OSD
OPEN DIGITAL SERVICE

Scannez-moi

WHATSAPP
+213 674 05 38 95

SITE WEB
osdigitalis.com

FICHE FORMATION · OSD

Automatisation avancée de pipelines DevOps

Workflows réutilisables, infrastructure automatisée, qualité, sécurité et releases

20 heures | 10 séances | niveau avancé | Ansible · Actions

Une formation pour **passer d'un pipeline qui fonctionne à une chaîne de livraison industrielle** : workflows réutilisables, exécuteurs privés, configuration automatisée des serveurs, contrôles de qualité et de sécurité, versions publiées automatiquement.

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

Description

Un premier pipeline résout le cas d'un projet. Une équipe qui gère plusieurs applications et plusieurs serveurs a besoin de plus : des briques réutilisables, des serveurs configurés automatiquement, des contrôles de sécurité, des déploiements progressifs et une supervision qui alerte avant les utilisateurs.

Les participants industrialisent **la chaîne de livraison de plusieurs applications** : modèles de workflows partagés, exécuteur auto-hébergé, provisionnement des serveurs avec **Ansible**, analyse de qualité et de vulnérabilités, publication automatique des versions et supervision avec **Prometheus** et **Grafana**.

Public visé

- Développeurs et administrateurs ayant déjà écrit un pipeline CI/CD
- Ingénieurs DevOps juniors
- Responsables techniques qui veulent standardiser les livraisons

Prérequis

- Formation « Pipeline CI/CD » ou expérience équivalente
- Maîtrise de Git et de Docker
- Bases d'administration Linux et de scripts shell

À l'issue de la formation, vous saurez

☐	factoriser les pipelines avec des **workflows réutilisables**, des actions composites et des **matrices** ;
☐	installer et sécuriser un **exécuteur auto-hébergé** ;
☐	configurer des serveurs de façon reproductible avec **Ansible**, déclenché depuis le pipeline ;
☐	intégrer des contrôles de **qualité** et de **sécurité** (analyse de code, dépendances, images) ;
☐	automatiser le **versionnement** et la publication des releases ;
☐	mettre en place des **déploiements progressifs** et la **supervision** de la chaîne.

Méthode pédagogique

- Courts exposés, puis ateliers sur une **chaîne de livraison fil rouge** à plusieurs dépôts.
- Approche « tout est code » : pipelines, configuration des serveurs et tableaux de bord versionnés.
- Études de cas réels de pannes de production et de failles dans des chaînes de livraison.
- Sécurité de la chaîne d'approvisionnement abordée à chaque étape.

Programme — 10 séances

1 **De la CI/CD à la chaîne de livraison** 2 h

Limites d'un pipeline isolé, indicateurs de performance des équipes, panorama des outils, présentation de la chaîne fil rouge.

`DevOps` `indicateurs` `chaîne`

2 **Workflows réutilisables et matrices** 2 h

Workflows appelables, actions composites, matrices de versions et de systèmes, factorisation entre dépôts.

`reusable workflows` `matrices` `composite`

3 **Exécuteurs auto-hébergés** 2 h

Installer un exécuteur sur un serveur Linux, l'isoler dans des conteneurs, le sécuriser et le superviser.

`self-hosted` `isolation` `sécurité`

4 **Automatiser les serveurs avec Ansible** 2 h

Inventaire, playbooks, rôles, idempotence : préparer un serveur Docker de façon reproductible.

`Ansible` `playbooks` `rôles`

5 **Ansible dans le pipeline** 2 h

Déclencher Ansible depuis le pipeline, coffre de secrets, environnements multiples.

`Ansible Vault` `environnements` `déploiement`

6 **Qualité et sécurité du code** 2 h

Analyse de qualité (SonarQube), dépendances vulnérables, secrets oubliés dans le code, seuils bloquants.

`SonarQube` `dépendances` `secrets`

7 **Sécurité des images et de la chaîne** 2 h

Analyse des images (Trivy), signature, nomenclature logicielle (SBOM), droits minimaux des pipelines.

`Trivy` `SBOM` `signature`

8 **Versions et releases automatiques** 2 h

Versionnement sémantique, messages de commit conventionnels, notes de version et publication automatiques.

`SemVer` `releases` `changelog`

9 **Déploiements progressifs et supervision** 2 h

Déploiement bleu-vert et canari, contrôles de santé, retour arrière automatique, métriques Prometheus et alertes Grafana.

`blue-green` `Prometheus` `Grafana`

10 **Projet final et évaluation** 2 h

Industrialiser la chaîne de livraison d'un ensemble d'applications fourni, puis soutenance technique.

`projet` `évaluation /20`

Évaluation et validation

- Ateliers vérifiés à chaque séance (exécutions et configurations versionnées).
- **Projet final** : chaîne de livraison industrialisée, présentée en soutenance technique et notée 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, et l'accès à deux serveurs Linux de test (VM ou VPS)
Supports remis	fiches de séance, modèles de workflows et de rôles Ansible, dépôts d'exercices
Formateur	Mohammed Slimani — architecte DevOps, ancien administrateur systèmes et responsable informatique

Intéressé(e) par cette formation ?

Programme adaptable (GitLab CI, Jenkins, Terraform, Kubernetes, 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