

Mis à jour le 23/06/2025

S'inscrire

Formation Screwdriver.CD

3 jours (21 heures)

Présentation

Notre formation Screwdriver.cd vous permettra de deployer une plateforme CI/CD open source pensée pour les environnements cloud-native et micro-services. Conçue par Yahoo, aujourd'hui incubée par la Continuous Delivery Foundation, elle orchestre chaque jour des dizaines de milliers de builds conteneurisés, du commit jusqu'au déploiement automatisé.

Notre formation Screwdriver.cd vous apprendra à industrialiser vos pipelines de build, test et déploiement, à connecter la plateforme à GitHub ou GitLab, à sécuriser vos clés et tokens, à utiliser les templates réutilisables et à publier vos images dans un registre Docker.

À l'issue de notre programme, vous serez capables d'installer et configurer la dernière version stable de Screwdriver.cd, de comprendre son architecture modulaire et surtout de construire des workflows CI/CD hautement scalables pour vos applications cloud-native comme pour vos workloads plus traditionnels.

Comme toutes nos formations, elle se déroulera sur la dernière version de l'outil : [Screwdriver 4.1](#)

Objectifs

- Comprendre l'architecture de Screwdriver
- Définir des pipelines as code complets
- Gérer les secrets et la sécurité
- Intégrer Screwdriver.cd à votre cosystème DevOps

Public visé

- Ingénieurs DevOps / SRE
- Développeurs Full-Stack
- Architectes Cloud-Native

- Chefs de projet techniques

Pré-requis

- Connaissances de base Linux, ligne de commande
- Compréhension des workflows Git (branch, PR)
- Bases sur Kubernetes et Docker

Pré-requis techniques

- Git installé et accès à un compte GitHub ou GitLab

Programme de notre formation Screwdriver.CD

Genèse et vision de Screwdriver.cd

- Contexte historique : des outils CI legacy à Yahoo !
- Objectifs clés : simplicité, scalabilité, pipelines as code
- Place dans la CDF et gouvernance open source
- Cas d'usage phares (microservices, cloud-native)
- Tour d'horizon des réussites en production (60 k builds/jour chez Yahoo)

Anatomie de la plateforme

- Vue macro : API, UI, Launcher, Queue, Store
- Executor plug-in : Docker, Kubernetes, Jenkins, Nomad
- Datastore et gestion d'état (MySQL, PostgreSQL, SQLite)
- Cycle de vie d'un build de A à Z
- Sécurité : isolation par conteneurs & secrets chiffrés

Installer sa forge CI/CD

- Pré-requis systèmes : Docker, Git, OAuth SCM
- Choix du déploiement : Docker Compose vs Helm Chart
- Configuration initiale (config/local.yaml)
- Raccordement OAuth (GitHub/GitLab/Bitbucket)
- Validation de l'instance via l'UI
- Atelier pratique : Déployer Screwdriver.cd en local
- Démarrer la stack Docker Compose, se connecter, vérifier l'API

Pipelines as Code

- Structure du fichier screwdriver.yaml
- Jobs, Steps, Stages : fan-in / fan-out, jobs manuels
- Triggers : commit, PR, pipeline distant, cron
- Variables & métadonnées inter-jobs
- Visualiser et relancer un pipeline dans l'UI
- Bonnes pratiques de nommage et découpage
- Atelier pratique : Créer un pipeline Node.js complet
- Lint, tests, build image Docker, publication artefact

Secrets & configuration dynamique

- Concepts : scope, allowInPR, rotation
- Ajout via UI / CLI / API
- Injection dans les conteneurs de build
- Sécurité et audit des accès
- Modèles d'utilisation (API keys, kubeconfig, tokens Slack)

Exécution avancée et gestion des artefacts

- Choisir son executor : critères de décision
- Isolation, performance et passage à l'échelle
- Cache de build partagé entre jobs
- Store : logs, fichiers, rapports de couverture
- Debug : hooks, redémarrage, timeouts paramétrables

Réutilisabilité : templates & commands

- Templates de pipelines (sd-template.yaml)
- Commands partagées (sd-command.yaml)
- Publication et versioning des templates
- Héritage et surcharge de paramètres
- Diffusion des bonnes pratiques DevOps à l'échelle

Intégrations & notifications

- SCM : checks de PR, statuts de commit
- Notifications : Slack, Discord, e-mail, webhooks custom
- Registres Docker : push automatisé (ECR, ACR, Docker Hub)
- SonarQube, Snyk : scans qualité & sécurité dans la chaîne
- Triggers croisés : synchroniser plusieurs microservices
- Atelier pratique : Pipeline avec notification Slack et push image
- Ajout secret webhook, publication d'image, message de succès/échec

Déploiement continu & extensions

- Déploiement canary / blue-green sur Kubernetes

- Pipelines multi-environnements (Dev ? Staging ? Prod)
- Executor VM pour workloads sensibles
- Monitoring et métriques de la plateforme
- Stratégies de migration depuis un autre outil CI/CD
- Feuille de route du projet & contribution open source

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.