur des exemples de **CI/CD** (Continuous Intégration / Continuous Delivery & Distribution) afin d'appréhender toutes les possibilités d'**automatisation**.

Cette formation vous permet non seulement de correctement **développer vos applications**, mais aussi de faciliter leur **déploiement** et leur **opération** (sécurité, monitoring, etc.).

Prérequis

Il est nécessaire d'avoir une bonne connaissance du **système Linux** et des lignes de commande (naviguer dans le système de fichiers, créer et modifier un fichier avec un éditeur comme Vi..).

Aucune connaissance préliminaire n'est nécessaire en terme de langage de programmation, ni sur les conteneurs et leur orchestration, ni sur les outils DevOps.

Modules de formation de notre pack DevOps

Chacun de ces modules de formation a une durée d'une journée.

1. Module "Fondamentaux Conteneurs et Docker "



OBJECTIFS

Maîtriser un **container engine** comme **Docker** pour **développer ses applications** (pour les développeurs) ou les **déployer** (pour les administrateurs système).

DESCRIPTION

Lors de cette journée de formation au format **workshop**, vous apprendrez tout ce qui est nécessaire pour bien démarrer avec les **conteneurs** et en tirer parti dans le **développement de vos applications**.

Chaque concept présenté fait l'objet d'**explications théoriques**, puis de **démonstrations concrètes** que vous serez invité-e-s à reproduire avec notre aide. Cet accent fort sur la pratique vous permettra d'être opérationnel-le avec les conteneurs dès l'issue de la formation.

La plupart des exercices mettent en œuvre **Docker**, mais vous serez aussi capables d'utiliser d'autres plateformes.

CONTENU

Lancer nos premiers conteneurs

- Définition des objectifs
- Qu'est-ce qu'un *container engine*?
- Premier contact
- Exécution interactive ou en arrière-plan

Construire des images

- La différence entre un conteneur et une image
- Fabriquer nos premières images à la main
- Rendre le processus automatique et reproductible
- Écrire son premier *Dockerfile*



Optimiser les images et leur construction

- Tirer parti du cache afin d'optimiser la vitesse de *build*
- Utiliser un *multi-stage build* afin d'optimiser la taille des images
- Bonnes pratiques pour la construction des images

Suivre ses conteneurs à la loupe

- Associer un identifiant unique à un conteneur
- Inspecter un conteneur en détail, en *live* ou *post-mortem*
- Collecter les *logs*

Connecter les conteneurs

- Description du modèle réseau des conteneurs
- Exposer un conteneur
- La *service discovery*

Développer au jour le jour

- « Containeriser » son environnement de développement
- Utiliser *Compose* pour décrire son environnement
- Écrire un *Compose file* pour connecter les services entre eux

2. Module "Fondamentaux Orchestration et Kubernetes"



OBJECTIFS

Maîtriser les **fondamentaux de l'orchestration et de Kubernetes**, afin de correctement développer ses applications (pour les développeurs), les déployer et les opérer (pour les administrateurs système).

DESCRIPTION

Cette journée de formation au format **workshop** vous permettra de prendre en main Kubernetes, comprendre ses concepts et son architecture, et être capable de l'utiliser pour vos applications.

Elle met l'accent sur les principes « **Cloud Native** » et les architectures en **micro-services**, mais est tout autant pertinente pour les applications **monolithiques** traditionnelles.

Le contenu de la formation et des exercices est adapté quel que soit le modèle de déploiement **Kubernetes : cloud public** (AKS, EKS, GKE...), **cloud privé** (OpenStack...), ou **on-premises** (sur vos propres machines en datacenter).

CONTENU

Concepts Kubernetes

- Que peut-on faire avec Kubernetes ?
- Des conteneurs aux pods
- Le plan de contrôle
- Modèles impératif et déclaratif
- Le réseau dans Kubernetes

Lancer nos premiers conteneurs avec Kubernetes

- Se familiariser avec kubectl
- Créer un déploiement, un *replica set*, un pod, un *cron job*



Service discovery et load balancing

- Exposer un service
- Différents types de services : ClusterIP, NodePort, LoadBalancer
- Utiliser CoreDNS pour la *service discovery*

Déployer une application de A à Z

- Démarrer des micro-services avec kubectl
- Connecter les micro-services ensemble
- Passage à l'échelle et répartition de charge

