



# *Plan de Formation*

## *GitLab*

## ***Description de la formation***



**GitLab a pénétré le monde de l'entreprise en offrant une solution Web permettant la gestion des référentiels Git. Fort de son implantation, Gitlab est devenu Gitlab CI en ajoutant diverses fonctionnalités : gestion d'issues, tableaux de bord de projet, pipeline CI/CD, intégration cloud et conteneurs, outils de tests, d'analyse etc. Gitlab CI apparaît aujourd'hui comme une plate-forme DevOps complète rassemblant tous les acteurs du projet : métier, développeurs, exploitants.**



## ***Description de la formation***



Cette formation GitLab CI/CD détaille toutes les préoccupations d'un projet DevOps. Elle commence par les expressions métiers, la planification et le suivi via l'utilisation des issues ; elle expose ensuite les différents patterns de collaboration autour de branches Git.

Puis, elle détaille en profondeur le fonctionnement et la mise en place de pipeline CI/CD, l'intégration avec les dépôts d'artefacts, les environnements et les outils de test.





## Objectifs

### Opérationnel

**Savoir maîtriser les fonctionnalités de l'outil GitLab : mise en place de dépôts, workflows de collaboration et intégration à une CI.**

### Pédagogiques

**À l'issue de cette formation GitLab CI/CD, vous aurez acquis les connaissances et compétences nécessaires pour :**

## *GitLab*

- Configurer un projet GitLab, rôles et membres, tableau de bord des issues, dépôt de sources
- Mettre en place des workflows de collaboration autour de branches GitLab
- Mettre en place des pipelines via le descripteur.gitlab-ci.yml
- Intégrer des dépôts d'artefacts : Maven, Docker, npm
- Définition des environnements et Intégration cloud et Kubernetes
- Intégration outils de test : Unitaire/intégration, Fonctionnels, Sécurité, Qualité

## ***A qui s'adresse cette formation ?***

### **Public**

---

**Ce cours GitLab CI/CD s'adresse à des utilisateurs de la solution GitLab qu'ils soient développeurs, chefs de projets, architectes, reviewers, administrateurs**



### **Prérequis**

---

**Pour suivre cette formation GitLab CI/CD, des notions sur Git sont nécessaires.**

## Contenu du cours

### Rappels sur Git

- Git : un système de versionning distribué
- Gestion des commits et bonnes pratiques
- Maîtriser l'usage des branches et des bonnes pratiques

### La solution GitLab

- Fonctionnalités et versions disponibles
- Architecture, modèle de données
- Découverte de l'interface et des éléments de base
- Configuration client, configuration des accès et des notifications
- Découverte de la configuration de la plate-forme école

### Travaux Pratiques :

Parcours de l'interface et des projets proposés

## Contenu du cours

### Fonctionnalités de base de GitLab

- Notion de projet et groupe dans GitLab
- Gestion des droits d'accès : visibilité, rôles
- Le fichier README.md et les bonnes pratiques
- Gestion des branches et des tags
- Visualisation de l'historique et de l'activité
- Création de projets et templates de projets
- L'utilisation des badges

### Travaux Pratiques :

Manipulation des sources d'un projet

## Contenu du cours

### Workflows de collaboration

- GitFlow et GitLab Flow
- Gestion des issues
- Configuration des notifications
- Branches et tags protégés
- Releases, Milestones
- Le processus de merge request

### Travaux Pratiques :

Travail en équipe, répartition des rôles, cycle de production d'une release

## Contenu du cours

### Les éléments clés de l'architecture CI/CD de GitLab

- Pipelines et workflows
- Les éléments clés du pipeline
- Runners et typologie, apports des technologies de conteneurisation
- Tags de runner

### Pipelines GitLab CI/CD (GitLab Continuous Integration & Delivery)

- Rôle et fonctionnement du fichier `.gitlab-ci.yml`
- Cache et artefacts
- Contrôle du flow : `only/except` et `rules`
- `include` et gabarits
- Environnement et déploiements
- Pipelines multi-projets
- Utiliser les spécificités de Docker dans un pipeline

### Travaux Pratiques :

Définition d'une pipeline de déploiement continu

## Contenu du cours

### Utilisation avancée du pipeline

- Artefacts, tests unitaires et exploitation des formats JUnit
- Utilisation des built-in templates de Gitlab : accessibilité, performance, qualité

### Travaux Pratiques :

Ajout des tests automatisés dans la pipeline

### Dépôt de packages

- Industrialiser la production de packages avec Gitlab : l'exemple de Python et PHP
- Industrialiser la production d'images Docker avec Gitlab : problématiques et mise en place

### Travaux Pratiques :

Automatisation de la construction d'une image Docker, mise à disposition dans la registry et utilisation

### Intégration de GitLab et Kubernetes

- Apport de l'orchestrateur de conteneurs
- Les apports de Podman
- Architecture de la CI/CD avec Kubernetes



Knowledge forges Empires

## Nos Contacts



**[empire-training.tn](http://empire-training.tn)**



**+216 55 826 628**  
**+216 74 201 616**



**16 Rue D'Athènes, Sfax**



**[Contact@empire-training.tn](mailto:Contact@empire-training.tn)**