

# GitLab CI/CD

3 j (21 heures)

Ref : GITLAB

## Public

Développeurs, chefs de projet, administrateurs systèmes, architectes.

## Pré-requis

Connaissances de base des commandes Linux  
Connaissances de base de la gestion de versions avec Git

## Moyens pédagogiques

Formation réalisée en présentiel ou à distance selon la formule retenue  
Alternance d'exposés et de cas pratiques, synthèse  
Un poste informatique par stagiaire connecté à internet, à une imprimante en réseau et au réseau informatique  
Les salles sont équipées d'un tableau interactif ou d'un vidéoprojecteur et d'un paperboard  
Support de cours fourni à chaque stagiaire

## Modalités de suivi et d'évaluation

Feuille de présence émargée par demi-journée par les stagiaires et le formateur  
Exercices de mise en pratique ou quiz de connaissances tout au long de la formation permettant de mesurer la progression des stagiaires  
Questionnaire d'évaluation de la satisfaction en fin de stage  
QCM d'évaluation des connaissances en fin de formation  
Attestation de fin de formation

## Objectifs

- Connaître l'offre GitLab
- Pratiquer la gestion de versions avec Git et collaborer avec GitLab
- Mettre en place l'intégration continue (CI) et le déploiement continu (CD) avec GitLab
- Appréhender les éléments constitutifs d'une usine logicielle DevOps

## Programme détaillé

### RAPPELS GIT

---

- Git : un système de versionning distribué
- Avantages par rapport aux systèmes centralisés

## **LA SOLUTION GITLAB**

---

Introduction, Releases disponibles  
Architecture, modèle de données  
Les rôles et la gestion des droits  
Installation de GitLab  
Découverte de l'interface, fonctionnalités d'administration  
Mise en place de dépôt  
Configuration client, configuration des accès

## **FONCTIONNALITES DE BASE DE GITLAB**

---

Notion de projet dans GitLab  
README.md, Wiki, Snippets...  
Gestion des branches et des tags  
Côté client : Synchronisation avec le serveur, Clone de dépôt  
Visualisation historique et activité

## **WORKFLOWS DE COLLABORATION**

---

GitFlow et GitLab Flow  
Gestion des issues  
Configuration des notifications  
Branches et tags protégés  
Releases, Milestone  
Merge request

## **PIPELINES GITLAB CI/CD (GITLAB CONTINUOUS INTEGRATION & DELIVERY)**

---

Pipelines, pipelines typiques DevOps, .gitlab-ci.yml  
Jobs et Runners, typologie des runners, l'apport de Docker  
Variables  
Environnement et déploiements  
Cache et artefacts  
Contrôle du flow  
Include et gabarits  
Services Docker

## **LES TESTS DANS LA PIPELINE**

---

Exploitation des formats JUnit  
Tests d'accessibilité, performance du navigateur  
Qualité de code  
Tests de charge

Tableau de bord de métriques

## **DEPOT DE PACKAGES**

---

Rôle du dépôt de package dans une pipeline DevOps

Format supporté par Gitlab

Registres d'image Docker

## **INTEGRATION KUBERNETES**

---

Apport de l'orchestrateur de conteneurs

Configuration de cluster

IaaS avec Terraform

Tableaux de bord de déploiement

---