Mettre à jour les conteneurs

- Mise à jour d'un déploiement
- Concept de *rolling update*
- Paramétrer la vitesse de déploiement

Contrôler l'état des applications

- Le rôle des *healthchecks*
- Les différentes sondes : *readiness, liveness, startup*
- Comprendre quel type de sonde utiliser, et quand

Définir ses ressources en YAML

- Exemple : transformer un déploiement en daemon set
- Exporter le YAML des ressources
- Traduire un type de ressource en un autre

Plongée au cœur des labels et selectors

- Comprendre où ces concepts apparaissent
- Les utiliser pour mettre à jour la configuration d'un *load balancer*
- Comprendre les templates de ressources

La ressource ingress

- Gestion du trafic HTTP entrant
- Déploiement de Traefik comme *ingress controller*
- Exposition d'applications HTTP et répartition de charge

3. Module "Méthodologies DevOps"



OBJECTIFS

Maîtriser les **outils de développement liés à la philosophie DevOps** en les appliquant à un environnement Kubernetes : Gitlab (Git, CI/CD), Helm, Skaffold, Jaeger/opentelemetry, etc.

Ce module de formation vous permettra non seulement de correctement **développer vos applications**, mais aussi de faciliter leur **déploiement** et leur **opération** dans un environnement épousant l'approche et les outils DevOps.

DESCRIPTION

Cette journée de formation au format **workshop** met l'accent sur les **outils du développeur** confronté à mettre une application déjà "containerisé" sur un cluster kubernetes.

Elle donne tous les éléments pour configurer les outils sur leur **poste de travail local** et apprendre à gérer l'ensemble des **configurations** d'un projet.

Elle aborde des exemples de **CI/CD** (Continuous Intégration / Continuous Delivery & Distribution) afin d'appréhender toutes les possibilités d'**automatisation**.

Un focus particulier est réalisé sur les **outils** suivants: Gitlab (Git, CI/CD), Helm, Skaffold, Kaniko, Sealed-secrets, jaeger/opentelemetry, prometheus, grafana.

Les outils présentés dans cette formation permettent également de faciliter l'opération de vos applications, avec un **monitoring adapté** et sans oublier bien sûr les aspects **secrétité** !

CONTENU

Travailler en local

- Lancer un cluster en local
- Gérer les *namespace* et contextes de multiples *clusters*
- Accéder à des services distants depuis un poste local
- Comment tester du code avec Docker



Industrialiser

- Gestion des configurations
- Packaging d'applications avec Kustomize et Helm
- Création de Helm Chart
- Bien gérer les secrets

Automatiser

- Fabriquer une image avec Kaniko
- Déployer chaque *feature* dans un environnement
- Les différentes *registries* disponibles
- Héberger soi-même sa *registry* et sécuriser ses images
- Techniques avancées : déploiement *blue/green, canary*

Observer

- Mise en place de sonde de monitoring applicatif
- Utilisation des outils Prometheus et Grafana
- Adaptation d'une application au système OpenTelemetry
- Utilisation basique d'une application de télémétrie

Sécuriser

- Cas d'usage : déployer sans les droits *root*
- Les *capabilities*

Outils utilisés dans les travaux pratiques :

Gitlab (parties Git et CI/CD), **helm**, **skaffold**, **kaniko**, **sealed-secrets**, jaeger, opentelemetry, **prometheus**, **grafana**.

Autres outils évoqués :

Argo, **jenkins/jenkins-x**, **knative/tekton**, **buildbot**, **spinnaker**, les alternative de **skaffold**, **kamus**.

Nos formateurs

Nos formateurs sont des **experts reconnus** par la communauté **open source**.

Ils interviennent lors de **conférences IT** majeures en **France** (comme Devvxx, Devops DDay...) ainsi qu'à **l'international** (O'Reilly Velocity, LinuxCon, Usenix LISA, DevopsDays...).

Ils réalisent également des workshops et des meetups réguliers auprès de tout type d'audiences techniques : DEV, DEVOPS, OPS.

