



epsi®



wis®

Formation B3 ASRBD

Projet CI/CD

Durée : 8h

Objectif

Manipuler Jenkins pour faire de l'intégration continue, de la livraison continue, et du déploiement continu.

Travail à faire

Exercice 1 : découverte de Jenkins

- installer Jenkins sur une VM Linux de votre poste
- créer une pipeline simple, que vous pourrez déclencher manuellement
- cette pipeline doit générer le fichier /tmp/exercice1.log, avec le contenu suivant :

Exécution de la pipeline Jenkins Exercice 1

Début d'exécution à \${date} sur \${nom de la machine}

Cette machine dispose de \${nb_coeurs} coeurs de CPU, et \${ram} de RAM.

Le disque principal est utilisé à \${util_disque}%.

Vous pouvez utiliser les fichiers /proc/cpuinfo et /proc/meminfo pour obtenir les infos sur le CPU et la RAM, et la commande df pour l'utilisation du disque.

Consignes pour la suite :

Vous aurez besoin d'un projet web disposant d'un front HTML/CSS/JS, et d'un back si possible en Java.

Le back **doit** contenir des tests (junit pour Java, phpunit ou atoum pour PHP, etc).

Si vous n'avez pas ce type de projets sous la main, vous pouvez coder rapidement une api en spring boot contenant une classe Employe avec une méthode setSalaire(float salaire) qui renvoie une exception si le salaire est négatif, et un controller qui a une méthode getEmploye() qui renvoie un employé codé en dur (pas besoin de base de données, de repository ou autre). Codez une page index.html qui fait appel à un product.js qui va fetch le product côté back.

Le projet doit être enregistré sur un git auquel votre Jenkins doit avoir accès (si vous utilisez github, créez un compte github spécifique, et donnez-lui accès à votre projet si celui-ci n'est pas public).

Exercice 2 : Jenkins et git

- créez une pipeline simple que vous pourrez déclencher manuellement
- la pipeline doit cloner le dépôt git, exécuter les tests du back, et afficher le résultat dans Jenkins

Exercice 3 : Jenkinsfile

- Dans votre projet git, créez une branche exercice3
- Rajoutez un Jenkinsfile à la racine du projet.
- Suivez les étapes détaillées ici :

<https://www.jenkins.io/doc/book/pipeline/getting-started/#defining-a-pipeline-in-scm> pour configurer Jenkins afin qu'il exécute les instructions définies dans votre Jenkinsfile

- Dans le Jenkinsfile, faites simplement exécuter les tests de votre projet pour le moment

Pour vous aider : <https://blog.stephane-robert.info/post/jenkins-pipeline-jenkinsfile-groovy/>

Exercice 4 : Livraison continue

- dans votre projet git, créez une branche exercice4
 - modifiez votre Jenkinsfile pour ajouter les stages de build : vous devez produire un zip pour le front, et une archive pour le back (jar pour le java, ou zip pour les autres langages)
- Attention : le build ne doit se faire que si les tests réussissent !
- mettez les deux zip dans un artifact Jenkins pour sauvegarder le résultat du build

Exercice 5 : Déploiement continu

- dans votre projet git, créez une branche exercice5
- modifiez votre Jenkinsfile pour déployer automatiquement votre application, en fonction des technologies utilisées
- vous aurez besoin d'un nom d'utilisateur et d'une clé privée SSH pour se connecter à votre VM linux d'hébergement : <https://www.jenkins.io/doc/book/using/using-credentials/>
- si vous avez un front en HTML/CSS/JS pur, déposez simplement le zip dans le dossier /var/www/html de votre machine Linux (avec apache ou nginx d'installé), et dézippez les fichiers
- si vous avez utilisé npm ou autre, déposez le zip sur votre serveur linux puis lancez les commandes pour redémarrer le service
- pour le back, déposez le jar ou le zip dans un dossier adapté, et démarrez/redémarrez le service.

Exercice 6 : variables et environnement

- dans votre projet git, créez une branche exercice6
- modifiez votre Jenkinsfile pour paramétrer votre projet au moment du build : vous devez enregistrer le numéro de port de votre application dans une variable d'environnement Jenkins par exemple

- un step de l'étape de build doit paramétrer un fichier de configuration de votre application pour définir le port en utilisant la valeur stockée dans la variable d'environnement Jenkins

Exercice 7 : Docker

- dans votre projet git, créez une branche exercice7
- modifiez votre jenkinsfile pour créer une image docker "projet-front" contenant le front de votre projet, et une image docker "projet-back" contenant le back de votre projet

Dans la vie réelle, on publierait ces images dans un dépôt docker local, pour faciliter le déploiement

Exercice 8 (bonus) : Gitlab

- déployez une instance locale de Gitlab, et utilisez le plugin gitlab de Jenkins pour lier les deux
- poussez votre projet sur le gitlab local
- configurer Jenkins pour exécuter la pipeline au push sur le projet

Technologies recommandées

- Serveurs : Linux Debian 12 ou RHEL 9 (version gratuite pour les développeurs)
- Pour PHP : installer php-fpm, pas le libmodapache2-php
- Pour le serveur web : apache2 ou nginx

Évaluation

Rendu :

- lien vers le dépôt git