TP7 : Création et Utilisation d'une Shared Library

Objectif

Création d'une Shared Library Jenkins et son utilisation dans un pipeline.

Prérequis

- Compte GitHub
- Accès administrateur à Jenkins
- Plugins Jenkins installés : Git, Pipeline

Étape 1 : Création de la structure de la Shared Library sur GitHub

1. Créez un nouveau dépôt GitHub nommé jenkins-shared-library
2. Clonez le dépôt localement
3. Créez la structure de fichiers suivante :

```
jenkins-shared-library/
├── vars/                    # Scripts globaux partagés
│   └── sayHello.groovy      # Exemple de script
├── src/                     # Code Groovy/Java
│   └── org/example/
│       └── Utils.groovy     # Classe utilitaire
└── README.md
```

4. Ajoutez du contenu aux fichiers :

vars/sayHello.groovy

```
def call(String name = 'World') {
    echo "Hello, ${name}!"
}
```

src/org/example/Utils.groovy

```
package org.example

class Utils {

    static String toUpperCase(String input) {

        return input.toUpperCase()

    }

}
```

5. Poussez les changements sur GitHub

Étape 2 : Configuration de la Shared Library dans Jenkins

1. Allez dans **Manage Jenkins** > **System Configuration** > **Configure System**
2. Descendez jusqu'à la section **Global Pipeline Libraries**
3. Ajoutez une nouvelle bibliothèque :
    - Nom : jenkins-shared-library
    - Default version : main (ou la branche que vous utilisez)
    - Vérification (retrieval method) : **Modern SCM**
    - Source Code Management : **Git**
    - URL du projet : https://github.com/votre-utilisateur/jenkins-shared-library.git
4. Sauvegardez la configuration

Étape 3 : Création d'un pipeline pour tester la Shared Library

Créez un nouveau pipeline (vous pouvez utiliser un Jenkinsfile ou le configurer directement dans l'interface Jenkins) :

```
@Library('jenkins-shared-library') _

pipeline {

    agent any

    stages {

        stage('Test Shared Library') {

            steps {

                script {

                    // Utilisation de la fonction globale

                    sayHello 'Jenkins'

                    sayHello 'Lab User'

                    // Utilisation de la classe utilitaire

                    def utils = new org.example.Utils()

                    def result = utils.toUpperCase('this should be uppercase')

                    echo "Uppercase result: ${result}"

                }

            }

        }

    }

}
```

Étape 4 : Exécution et validation

Exécutez le pipeline

Vérifiez dans les logs que :

- Les messages "Hello, Jenkins!" et "Hello, Lab User!" apparaissent
- Le message "THIS SHOULD BE UPPERCASE" apparaît