Jérôme Petazzoni



Jérôme fait partie de l'équipe qui a **conçu et popularisé Docker**. Il a passé 7 années au service de la fameuse startup Californienne, durant lesquelles il a été responsable du PaaS dotCloud, avant de représenter Docker lors de conférences internationales comme LinuxCon, Yandex YAC, OSCON, Velocity, LISA, ou encore AWS re:invent. **Depuis 2014, il a formé plusieurs milliers de personnes sur les conteneurs et l'orchestration.**

Avant Docker, il a été co-fondateur d'Enix, et il a porté diverses casquettes dans les domaines de l'encodage et la diffusion de flux vidéo, les systèmes géographiques, la fibre optique, et la voix sur IP. Il est titulaire d'un DEA de l'Université de Marne-la-Vallée Paris-Est, possède un nombre indéterminé d'instruments de musique, et sait jouer le thème de **Zelda** sur la plupart d'entre eux.

Julien Girardin



Après une formation d'électronicien, **Julien** s'est rapidement bien plus intéressé à **Linux**, l'**open source** et au développement informatique. Il prend son premier poste en tant qu'admin-sys dans une petite société, puis il rejoint SmartJog en tant que développeur **Python**. Il y fera de nombreux déploiements avec **Ansible** (dont il **co-animera le meetup** parisien) pour la construction et le déploiement d'un CDN (Content Delivery Network). C'est dans ce contexte que Julien s'initie aux conteneurs et teste un nouveau logiciel à l'air prometteur dans sa **version 1.0 : Kubernetes**.

Il affutera ensuite ses qualités de développeur python chez Scality, un des leaders mondiaux du SDS (Software Defined Storage), où il aura plusieurs casquettes: **DevOps** dans l'équipe d'automatisation des releases, ainsi que **lead-dev** pour la transition à Kubernetes. En Octobre 2019, Julien rejoint Enix en tant qu'expert Kubernetes et pour y être formateur notamment sur les sujets DevOps à destination des **développeurs**. Julien aime vivre dangereusement en ayant archlinux et btrfs sur tous les ordinateurs qu'il possède.



Méthodes et moyens pédagogiques : notre format workshop !

Chaque module de formation fait l'objet d'une **présentation théorique**, suivie d'une **démonstration** des concepts dans un environnement réel.

Puis, les participant·e·s sont invité·e·s à reproduire les démonstrations et à réaliser de **nombreux exercices concrets** dans un environnement personnel. Cet environnement est pré-installé pour être immédiatement opérationnel, mais possède tous les attributs d'un environnement réel, laissant libre cours à l'expérimentation.

Les travaux pratiques se feront sur des machines virtuelles (fournies par nos soins) auxquelles vous vous connecterez via SSH. Vous devrez donc venir à la formation **avec votre ordinateur portable**.

Etant donné que tous les logiciels utilisés sont en anglais, et que certains termes techniques n'ont pas de traduction précise en français, les **supports de formation fournis sont en anglais**. En revanche, la **formation est en français**, par des intervenant·e·s bilingues français/anglais.

Modalités d'évaluation et de sanction de la formation

Les participant·e·s sont évalué·e·s en permanence par les intervenant·e·s, qui s'assurent à intervalles réguliers de la bonne compréhension du support et des exercices. Les résultats attendus des exercices sont clairement exprimés dans le support de formation, permettant à chacun·e de vérifier son avancement et la bonne exécution des travaux pratiques.

Durant toute la formation, il est possible de poser des questions en personnes, mais aussi via une chat room qui facilite l'interaction, en particulier pour les stagiaires timides ou ayant peur de prendre la parole en public.

A l'issue de la formation, il est remis à chaque participant une attestation d'assiduité et une attestation de fin de formation.

Informations pratiques

Nous vous accueillerons chaque jour **à partir de 9 heures** du matin dans un **lieu situé dans le centre de Paris**. Ce lieu vous sera spécifié lors de votre inscription.

Nous vous servirons un **petit-déjeuner** (café, jus d'orange, viennoiseries) et la formation débutera aussitôt après. Le programme se termine vers 17h30. Nos intervenant·e·s restent ensuite à votre disposition si vous souhaitez poser des questions en privé ou obtenir des précisions sur le contenu présenté.

La société Enix est un organisme de formation certifié CNEFOP et listé sur [Datadock](#).

Type d'action: action de formation

Vous avez toujours des questions?

- Personnaliser votre parcours de formation en retirant ou en ajoutant un module?
- Assister à plusieurs ou offrir cette formation en interne dans votre entreprise?
- Connaître nos tarifs et nos réductions?

Notre responsable des relations avec les stagiaires: Magalie Barbier, +33 1 86 76 08 08, formation@enix.io



Call us
+33 1 86 76 08 08



Email us
formation@enix.fr



Address
94 boulevard de Sébastopol, Paris