

## Formation Jenkins

### Maîtriser l'intégration continue avec Jenkins

Durant cette formation Jenkins, les stagiaires apprendront à utiliser cet outil devenu la référence en matière d'intégration continue. Au travers de nombreux ateliers et exercices pratiques, ils s'entraîneront à automatiser builds, tests et déploiements, pour optimiser les processus de développement logiciel. A destination des ingénieurs DevOps, développeurs, responsables qualité ou encore architectes, notre formation à l'intégration continue avec Jenkins fournit une compréhension approfondie du rôle que joue Jenkins dans une approche moderne du développement. Les apprenants auront ainsi toutes les clés pour une mise en œuvre efficace de l'outil dès leur retour en entreprise.

### Durée

2 jours

### Objectifs pédagogiques

- Maîtriser les principes de livraison et déploiement continu et comprendre le rôle de Jenkins dans le cycle de développement logiciel
- Installer, configurer et dimensionner un serveur Jenkins
- Gérer l'intégration avec d'autres outils : Git, SonarQube, Docker...
- Mettre en œuvre les bonnes pratiques pour automatiser builds, tests et déploiements
- Implémenter des pipelines de livraison continue avec la suite de plugins Jenkins Pipeline
- Adapter l'exploitation de Jenkins à son contexte professionnel

### Public

Responsables qualité, Ingénieurs DevOps,  
Architectes techniques, Chefs de projet,  
Développeurs...

### Prérequis

Connaître Java et avoir les notions de base sur le cycle de développement logiciel. Une familiarité avec des outils comme Docker, Git, Maven ou JUnit est un bon plus mais non obligatoire.

### Programme de formation

Introduction à la formation à l'intégration continue avec Jenkins  
Présentation générale de la formation, objectifs et approches pédagogiques  
Les pratiques agiles, DevOps et la philosophie « continue » (intégration continue, livraison continue, déploiement continu et feedbacks)  
Qu'est-ce que Jenkins ? L'orchestrateur de l'intégration continue

Bref retour sur les origines de Jenkins (anciennement Hudson)  
Atouts, positionnement et outils concurrents : TravisCI, Bamboo, Teamcity...  
Le workflow Jenkins : les pipelines CI/CD  
Gestion du code source (Source Code Management) : commits, branches et Pull Requests  
Activités de test logiciel et automatisation  
Installer et démarrer le serveur Jenkins

Exemples de cas pratiques : tour de table, présentation des participants et du formateur ou de la formatrice, échanges sur les différents contextes professionnels, les pratiques d'intégration et de livraison continues, et les perspectives d'application des acquis de la formation. Présentation des différentes possibilités d'installation (manuelle, image docker, packages...).

### Prise en main et configuration

Terminologie Jenkins : master, node, job, agent...

Structure des répertoires jenkins\_home et jobs

L'interface utilisateur et ses composants

Configurer les outils Git, Java, Maven/Make/Gradle/Ant et le serveur de mail

Gérer les plugins

Gérer les notifications, en natif ou avec un plugin (Email-Ext, Slack Notification)

Différents types de projets Jenkins : freestyle vs pipeline

Exemples de cas pratiques : découverte de l'interface Jenkins, configuration générale du serveur, configuration des outils et manipulations de plugins, mise en place d'un premier build simple, freestyle ou Maven.

### Automatisation du build : jobs freestyle

Apprendre Jenkins pour le build, vue d'ensemble et meilleures pratiques

Les différentes sections de configuration d'un job freestyle

Gérer les interactions avec Git

Définir les déclencheurs et étapes de build

Les variables d'environnement Jenkins

Artefacts et bonnes pratiques d'archivage

Les empreintes (fingerprints)

Exemple de cas pratiques : création d'un job basé sur un repo Git, mise en œuvre des bonnes pratiques de configuration.

### Automatisation des tests et métriques qualité

Intégrer la qualité dans le processus de build : bonnes pratiques et outils

Types de tests et automatisation

Les différents frameworks de test : JUnit, TestNG, JMeter, Cucumber...

Configurer des rapports

Mesurer la couverture de test

Exemples de cas pratiques : configuration d'un projet Jenkins intégrant l'automatisation de tests unitaires, d'intégration, d'acceptance ou de performance, utilisation de différents plugins et outils (Violations, SonarQube, Checkstyle, FindBugs...).

### Distribution de jobs : mise en place d'une architecture maître/agent

Présentation des nœuds maître et agent (esclave)

Types de nœuds et champs

Les différentes possibilités pour démarrer les nœuds : SSH, Java Web Start, service Windows ou script manuel

Agents Cloud et Docker

Gestion des configurations, utilisation d'outils complémentaires (Puppet, Chef, Ansible...)

Bonnes pratiques pour dimensionner l'architecture

Exemples de cas pratiques : utilisation de Jenkins en cluster, configuration des nœuds, répartition de jobs entre agents, utilisation d'agents Docker.

### Paramètres et relations entre jobs

Paramétrer un job (manuellement ou avec le plugin Parameterized Trigger)

Axes et matrice de configuration : les jobs multi-configurés

Chaînage de jobs : passer des paramètres ou artefacts entre jobs

Exemples de cas pratiques durant cette formation Jenkins : utilisation des paramètres de jobs, multi-configuration, passage de données entre jobs

### Pipelines

Présentation de l'approche, bénéfices et cas d'usage

Les syntaxes : déclarative ou de script Groovy et DSL

Le fichier Jenkinsfile : pipeline-as-code

Sections et étapes d'un pipeline

Créer et exécuter un pipeline

Paralléliser des tâches

Modifier et rejouer un pipeline

Gérer les erreurs

Utiliser des images Docker comme environnement d'exécution pour un pipeline

Introduction aux bibliothèques partagées

Exemples de cas pratiques : création de pipelines avec l'éditeur Blue Ocean, utilisation de la syntaxe déclarative ou de script, mise en place

de tâches parallèles, exécution avec Docker, modification de pipelines.

## Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques.
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

## Modalités d'évaluation

- **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience.
- **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.