Git : Gestion de dépôts + Gitlab-CI / Intégration continue

INFORMATIONS GÉNÉRALES

Type de formation : Formation continue

Éligible au CPF : Non

Domaine : Développement

Action collective : Non

Filière : DevOps

Rubrique : Outils

PRÉSENTATION

Objectifs & compétences

A l'issue de la formation, le stagiaire sera capable :

- Maîtriser l'usage de commandes Git pour la gestion d'un dépôt de sources -
- Mettre en oeuvre et exploiter un serveur d'intégration continue. Gérer les
- interconnexions avec un système de build et de tests

Public visé

Opérationnels, Développeurs, Chefs de projets

Pré-requis

Connaissance du cycle de vie d'une application, maîtrise des commandes de base Git

PROGRAMME

Découvrir Git
Principes de gestion de contrôle de source (SCM)
Historique, contrôle local, centralisé et distribué
Fonctionnement des instantanés, comparaison avec les différences
Installation (Linux, MacOS, Windows)
Accès au manuel : man / help
Configuration initiale de Git : préférences, profil utilisateur
Initialisation d'un dépôt local
Atelier : Installation de Git - Création d'un projet

Comprendre le cycle de vie du répertoire de travail
Concepts de répertoire de travail, index et dépôt
Vérifier l'état de la copie de travail : status
Indexer ses modifications : add
Ignorer des fichier : .gitignore
Valider ses modifications : commit
Supprimer et déplacer des fichiers
Atelier : contributions et validations

Visualiser l'historique
Visualiser les modifications : log
Personnaliser le format : stat, pretty, …
Filtrer par date, auteur, message de commit, contenu modifié, …
Visualiser et exporter une différence (format natif, outil externe)
Étiqueter ses validations : étiquettes légères et annotées
Rechercher avec git-grep

Annuler des actions
Réécrire la dernière validation
Désindexer un fichier
Réinitialiser un fichier

Code de formation : LEDN209

€ Tarifs

Prix public : 3475 €

Tarif & financement :
Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :
Le plan de développement des compétences de votre entreprise : rapprochez-vous de votre service RH.
Le dispositif FNE-Formation.
L'OPCO (opérateurs de compétences) de votre entreprise.
France Travail: sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.
CPF -MonCompteFormation
Contactez nous pour plus d'information : contact@aston-institut.com

Lieux & Horaires

Durée : 35 heures

Délai d'accès : Jusqu'a 8 jours avant le début de la formation, sous condition d'un dossier d'insciption complet

Prochaines sessions

Consultez-nous pour les prochaines sessions.

Travailler avec les branches
Principe de branche, le pointeur HEAD
Créer une branche
Basculer entre les branches, le mode détaché
Fusionner les branches : avance-rapide, trois sources
Gérer les conflits de fusion
Outil de fusion externe : mergetool (emerge, vimdiff, meld, …)
Visualiser les branches existantes, celles qui ont été fusionnées
Supprimer une branche
Stratégies de gestion de branches : branche longue, thématique, …

Travailler avec un dépôt distant
Dépôt distant, branches distantes, suivi de branche
Afficher et inspecter les dépôts distants
Ajouter, renommer, retirer ses dépôts distants
Tirer, pousser et supprimer une branche distante

Réécrire l'histoire, rebaser
Mise en garde : les dangers de la réécriture
Rebaser une portion de branche
Quand rebaser et quand fusionner

Remiser et nettoyer
Remiser son travail en cours
Créer une branche depuis une remise
Nettoyer son répertoire de travail

Personnaliser Git
Configurer l'éditeur par défaut, exclusions automatiques, …
Création et utilisation d'alias
Outils graphiques : Git-Gui, GitKraken, SmartGit, …
Créer des filtres : smudge et clean
Crochets côté client : pre-commit, pre-rebase, post-rewrite…
Crochets côté serveur : pre-receive, update, post-receive

Faire référence à un projet externe
Principe des sous-modules
Déclarer, tirer et mettre à jour un sous-module
Modifier et gérer les conflits sur une bibliothèque externe
Problèmes des sous-modules

