

## TP3 : Job Multi-configurations et pipeline Legacy

Dans ce TP, nous transformons le job freestyle des Tps précédents en un job Maven.

### Objectifs

- Créer son premier build Maven
- Faire une job multi-configuration
- Utilisation des paramètres
- Exécuter une pipeline Legacy

### 3.1 Job Maven et multi-modules

1. Installer le plugin *Maven Integration*
2. Créer un job Maven nommé *1\_weather\_unit* effectuant les cibles *clean package*, et publiant les tests
3. Exécuter manuellement le build et observer les résultats
4. Observer l'archivage automatique et les artefacts archivés, la configuration multi-modules

### 3.2 Jobs multi-configuration

Installer un deuxième JDK via une installation automatique.

Créer un second axe selon les nœuds esclaves à disposition

Créer un nouveau projet multi-configuration avec les 2 axes précédents à partir du job *1\_weather\_unit*

### 3.3 Chaînage de jobs

#### Configuration Maven et Jenkins

1. Récupérer la nouvelle version du projet et observer les changements dans le fichier *pom.xml* du répertoire *parent-project*
2. Créer un job Maven nommé *2\_weather\_integration* invoquant la cible Maven *integration-test* dans le profil *integration-Pintegration integration-test*
3. Chaîner ce build avec le build précédent, faire en sorte que les 2 builds soient sur le même repository Git

*Est-ce que cette configuration est satisfaisante ?*

### 3.4 Passage de paramètres

Objectif : Passage de la clé de hash Git entre *1\_weather\_unit* et les jobs

- Installer le plugin *Git Parameter Plugin*
- Définir un paramètre pour le job *1\_weather\_unit* dont la liste des valeurs est les différents hash de commit du repository Git
- Tester le job manuellement, afficher les variables d'environnement
- Installer les plugins *Parametrized Trigger Plugin* et *Jenkins Git Plugin*
- Modifier le lancement du builds *2\_weather\_integration* afin que le paramètre du premier job soit passé
- Utiliser ce paramètre pour effectuer un checkout du même commit

### 3.5 Mise en place d'une analyse qualité avec Sonar

Installer le plugins *SonarScanner*

Configurer Sonar dans :

- *Configuration Système* : Indiquer l'URL du serveur Sonar
- *Configuration des outils* : Définir un scanner s'installant automatiquement

Créer un nouveau job *2\_weather\_metrics* sur le même modèle que *2\_weather\_integration*, dans la partie build appeler le Scanner Sonar en indiquant le fichier de configuration Sonar fourni (*sonar-weather.properties*)

Tester l'exécution du job et l'apparition des résultats dans Sonar

### 3.6 Mise en place d'un pipeline

Installer les plugins *DependencyGraph*, *Copy Artifact* et *Join*

Créer un nouveau job *3\_weather\_deploy\_integration* qui se déclenchera lorsque les 2 jobs *2\_weather\_integration* et *2\_weather\_metrics* sont *successful*

Ce job se contentera d'exécuter un shell effectuant une copie du war créé lors de *1\_weather\_unit* dans un répertoire de votre choix