

Mise en application : Projet .NET Core avec Jenkins

Pour ce TD, vous allez devoir réaliser la mise en place d'un environnement d'intégration continue. Vous avez à votre disposition, un projet de développement qui sera accompagné des tests unitaires. Le projet est fonctionnel et le développement est terminé, mais on souhaiterait le mettre dans un environnement d'intégration continue afin de pouvoir vérifier les prochaines modifications qui seront apportées au projet.

Voici la liste des tâches que vous devez réaliser :

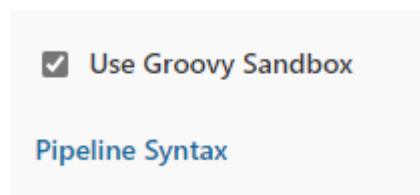
- Récupérer le projet et faire en sorte qu'il soit stocké sur une plateforme utilisant un système de gestion de version.
- Créer un job Jenkins qui va tester l'application. Il devra être lancé toutes les quinze minutes.
- Créer un job Jenkins qui va construire l'application en Debug. Le résultat de cette construction sera ensuite stocké dans un autre dossier qui simulera votre environnement de développement. Il devra être lancé toutes les heures.
- Créer un job Jenkins qui va construire l'application en Release. Le résultat de cette construction sera ensuite stocké dans un autre dossier qui simulera votre environnement de production. Il devra être lancé toutes les heures.
- Créer ensuite un Pipeline Jenkins qui aura 4 étapes :
 - Etape 1 : Récupération des données via le système de gestion de version
 - Etape 2 : Lancement des tests
 - Etape 3 : Construction pour l'environnement de développement
 - Etape 4 : Construction pour l'environnement de production

Plusieurs prérequis seront à installer sur la machine qui contiendra votre Jenkins.

Il faudra tout d'abord installer le framework .NET 5.0 car l'application est une application qui a pour cible ce framework. Voici les lignes de commandes à effectuer pour lancer les tests et construire l'application dans un environnement de développement :

- **dotnet test**
- **dotnet build --configuration Debug**

Pour le pipeline, pour récupérer la ligne à mettre dans le block "steps", il faut utiliser le raccourci "Pipeline Syntax" qui se trouve en bas de la page de configuration du pipeline.



Une liste déroulante s'offre à vous, et choisissez Git. On vous demandera alors vos identifiants de connexion à Git. Après avoir fait cela, vous pouvez cliquer sur "Generate Pipeline Syntax" et copier le code qui a été généré dans le block "steps".

Bonus :

Regardez comment générer un fichier de log lorsque les tests sont effectués puis mettre en avant sur la page du job de test puis sur le pipeline, le résultat de ces tests via le plugin Jenkins MSTest.