

TP2 PROJET FIL ROUGE

Architecture CI/CD Hybride, Automatisation Jenkins & Infrastructure as Code (Terraform) — Suite du TP1

Auteur : ALIMOU DIALLO (Santou AI Lab) | **Module :** CICD
Projet : Projet Fil Rouge - Back-End & DevOps | **Date :** Septembre 2026

1. ARCHITECTURE DE LA CHAÎNE CI/CD HYBRIDE

Faisant suite aux bases de conteneurisation établies lors du TP1, ce second volet formalise une architecture hybride et modulaire séparant les responsabilités entre les différents moteurs d'automatisation pour industrialiser les processus de déploiement :

- **Jenkins** : Orchestrateur central pour la compilation, l'exécution des batteries de tests unitaires, l'analyse statique de code via SonarQube et la publication initiale des artefacts conteneurisés sur Docker Hub.
- **GitLab CI** : Chargé de la récupération et de la validation des tags d'images prêts pour la production, et du déclenchement de l'infrastructure as code.
- **Terraform** : Responsable du provisionnement de l'infrastructure cible, de la configuration sécurisée et de la synchronisation des déploiements.

Outil / Composant	Rôle Principal	Environnement Cible
Jenkins	Build, Tests Unitaires, Analyse SonarQube & Push Docker Hub	Serveur CI Interne
GitLab CI	Validation de branches, Déclenchement Terraform	GitLab Runner / Cloud
Terraform	Provisionnement distant & Orchestration Docker Compose	Serveur de Production

2. AUTOMATISATION DES PIPELINES ET CONFORMITÉ QUALITÉ

2.1. Pipeline Jenkins (Build, Tests & Analyse SonarQube)

Le pipeline Jenkins implémente un contrôle qualité strict. Chaque commit déclenche séquentiellement :

1. **Build & Compilation** : Résolution des dépendances et compilation du code source.
2. **Tests Unitaires** : Exécution de la suite de tests avec validation d'intégrité.
3. **Analyse SonarQube** : Mesure de la dette technique et vérification impérative du seuil de couverture de code (Coverage $\geq$ 80%).
4. **Publish Docker Hub** : Construction de l'image optimisée et envoi vers le registre.

2.2. Variables d'Environnement GitLab CI

Pour automatiser les étapes via GitLab CI et Terraform, les variables suivantes doivent être configurées dans les paramètres du projet (Settings > CI/CD > Variables) :

- **DOCKER_HUB_USERNAME** : Identifiant du registre Docker Hub.
- **DOCKER_HUB_ACCESS_TOKEN** : Jeton d'accès sécurisé.
- **SSH_PRIVATE_KEY** : Clé privée SSH au format PEM pour la connexion au serveur.
- **SERVER_IP** : Adresse IP du serveur cible.
- **TF_VAR_mysql_root_password & TF_VAR_mysql_password** : Mots de passe pour les bases de données.

3. LIVRABLES ET RAPPORT D'ANALYSE

Les livrables attendus pour ce TP incluent :

- **Dépôt GitLab** : Code structuré, incluant les configurations Terraform et les fichiers de pipeline.
- **README.md** : Documentation claire des étapes et des prérequis.
- **Rapport d'analyse SonarQube** : Preuve de la validation du Quality Gate avec un taux de couverture supérieur à 80% et absence de vulnérabilités bloquantes.
- **Application Déployée** : Conteneurs opérationnels et vérifiés sur le serveur cible.