

## Formation Intégration et déploiement continue avec Gitlab-CI

<b>Durée :</b>	5 jours
<b>Public :</b>	Opérationnels, Développeurs, Chefs de projets
<b>Pré-requis :</b>	Connaissance du cycle de vie d'une application, maîtrise des commandes de base Git
<b>Objectifs :</b>	Mettre en oeuvre et exploiter un serveur d'intégration continue. Gérer les interconnexions avec un système de build et de tests
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	DEV101596-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Comprendre l'intégration continue et découvrir GitLab

Processus de développement, tests unitaires / d'intégration  
Intégration continue : présentation, positionnement dans une démarche agile  
Gestion des environnements : développement, recette, production  
Outils de conteneurs applicatifs (Docker)  
Configurations système et applicative et outils de centralisation (Puppet, Ansible)  
Panorama outils de gestion : versionnement, build, tests, qualité  
GitLab-CI : présentation, fonctionnalités  
Types d'installation  
Notion de projet, documentation (README.md, Wiki, ...)

**Atelier : Mise en place de GitLab, tour d'horizon de l'interface, création de dépôts et paramétrage**

### Maîtriser les bases du YAML

YAML : syntaxe de base, spécificités  
Déclaration et utilisation de variables  
Collections  
Ancres

### Gérer des builds avec GitLab CI

Principe de fonctionnement : pipelines, stages, tasks, artefacts, tags  
Structure d'un build de projets, le fichier manifeste .gitlab-ci.yml  
Jobs et Runners, utilisation de Docker  
Mise en place de builds : automatiques / manuels  
Plug-ins pour la gestion des dépôts de source

Outils de build : Maven, Gradle, ...  
Organisation des branches et des tags  
Gestion des dépendances et dépôts, mise en place d'un cache  
Intégration des dépôts avec les outils de build  
Gestion des notifications  
Création et utilisation de variables dans les paramètres CI/CD  
Lancement de jobs en parallèle

**Atelier : Interfaçage avec des dépôts de dépendances - Configuration et lancement de builds (applications web JS ou services Java)**

### **Contrôler la qualité du code**

Présentation, gestion de la qualité du code  
Panorama des outils : Checkstyle, FindBugs, ...  
Rapport de qualité : configuration, plugins (Violations)  
Autres rapports : complexité, tâches, ...

**Atelier : Intégration d'outils de gestion de qualité du code (SonarQube) dans une démarche d'intégration continue**

### **Automatiser les tests**

Types de tests  
Automatisation, couverture  
Tests unitaires et d'intégration  
Tests d'acceptance, tests de performances  
Optimisation des tests

**Atelier : Multiples scénarios d'automatisation de tests unitaires, d'intégration, de performances**

### **Mettre en place une stratégie de déploiement**

Stratégie globale d'automatisation  
Scripts de déploiement et de mise à jour  
Rollbacks

Gestion des artefacts (archivage)  
Utilisation des groupes de ressources pour limiter la concurrence

**Atelier : Construction de scripts de déploiement**

### **Administrer les outils**

Sécurité du serveur d'intégration continue  
Gestion des utilisateurs : bases, rôles, autorisations  
Gestion des journaux  
Espace mémoire/charge CPU, espace disque  
Monitoring

**Atelier : Multiples tâches d'administration du serveur**