

TP2 PROJET FIL ROUGE

Trame de Travail : Récupération Docker Hub & Déploiement Automatisé (Terraform & GitLab CI)

Formateur : ALIMOU DIALLO (Santou AI Lab) | **Module :** CICD

Objectif : Configurer les variables GitLab CI, structurer Terraform et automatiser le pipeline.

ÉTAPE 1 : CONFIGURATION DES VARIABLES GITLAB CI/CD

Rendez-vous dans votre dépôt GitLab sous **Settings > CI/CD > Variables** et configurez les variables masquées et protégées indispensables pour l'authentification et l'infrastructure :

- **DOCKER_USER** : santoudllo54242 (Votre nom d'utilisateur Docker Hub)
- **DOCKER_PASS** : Votre mot de passe ou Personal Access Token (PAT) Docker Hub
- **SERVER_IP** : L'adresse IP de votre serveur de déploiement cible
- **SSH_USER** : L'utilisateur de connexion (ex: `root` ou `ubuntu`)
- **SSH_PRIVATE_KEY** : Votre clé privée SSH (contenu de `cat ~/.ssh/id_rsa`)
- **MYSQL_ROOT_PASSWORD** : Mot de passe root pour la base de données
- **MYSQL_PASSWORD** : Mot de passe utilisateur pour la base de données

ÉTAPE 2 : INITIALISATION DU DOSSIER TERRAFORM

Dans votre projet, créez un dossier `terraform/` contenant les deux fichiers fondamentaux suivants :

1. Le fichier `variables.tf` (Déclaration des variables)

```
variable "server_ip" {
  type      = string
  description = "Adresse IP du serveur cible"
}

variable "ssh_user" {
  type      = string
  description = "Utilisateur SSH pour la connexion"
}

variable "ssh_private_key" {
  type      = string
  description = "Clé privée SSH pour le provisionnement"
  sensitive = true
}

variable "mysql_root_password" {
  type      = string
  description = "Mot de passe root MySQL"
  sensitive = true
}

variable "mysql_password" {
  type      = string
  description = "Mot de passe applicatif MySQL"
```

```
sensitive    = true
}
```

2. Le fichier `main.tf` (Configuration de l'infrastructure)

```
terraform {
  required_version = ">= 1.0.0"
  required_providers {
    local = {
      source  = "hashicorp/local"
      version = "~> 2.0"
    }
  }
}

provider "local" {}

# Exemple d'utilisation pour générer un fichier de configuration ou préparer le déploiement distant
resource "local_file" "deployment_config" {
  filename = "${path.module}/config_env.txt"
  content  = <
```

Note : Récupérez votre clé publique locale avec la commande `cat ~/.ssh/id_rsa.pub` pour l'ajouter sur votre serveur distant si ce n'est pas déjà fait.

ÉTAPE 3 : ÉCRITURE DU PIPELINE GITLAB CI (STAGES BUILD & INFRA)

Créez le fichier `.gitlab-ci.yml` à la racine de votre dépôt avec les deux stages principaux :

```
stages:
  - build
  - infra

variables:
  DOCKER_IMAGE: "$DOCKER_USER/projet-fil-rouge:latest"

docker_build:
  stage: build
  image: docker:latest
  services:
    - docker:dind
  script:
    - echo "Connexion à Docker Hub..."
    - echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin
    - echo "Construction et push de l'image..."
    - docker build -t $DOCKER_IMAGE .
    - docker push $DOCKER_IMAGE
```

```

only:
  - main

terraform_deploy:
  stage: infra
  image:
    name: hashicorp/terraform:latest
    entrypoint: [""]
  script:
    - cd terraform
    - echo "Initialisation de Terraform..."
    - terraform init
    - echo "Planification du déploiement..."
    - terraform plan -auto-approve -var="server_ip=$SERVER_IP" -
var="ssh_user=$SSH_USER" -var="ssh_private_key=$SSH_PRIVATE_KEY" -
var="mysql_root_password=$MYSQL_ROOT_PASSWORD" -
var="mysql_password=$MYSQL_PASSWORD"
    - echo "Application de l'infrastructure..."
    - terraform apply -auto-approve -var="server_ip=$SERVER_IP" -
var="ssh_user=$SSH_USER" -var="ssh_private_key=$SSH_PRIVATE_KEY" -
var="mysql_root_password=$MYSQL_ROOT_PASSWORD" -
var="mysql_password=$MYSQL_PASSWORD"
  only:
    - main

```