

TP9 Jenkins : Automatisation et Déploiement de Tests sur une Page HTML pour l'Entreprise Ynov

Objectifs

1. Installer et configurer Jenkins pour automatiser des tâches.
 2. Utiliser Jenkins pour déployer une page HTML mise à jour depuis un dépôt GitHub.
 3. Automatiser des tests :
 - Vérifier que la page mise à jour ne retourne pas d'erreurs (403, 404).
 - Si les tests réussissent, déployer la page sur le domaine principal `ynov.local`.
-

I. Étapes d'Installation et Prérequis

Référez-vous à la section **Installation de Jenkins** ci-dessus pour installer Jenkins sur une machine Linux, configurer Java et installer le plugin Git.

II. Structure du Workflow

1. **Dépôt GitHub :**
 - Le fichier `index.html` est maintenu dans un dépôt GitHub.
 - Toute mise à jour du fichier doit être testée avant d'être déployée.
 2. **Environnements Web :**
 - `test.ynov.local` : Un environnement de test où la page HTML mise à jour est déployée pour vérification.
 - `ynov.local` : L'environnement de production où la page HTML est déployée si tous les tests réussissent.
 3. **Tests Automatisés :**
 - Vérifiez que la page HTML sur `test.ynov.local` ne retourne pas d'erreurs (403, 404).
 - Déployez sur `ynov.local` uniquement si les tests réussissent.
-

III. Commandes et Explications pour Git et Jenkins

1. GitHub

1. Créer un dépôt Git :

- Sur GitHub, créez un dépôt nommé **ynov-html-deployment**.
- Ajoutez un fichier **index.html** avec un contenu de base.

2. Cloner le dépôt en local, modifier et pousser :

```
git clone
https://github.com/<votre-utilisateur>/ynov-html-deployment.git
cd ynov-html-deployment
echo "<!DOCTYPE html><html><head><title>Test
Ynov</title></head><body><h1>Mise à jour</h1></body></html>" >
index.html
git add index.html
git commit -m "Mise à jour de la page HTML"
git push origin main
```

2. Jenkins : Création d'un Job

Étape 1 : Déployer la page HTML sur **test.ynov.local**

1. Créer un Job Jenkins :

- Allez dans Jenkins > **New Item** > **Pipeline** et nommez-le **Deploy_Test**.

2. Configurer le Job :

- Ajoutez une étape pour cloner le dépôt Git :
 - Allez dans la section **Pipeline**.
 - Utilisez un script pour cloner le dépôt et déployer la page sur **test.ynov.local**.
- Exemple de pipeline Jenkins :

```
pipeline {
    agent any
    stages {
        stage('Cloner le dépôt') {
            steps {
                git
                'https://github.com/<votre-utilisateur>/ynov-html-deployment.git'
            }
        }
        stage('Déployer sur test.ynov.local') {
            steps {
                sh '''
                    cp index.html /var/www/test.ynov.local/
                '''
            }
        }
    }
}
```

```
pipeline {
    agent any

    stages {
        stage('Cloner le dépôt') {
            steps {
                git
                'https://github.com/<votre-utilisateur>/ynov-html-deployment.
                git'
            }
        }

        stage('Déployer sur test.ynov.local') {
            steps {
                sh '''
                cp index.html /var/www/test.ynov.local/
                '''
            }
        }
    }
}
```

```

    }

    stage('Tester la page sur test.ynov.local') {

        steps {

            sh '''

                STATUS=$(curl -o /dev/null -s -w
"%{http_code}" http://test.ynov.local/index.html)

                if [ $STATUS -eq 200 ]; then

                    echo "Page accessible, test réussi."

                else

                    echo "Erreur HTTP détectée : $STATUS"

                    exit 1

                fi

            '''

        }

    }

    stage('Déployer sur ynov.local') {

        steps {

            sh '''

                cp index.html /var/www/ynov.local/

            '''

        }

    }

}

}

```

3. Exécuter le Pipeline

1. Lancez le job Jenkins et observez les étapes.

2. Consultez les logs pour vérifier :
 - Si le déploiement sur `test.ynov.local` a été effectué.
 - Si les tests HTTP ont réussi.
 - Si le déploiement sur `ynov.local` a été finalisé.
-

IV. Résultats Attendus

1. La page `index.html` mise à jour depuis GitHub est d'abord déployée sur `test.ynov.local`.
2. Les tests HTTP (403, 404) vérifient que la page est accessible.
3. Si les tests réussissent, la page est déployée sur `ynov.local`.
4. Les logs Jenkins affichent clairement le statut de chaque étape.