OPEN DIGITAL SERVICE

Scannez-moi

WHATSAPP
+213 674 05 38 95

SITE WEB
osdigitalis.com

FICHE FORMATION · OSD

Intégration et déploiement continus : votre premier pipeline CI/CD

Construire, tester et déployer automatiquement une application à chaque modification

20 heures | 10 séances | GitHub Actions | GitLab CI

Une formation pratique pour **mettre en place un pipeline d'intégration et de déploiement continus** : chaque modification poussée sur Git est automatiquement construite, testée, empaquetée et déployée.

20 h	10 × 2 h	intermédiaire	projet
DURÉE	SÉANCES	NIVEAU	FORMAT

Description

L'intégration et le déploiement continus (CI/CD) sont au cœur des pratiques DevOps : ils réduisent les erreurs, accélèrent les livraisons et rendent chaque mise en production banale et réversible. Toute équipe de développement moderne attend de ses membres qu'ils sachent lire, écrire et réparer un pipeline.

Les participants construisent **le pipeline complet d'une application** étape par étape : compilation, tests, analyse de qualité, construction d'une image Docker, publication, puis déploiement sur un serveur. Le pipeline est d'abord écrit avec **GitHub Actions**, puis comparé à **GitLab CI**.

Public visé

- Développeurs qui veulent automatiser leurs livraisons
- Administrateurs systèmes et futurs ingénieurs DevOps
- Étudiants ayant suivi Git et Docker

Prérequis

- Maîtrise de Git (branches, pull requests)
- Bases de Docker (images, Dockerfile, Compose)
- Notions de ligne de commande Linux

À l'issue de la formation, vous saurez

☐	expliquer les principes de l'**intégration**, de la **livraison** et du **déploiement continus** ;
☐	écrire un **workflow GitHub Actions** : déclencheurs, jobs, étapes, exécuteurs ;
☐	automatiser la **compilation**, les **tests** et l'**analyse de qualité** du code ;
☐	construire et publier une **image Docker** depuis le pipeline ;
☐	**déployer** automatiquement sur un serveur, avec gestion des **secrets** et des environnements ;
☐	transposer un pipeline vers **GitLab CI** et dépanner un pipeline en échec.

Méthode pédagogique

- Courts exposés, puis construction **pas à pas d'un pipeline réel**.
- Échecs provoqués volontairement pour apprendre à lire les journaux d'exécution.
- Comparaison GitHub Actions / GitLab CI sur le même projet.
- Sécurité dès le départ : secrets, droits minimaux, branches protégées.

Programme — 10 séances

1 **Pourquoi la CI/CD** 2 h

Problèmes des livraisons manuelles, intégration, livraison et déploiement continus, place dans la démarche DevOps.

`DevOps` `CI` `CD`

2 **Premier workflow GitHub Actions** 2 h

Structure YAML, déclencheurs, jobs, étapes, exécuteurs hébergés, lecture des journaux.

`YAML` `on: push` `runners`

3 **Compiler et tester automatiquement** 2 h

Installer l'environnement du langage, mettre en cache les dépendances, lancer les tests à chaque push.

`build` `tests` `cache`

4 **Qualité et protection des branches** 2 h

Analyse statique, formatage, couverture de tests, contrôles obligatoires avant fusion.

`lint` `couverture` `status checks`

5 **Construire et publier une image Docker** 2 h

Construire l'image dans le pipeline, l'étiqueter selon la version, la pousser sur un registre.

`docker build` `registre` `tags`

6 **Secrets et variables** 2 h

Secrets du dépôt et des environnements, variables, bonnes pratiques pour ne jamais exposer un mot de passe.

`secrets` `variables` `sécurité`

7 **Déployer sur un serveur** 2 h

Déploiement par SSH et Docker Compose, environnements de recette et de production, approbation manuelle.

`SSH` `environnements` `approbation`

8 **GitLab CI** 2 h

Le même pipeline en .gitlab-ci.yml : étapes, jobs, artefacts, exécuteurs, différences avec GitHub Actions.

`GitLab CI` `stages` `artefacts`

9 **Dépanner et fiabiliser** 2 h

Analyser un échec, relancer, rendre un pipeline rapide et fiable, notifications et retour arrière.

`dépannage` `rollback` `notifications`

10 **Projet final et évaluation** 2 h

Mettre en place le pipeline complet d'une application fournie, puis soutenance.

`projet` `évaluation /20`

Évaluation et validation

- Pipelines vérifiés à chaque séance (exécutions réussies).
- **Projet final** : pipeline complet, du push au déploiement, présenté et noté sur 20.
- Attestation de formation délivrée par OSD — Open Digital Service.

Modalités

Durée	20 heures, en 10 séances de 2 heures
Format	présentiel en salle informatique, ou intra-entreprise / établissement
Effectif	groupes de 8 à 20 participants
Matériel	un poste par participant avec Git, Docker, VS Code, un compte GitHub et un accès à un serveur de test
Supports remis	fiches de séance, modèles de workflows commentés, dépôts d'exercices
Formateur	Mohammed Slimani — architecte DevOps, ancien administrateur systèmes et responsable informatique

Intéressé(e) par cette formation ?

Programme adaptable (GitLab, Jenkins, langage de l'application, volume horaire). Contact : contact@osdigitalis.com · WhatsApp +213 674 05 38 95 · osdigitalis.com

Open Digital Service

Mohammed Slimani — Formateur & consultant

+213 (0)6 74 05 38 95 · contact@osdigitalis.com · msslimani@osdigitalis.com

osdigitalis.com · LinkedIn : in/mohammed-slimani-osdigitalis