DAWAN

Plan de la formation

www.dawan.fr

Formation Intégration continue, déploiement et conteneurisation avec GitLab CI pour projets Java

Formation éligible au CPF, contactez-nous au 02/318.50.01

Durée :	5 jours (35 heures)
Tarif inter-entreprises :	3 975,00 € HT (Présentiel) 3 180,00 € HT (Distanciel)
Public :	Opérationnels, Développeurs, Chefs de projets
Pré-requis :	Connaissance du cycle de vie d'une application, maîtrise des commandes de base Git
Objectifs :	Mettre en oeuvre et exploiter un serveur d'intégration continue. Gérer les interconnexions avec un système de build et de tests
Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
Modalité d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
Sanction :	Passage de la certification professionnelle selon les modalités d'évaluation certificative définies par le référentiel. Date de session d'examen envisagées indiquées dans le contenu de la formation.

Référence :	DEV103002-F
Note de satisfaction des participants :	Pas de données disponibles
Contacts :	commercial@dawan.fr - 09 72 37 73 73
Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
Délais d'accès :	Variable selon le type de financement.
Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Comprendre l’intégration continue et découvrir GitLab CI/CD

Rappeler le cycle de vie d’une application Java et les enjeux de l’intégration continue dans une démarche agile et DevOps.
Identifier les environnements de développement, test, recette et production.
Comprendre le rôle de GitLab dans la gestion du code source, des pipelines, des artefacts et des livrables applicatifs.
Positionner GitLab CI/CD dans une chaîne Java : dépôt Git, build Maven ou Gradle, tests automatisés, analyse qualité, packaging et déploiement.
Présenter les outils complémentaires : Docker, Docker Compose, SonarQube, registre d’artefacts, gestion des variables et secrets.
Atelier fil rouge : Mise en place d’un projet GitLab pour une application Spring Boot, création du dépôt, paramétrage initial, README.md et première organisation des branches.

Maîtriser les bases du YAML pour GitLab CI

Comprendre la syntaxe de base du YAML et ses règles d’écriture.
Déclarer des variables, collections, ancres et structures réutilisables.
Identifier les erreurs fréquentes dans un fichier .gitlab-ci.yml.
Structurer un premier fichier de pipeline lisible et maintenable.
Adapter les conventions d’écriture YAML aux besoins d’un projet Java.
Atelier fil rouge : Création du premier fichier .gitlab-ci.yml, validation de la syntaxe, déclaration de variables et factorisation des premiers éléments du pipeline.

Gérer des builds Java avec GitLab CI

Comprendre le fonctionnement des pipelines, stages, jobs, tags, artefacts et runners GitLab.
Configurer des jobs automatiques et manuels.
Utiliser un runner GitLab avec une image Docker adaptée au build Java.
Mettre en place un build Maven ou Gradle pour une application Spring Boot.
Gérer les dépendances, les dépôts, les caches et les versions de livrables.
Organiser les branches, tags et règles d'exécution des pipelines.
Produire et conserver les artefacts de build : jar, rapport de compilation, package applicatif.
Utiliser les variables CI/CD pour paramétrer le comportement du build.
Atelier fil rouge : Configuration d'un pipeline de build Maven ou Gradle pour une application Spring Boot, génération du livrable, gestion du cache de dépendances et archivage des artefacts.

Contrôler la qualité du code avec SonarQube

Comprendre les objectifs de la qualité de code dans une chaîne d'intégration continue.
Présenter les principaux contrôles : règles de codage, complexité, duplication, dette technique, vulnérabilités et code smells.
Intégrer SonarQube dans un pipeline GitLab CI.
Configurer l'analyse d'un projet Java Spring Boot.
Exploiter les rapports de qualité et interpréter les résultats.
Mettre en place des seuils de qualité et des critères d'acceptation avant fusion ou déploiement.
Atelier fil rouge : Ajout d'une analyse SonarQube au pipeline GitLab CI, génération du rapport qualité, interprétation des anomalies et correction d'exemples simples.

Automatiser les tests dans une chaîne CI/CD Java

Identifier les différents niveaux de tests : tests unitaires, tests d'intégration, tests de non-régression, tests de sécurité et tests de performance.
Exécuter automatiquement les tests Java dans GitLab CI.
Produire et publier les rapports de tests dans GitLab.
Mesurer la couverture de tests et exploiter les résultats dans le pipeline.
Bloquer ou autoriser la poursuite du pipeline selon les résultats obtenus.
Préparer les tests d'intégration nécessaires à une application Spring Boot utilisant une base de données ou des services externes.
Atelier fil rouge : Intégration de tests unitaires et de tests d'intégration Java dans le pipeline, publication des rapports, analyse des échecs et correction

du pipeline.

