TP Linux - Automatisation des processus de développement, tests et déploiement

Objectif : Apprendre à automatiser des processus de développement, de tests et de déploiement à l'aide de scripts Bash, Git, et d'outils basiques pour garantir une livraison rapide et fiable des applications.

Partie 1 : Initialisation du projet et gestion de version avec Git

1. Initialisation d'un dépôt Git :

Créez un répertoire pour le projet et initialisez un dépôt Git :

```
mkdir tp_bash_automatisation && cd tp_bash_automatisation

git init
```

2. Création d'un simple fichier HTML :

Créez une page index.html qui sera déployée par la suite :

```
<!DOCTYPE html>

<html lang="fr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,
initial-scale=1.0">

<title>Application Automatisée</title>

</head>

<body>

<h1>Hello World, Automatisation avec Bash !</h1>

</body>

</html>
```

3. Ajout des fichiers dans le dépôt Git :

Ajoutez et committez le fichier HTML :

```
git add index.html

git commit -m 'Ajout de la page HTML initiale'
```

4. Création d'une branche de développement :

```
git checkout -b dev
```

Partie 2 : Automatisation des tests avec Bash

1. Création d'un script de vérification de syntaxe HTML :

Créez un script Bash test_html.sh pour vérifier que le fichier index.html est bien présent et valide :

```
#!/bin/bash

if [ -f "index.html" ]; then

echo "Le fichier index.html existe."

if grep -q "<h1>" index.html; then

echo "Test réussi : Balise <h1> trouvée."

else

echo "Test échoué : Balise <h1> non trouvée."

exit 1

fi

else

echo "Test échoué : Le fichier index.html n'existe pas."

exit 1

fi
```

2. Rendre le script exécutable et l'exécuter :

Rendez le script exécutable et exécutez-le :

```
chmod +x test_html.sh

./test_html.sh
```

3. Automatisation avec un script global :

Créez un script global run_tests.sh pour exécuter automatiquement tous les tests :

```
#!/bin/bash

echo "Lancement des tests..."

./test_html.sh
```

4. Ajout du script au dépôt Git :

Ajoutez les scripts de test et faites un commit :

```
git add test_html.sh run_tests.sh

git commit -m 'Ajout des scripts de tests automatiques'
```

Partie 3 : Déploiement automatique sur un serveur web local

1. Mise en place d'un serveur web local (Apache ou Nginx) :

Installez Apache ou Nginx sur votre machine si ce n'est pas déjà fait :

```
sudo apt update

sudo apt install apache2 -y # Pour Apache
```

2. Script de déploiement automatique :

Créez un script deploy.sh qui copie les fichiers HTML dans le répertoire du serveur web :

```
#!/bin/bash

echo "Déploiement de l'application..."

sudo cp index.html /var/www/html/

sudo systemctl restart apache2
```

3. Exécuter le script de déploiement :

Rendez le script exécutable et lancez-le :

```
chmod +x deploy.sh

./deploy.sh
```

4. Vérification du déploiement :

Ouvrez un navigateur et allez à l'adresse http://localhost pour voir si la page HTML a bien été
déployée.

Partie 4 : Automatisation complète avec Git Hooks et CI

1. Automatisation des tests avec des hooks Git :

Créez un hook Git pre-commit pour exécuter automatiquement les tests avant chaque commit :

```
nano .git/hooks/pre-commit

#!/bin/bash

echo "Exécution des tests avant le commit..."

./run_tests.sh

if [ $? -ne 0 ]; then

echo "Tests échoués. Commit annulé."

exit 1

fi
```

2. Simulation d'une intégration continue (CI) locale :

Pour simuler une CI locale, créez un script ci_pipeline.sh :

```
#!/bin/bash

echo "Démarrage du pipeline CI..."

./run_tests.sh

if [ $? -ne 0 ]; then

echo "Pipeline échoué : Les tests ont échoué."

exit 1

fi

./deploy.sh

echo "Pipeline terminé avec succès.
```