

Documentation : Installation et Configuration de GitLab avec Intégration GitHub

1. Installation de GitLab sur Ubuntu 22.04

Étape 1 : Mettre à jour le système et Configurer le pare-feu

```
sudo apt update && sudo apt upgrade -y
```

```
sudo ufw allow http  
sudo ufw allow https  
sudo ufw allow OpenSSH  
sudo ufw reload  
sudo ufw enable
```

Étape 2 : Installer les dépendances nécessaires

```
sudo apt install -y curl openssh-server ca-certificates tzdata perl
```

Étape 3 : Ajouter le dépôt GitLab

```
curl -fsSL https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash
```

Étape 4 : Installer GitLab

```
sudo apt install -y gitlab-ce
```

✅ Étape 5 : Configurer GitLab

Modifier l'URL externe dans le fichier de configuration GitLab :

```
sudo nano /etc/gitlab/gitlab.rb
```

Remplacez la ligne suivante :

```
external_url 'http://192.168.193.130'
```

Appliquer la configuration :

```
sudo gitlab-ctl reconfigure
```

✅ Étape 6 : Vérifier le statut de GitLab et Accéder à l'interface Web

```
sudo gitlab-ctl status  
http://192.168.193.130  
sudo cat /etc/gitlab/initial_root_password
```

🔑 2. Configuration des Clés SSH

✅ Étape 1 : Générer une clé SSH (si ce n'est pas déjà fait)

```
ssh-keygen -t rsa -b 4096 -C "votre-email@example.com"
```

Appuyez sur **Entrée** pour accepter le chemin par défaut.

✅ Étape 2 : Ajouter la clé SSH à l'agent SSH

```
eval $(ssh-agent -s)  
ssh-add ~/.ssh/id_rsa
```

✅ Étape 3 : Ajouter la clé SSH à GitLab

Afficher la clé publique :

```
cat ~/.ssh/id_rsa.pub
```

1. Copiez la clé.
2. Allez sur **GitLab** → [Profile Settings](#) → [SSH Keys](#) .
3. Collez la clé et cliquez sur **Add Key**.

✅ Étape 4 : Tester la connexion SSH avec GitLab

```
ssh -T git@192.168.193.130
```

Vous devriez voir :

```
Welcome to GitLab, @votre-utilisateur!
```

🔑 3. Configuration des Tokens d'Accès GitLab

✅ Étape 1 : Générer un Token GitLab

1. Aller dans **GitLab** → [User Settings](#) → [Access Tokens](#) .
2. Créer un nouveau token avec ces permissions :
 - [api](#)
 - [read_repository](#)
 - [write_repository](#)
3. Copier le token généré.

✅ Étape 2 : Tester l'accès API GitLab

```
TOKEN_GITLAB="glpat_XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
GITLAB_URL="http://192.168.193.130"
```

```
curl --header "PRIVATE-TOKEN: $TOKEN_GITLAB" "$GITLAB_URL/api/v4/projects"
```

4. Connexion GitLab avec un Compte GitHub

✓ Étape 1 : Créer une Application OAuth GitHub

1. Aller sur [GitHub Developer Settings](#).
2. Créer une nouvelle application OAuth avec :
 - **Homepage URL** : `http://192.168.193.130`
 - **Authorization callback URL** : `http://192.168.193.130/users/auth/github/callback`
3. Copier le **Client ID** et **Client Secret** générés.

✓ Étape 2 : Activer GitHub OAuth dans GitLab

Modifier le fichier de configuration GitLab :

```
sudo nano /etc/gitlab/gitlab.rb
```

Ajouter ces lignes :

```
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = ['github']
gitlab_rails['omniauth_auto_sign_in_with_provider'] = 'github'
gitlab_rails['omniauth_providers'] = [
  {
    "name" => "github",
    "app_id" => "VOTRE_CLIENT_ID",
    "app_secret" => "VOTRE_CLIENT_SECRET",
    "args" => { "scope" => "user:email" }
  }
]
```

Appliquer la configuration :

```
sudo gitlab-ctl reconfigure
```

5. Importation des Projets GitHub vers GitLab

✓ Étape 1 : Lister les repositories GitHub

```
TOKEN_GITHUB="ghp_XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

```
curl -H "Authorization: token $TOKEN_GITHUB" \  
  -H "Accept: application/vnd.github.v3+json" \  
  https://api.github.com/user/repos?per_page=100 | jq -r '.[].clone_url'
```

✓ Étape 2 : Importer un projet GitHub spécifique dans GitLab

Exemple pour `testpipeline` :

```
GITHUB_REPO="https://github.com/motrabelsi10/testpipeline.git"  
GITLAB_REPO="git@192.168.193.130:motrabelsi10/testpipeline.git"
```

```
git clone --mirror "$GITHUB_REPO"  
cd testpipeline.git  
git remote remove origin  
git remote add origin "$GITLAB_REPO"  
git push --mirror origin
```

✓ Étape 3 : Vérifier l'importation sur GitLab

```
git ls-remote git@192.168.193.130:motrabelsi10/testpipeline.git
```

Si la sortie est vide, vérifier :

- **Les permissions du token GitLab**
- **L'URL du remote**
- **Les logs GitLab avec** `sudo gitlab-ctl tail`

6. Configuration Git avec Joget

1. Aller dans Joget → Settings → Git Integration

2. Entrer l'URL du dépôt GitLab :

```
http://192.168.193.130/motrabelsi10/joget-app.git
```

3. Ajouter le nom d'utilisateur GitLab : motrabelsi10

4. Utiliser le Token GitLab généré précédemment

5. Cocher les options :

☒ Always Pull from Remote Server

☒ Auto Sync DB From Git

6. Enregistrer la configuration et tester la connexion

7. Modifier le Jenkinsfile pour GitLab

Dans le pipeline Jenkins, modifier le Checkout Code :

```
stages {
  stage('Checkout Code') {
    steps {
      script {
        echo "🚀 Cloning repository: main"
      }
      git branch: 'main', url: 'http://192.168.193.130/motrabelsi10/yoyo.git'
    }
  }
}
```

 **GitLab est maintenant prêt pour être utilisé avec Joget et Jenkins !**