Conteneuriser une application Spring Boot avec Docker

Comprendre le rôle de Docker dans une chaîne CI/CD applicative.
Créer un Dockerfile adapté à une application Spring Boot.
Construire une image applicative à partir du livrable Java.
Optimiser l'image : image de base, couches, variables d'environnement, exposition des ports, sécurité minimale.
Exécuter localement l'application conteneurisée.
Gérer les tags d'images en fonction des branches, commits ou versions applicatives.
Atelier fil rouge : Création du Dockerfile de l'application Spring Boot, construction de l'image Docker dans GitLab CI et exécution du conteneur applicatif.

Orchestrer un environnement applicatif avec Docker Compose

Comprendre l'intérêt de Docker Compose pour reconstituer un environnement de développement, de test ou de recette.
Définir les services nécessaires : application Spring Boot, base de données, SonarQube ou service externe simulé selon le projet.
Configurer les réseaux, volumes, variables d'environnement et dépendances entre services.
Utiliser Docker Compose dans les tests d'intégration ou dans une recette technique simple.
Documenter l'environnement conteneurisé pour faciliter la reprise du projet par une autre équipe.
Atelier fil rouge : Création d'un fichier docker-compose.yml pour lancer l'application Spring Boot et sa base de données, puis utilisation dans une étape de test ou de recette.

Gérer les variables, secrets et configurations CI/CD

Distinguer variables de projet, variables de groupe, variables protégées et variables masquées.
Gérer les configurations selon les environnements : développement, test, recette, production.
Sécuriser les informations sensibles : tokens, mots de passe, accès base de données, clés de déploiement.
Éviter l'exposition de secrets dans le code source, les logs ou les artefacts.
Utiliser les variables CI/CD dans les scripts de build, de test, de packaging et de

déploiement.
Atelier fil rouge : Paramétrage des variables et secrets GitLab CI/CD pour construire, tester et déployer l'application sans exposer d'informations sensibles.

Mettre en place une stratégie de déploiement

Définir une stratégie globale d'automatisation du déploiement.
Préparer les scripts de déploiement et de mise à jour.
Utiliser les artefacts et images Docker produits par la chaîne CI/CD.
Définir les étapes de déploiement manuel ou automatique selon l'environnement cible.
Gérer les environnements GitLab, les validations manuelles et les restrictions de déploiement.
Limiter la concurrence entre déploiements avec les groupes de ressources.
Prévoir les scénarios de rollback et de retour à une version stable.
Atelier fil rouge : Construction d'un script de déploiement, déclenchement manuel vers un environnement de recette, validation du livrable et préparation d'un scénario de rollback.

Documenter le déploiement et les procédures d'exploitation

Rédiger une procédure de déploiement claire et exploitable par une équipe projet ou une équipe d'exploitation.
Décrire les prérequis techniques, variables nécessaires, étapes du pipeline, scripts utilisés et points de contrôle.
Documenter les environnements de test, recette et production.
Prévoir les procédures de vérification après déploiement et de retour arrière.
Relier la documentation technique au README.md, au wiki GitLab ou au dossier projet CDA.
Atelier fil rouge : Rédaction de la documentation de déploiement de l'application Spring Boot, incluant build, tests, image Docker, Docker Compose, variables CI/CD, procédure de déploiement et rollback.

Administrer et sécuriser les outils GitLab CI/CD

Comprendre les principaux points d'administration d'un serveur ou espace GitLab utilisé pour l'intégration continue.
Gérer les utilisateurs, rôles, permissions et accès aux projets.
Surveiller les journaux, l'espace disque, la charge CPU et la consommation des runners.
Identifier les risques liés aux runners partagés, aux permissions trop larges et aux

secrets mal protégés.
Appliquer les bonnes pratiques de sécurité dans une chaîne CI/CD : moindre privilège, protection des branches, validation des merge requests, contrôle des variables et audit des pipelines.
Atelier fil rouge : Vérification des paramètres de sécurité du projet GitLab, protection des branches, contrôle des droits et revue des points sensibles de la chaîne CI/CD.

Synthèse et mise en situation finale

Reprendre l'ensemble de la chaîne CI/CD mise en place pendant la formation.
Vérifier le passage complet du pipeline : build, tests, qualité, packaging, image Docker, artefacts, déploiement et rollback.
Analyser les rapports produits et identifier les points d'amélioration.
Préparer les livrables utiles pour un projet CDA : fichier .gitlab-ci.yml, Dockerfile, docker-compose.yml, rapports de tests, rapport qualité, scripts de déploiement et documentation technique.
Atelier fil rouge : Exécution complète de la chaîne CI/CD sur l'application Spring Boot, correction des anomalies, validation du livrable final et présentation synthétique de la démarche DevOps mise en œuvre.

Supports pédagogiques

Support du formateur, exercices pratiques, cas fil rouge Java / Spring Boot, exemples de fichiers .gitlab-ci.yml, Dockerfile, docker-compose.yml, scripts de déploiement, ressources complémentaires et éléments mobilisables pour le dossier projet CDA.