Git sur un serveur
Les protocoles : local, HTTP, SSH, Git
Création d'un dépôt nu, comptes utilisateurs
Utilisateur git unique, clés SSH et git-shell
Démon Git
Atelier : Mise en place d'un serveur Git

Gestion de dépôt web
Un serveur simple et léger : GitWeb
Une plate-forme plus complète : GitLab
GitLab : configuration des utilisateurs
GitLab : exploration de projet, suivi des activités, wiki
GitLab : issue manager, web hooks, revue de code
Un service hébergé clé-en-main : GitHub
GitHub : création de compte et configuration
GitHub : règles de contribution
GitHub : maintenance d'un projet
Atelier : Récupération et exploration d'un GitLab

Comprendre l'intégration continue et découvrir GitLab
Processus de développement, tests unitaires / d'intégration
Intégration continue : présentation, positionnement dans une démarche agile
Gestion des environnements : développement, recette, production
Outils de conteneurs applicatifs (Docker)
Configurations système et applicative et outils de centralisation (Puppet, Ansible)
Panorama outils de gestion : versionnement, build, tests, qualité
GitLab-CI : présentation, fonctionnalités
Types d'installation
Notion de projet, documentation (README.md, Wiki, ...)
Atelier : Mise en place de GitLab, tour d'horizon de l'interface, création de dépôts et Paramétrage

Maîtriser les bases du YAML
YAML : syntaxe de base, spécificités

Déclaration et utilisation de variables
Collections
Ancres

Gérer des builds avec GitLab CI
Principe de fonctionnement : pipelines, stages, tasks, artefacts, tags
Structure d'un build de projets, le fichier manifeste .gitlab-ci.yml
Jobs et Runners, utilisation de Docker
Mise en place de builds : automatiques / manuels
Plugins pour la gestion des dépôts de source
Outils de build : Maven, Gradle,...
Organisation des branches et des tags
Gestion des dépendances et dépôts, mise en place d'un cache
Intégration des dépôts avec les outils de build
Gestion des notifications
Création et utilisation de variables dans les paramètres CI/CD
Lancement de jobs en parallèle
Atelier : Interfaçage avec des dépôts de dépendances - Configuration et lancement de builds
(applications web JS ou services Java)

Contrôler la qualité du code
Présentation, gestion de la qualité du code
Panorama des outils : Checkstyle, FindBugs, ...
Rapport de qualité : configuration, plugins (Violations)
Autres rapports : complexité, tâches, ...
Atelier : Intégration d'outils de gestion de qualité du code (SonarQube) dans une démarche
d'intégration continue

Automatiser les tests
Types de tests
Automatisation, couverture
Tests unitaires et d'intégration
Tests d'acceptance, tests de performances
Optimisation des tests
Atelier : Multiples scénarios d'automatisation de tests unitaires, d'intégration, de Performances

Mettre en place une stratégie de déploiement
Stratégie globale d'automatisation
Scripts de déploiement et de mise à jour
Rollbacks
Gestion des artefacts (archivage)
Utilisation des groupes de ressources pour limiter la concurrence
Atelier : Construction de scripts de déploiement

Administrer les outils
Sécurité du serveur d'intégration continue
Gestion des utilisateurs : bases, rôles, autorisations
Gestion des journaux
Espace mémoire/charge CPU, espace disque
Monitoring
Atelier : Multiples tâches d'administration du serveur

MODALITÉS

Modalités

Modalités : en présentiel, distanciel ou mixte . Toutes les formations sont en présentiel par défaut mais les salles sont équipées pour faire de l'hybride. – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise.
Pédagogie : essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.
Ressources techniques et pédagogiques : Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom.
Pendant la formation : mises en situation, autodiagnostics, travail individuel ou en sous-groupe sur des cas réels.

Méthode

Fin de formation : entretien individuel.
Satisfaction des participants : questionnaire de satisfaction réalisé en fin de formation.
Assiduité : certificat de réalisation.
Validations des acquis : grille d'evalution des acquis établie par le formateur en fin de formation.