

Formation Git : GitLab & GitFlow

Durée

2 jours (14 heures)

Présentation

En quelques années, [Git](#) est devenu le gestionnaire de source le plus utilisé au monde. Basé sur une gestion de versions décentralisée, Git est un logiciel libre et multiplateforme créé par le non moins célèbre Linus Torvalds. Il se veut simple et performant, Git a pour tâche principale de gérer l'évolution du contenu d'une arborescence.

Dans cette formation, vous apprendrez à maîtriser toute sa puissance ainsi qu'à utiliser les nombreux outils autour de ce projet, tels que GitLab et GitHub, et à collaborer efficacement en équipe grâce à Gitflow.

Nous commencerons par appréhender toutes les subtilités de Git en mode développement à travers l'utilisation de commandes de base, puis avancées, pour finir sur la gestion de l'historique et l'interaction avec des dépôts distants.

Nous aborderons également une approche approfondie de GitLab qui depuis sa création a su développer une infrastructure moderne, simple et claire pour devenir aujourd'hui un outil incontournable de gestion de projet web. Plus qu'un simple gestionnaire de code source, GitLab vous permettra de gérer les droits d'accès, les remontées de bugs, les sprints de votre cycle de vie logiciel.

GitLab permet également d'automatiser le déploiement de projets vers la production et avec ses outils complets, il vous permettra une intégration optimale et un développement en continu avec GitLab CI/CD (Continuous Integration & Delivery) qui vous accompagnera dans vos démarches DevOps d'intégration continue.

Et enfin GitFlow le workflow collaboratif multi-branche qui vous permettra une élaboration et maîtrise de vos branches afin de mieux gérer des demandes clients ou leurs documents. Git Flow est une méthode, une architecture Git permettant de séparer au maximum le travail et de toucher le moins possible à la branche Master. Cette méthode représente donc une architecture en branches très efficaces pour vous aider dans vos développements web & logiciel.

Des problèmes de migration vers Git ? Notre module complémentaire est fait pour vous : dans cette partie optionnelle nous vous proposons de vous accompagner lors de la migration de SVN (Subversion) vers GIT.

Comme dans toutes nos formations nous utiliserons la dernière version des logiciels ([GitLab 13.6](#) sortie en février 2019 et [Git-Flow AVH 1.12](#))

Objectifs

- Installer et configurer l'outil GIT
- Comprendre les différences entre la gestion de version centralisée et décentralisée
- Gérer efficacement les dépôts locaux et distants
- Manipuler les commandes de GIT pour gérer les fichiers et les branches
- Gestion de workflows et de dépôts git avec GitFlow
- Utilisation des outils d'intégration et de déploiement continu avec GitLab CI
- Gagner en efficacité en utilisant les outils les plus efficaces de l'écosystème GIT
- Intégrer et former rapidement un nouvel arrivant à votre projet
- Appréhender la migration de SVN vers GIT

Public visé

Développeurs, Architectes, Administrateurs systèmes

Pré-requis

Connaissances de base d'un gestionnaire de source

Programme de notre formation GitLab CI & GitFlow avec GIT

Jour 1 - Git & GitFlow

Introduction

- Contexte et historique de la gestion de versions
- Cas d'utilisation
- Présentation de Git
- Avantages de Git

Commandes de base

- Création du dépôt, dossier .git
- Les 3 états de Git
- Commandes add et commit
- Commandes status, log et diff

Les branches

- Intérêt du branching
- Pointeurs et HEAD
- Divergence
- Les différentes méthodes de fusion
- Bonnes pratiques de fusion

Remotes

- Dépôt central et multi-dépôt distants
- Les commandes fetch et pull
- La commande push
- Bonnes pratiques d'interaction

Commandes avancées

- Reset et reflog
- Rebasing, Squashing
- Gestion du Stash
- Blame
- Recherche d'erreurs avec bisect
- Cherry-pick

Jour 2 - Git Flow et GitLab

Workflows collaboratifs

- Pourquoi utiliser un workflow
- Les principaux workflows
- Zoom sur Git Flow et mise en pratique
- Github Flow et GitLab Flow
- Introduction à GitLab
- Forges logicielles
- Features de GitLab
- Comment choisir une forge logicielle
- CE vs EE

Installation de GitLab

- Omnibus vs Docker
- Installation sur GCP
- Configuration GitLab SMTP
- Configuration bas-niveau

Administration de GitLab

- Tour d'horizon de l'interface de gestion
- Administration des projets
- Gestion des commits et du code source
- Gestion des accès
- Assignment des permissions
- Utilisation des issues
- Milestones pour la gestion de projets

Bonnes pratiques de collaboration avec Git et GitLab CI

- Merge requests
- Rebase (+squash, +fast-forward)
- Branches protégées
- Wiki et documentation
- Intégration Continue avec GitLab CI
 - Configuration GitLab CI/CD
 - Pipelines de CI/CD
 - Gestion des runners
- Déploiement continu et GitLab Runner
- Mise en place d'un workflow de collaboration sur GitLab

GitLab CI Avancé (MODULE COMPLÉMENTAIRE + 2 jours)

Utilisez toute la puissance de GitFlow ainsi que la puissante chaîne CI de GitLab

GitFlow

- Branches de features/bugfix
- Branches de versions (et tags)

Installation et déploiement de GitLab

- Les deux modes d'installation seront présentés (docker & .deb)
- Retour d'expérience de la partie administration de GitLab
- A l'issue de la formation, nous laisserons une instance fonctionnelle sur leur serveur

GitLab & GitFlow

- Notions du Merge Request (MR)
- Explorer au travers d'un TP les interactions et collaborations qui peuvent être faites avec une MR, les fonctions qui assistent la revue de code, et les subtilités qui peuvent arriver
- Le Workflow idéal
- Outils externes qui peuvent compléter GitLab dans cette partie

GitLab Issues

- Présentation des issues
- Faire évoluer vos workflows graduellement pour favoriser l'adoption, plutôt qu'être trop disruptif
- Une démonstration & live démo

GitLab CI

- Faire un TP CI, pour au moins détailler toute la partie GitLab (et éviter Docker)
- Montrer un cas simple avec une CI de tests unitaires (sans Docker)
- Montrer un cas plus intéressant avec des tests fonctionnels. (avec un cas Docker simple)
- Conclusion

MIGRER DE SUBVERSION VERS GIT (MODULE COMPLÉMENTAIRE + 1 jour)

Utilisez GIT en multidépôt et modifiez votre historique

TERMINOLOGIE GIT VS SVN (QR)

MIGRER DE SVN VERS GIT

- Retour d'expérience
- Les différentes méthodes

LA COMMANDE GIT-SVN

- La migration en 2 étapes
- Avantages et inconvénients
- TP: Migrer un dépôt SVN vers GIT

GESTION DE PLUSIEURS DÉPÔTS DISTANTS (REMOTE) AVEC GIT

- Contribuer sur un seul dépôt
- Contribuer sur plusieurs dépôts
- Les différents modes d'intégrations
- Exemple de maintenance d'un dépôt complexe : le projet GIT
- TP: Gérer plusieurs dépôts distants

COMMANDES BASIQUES ET MIGRATION

- Commandes basiques de GIT
- TP : Migration de SVN > GIT

COMMANDES AVANCÉES DE GIT

LES TYPES D'OBJETS

- Commit
- Blob
- Tree
- Tag

PORCELAINE

- Développer
- Inspecter
- Partager
- Fusionner
- Patcher
- Déboguer
- Administrer
- TP: Gérer un cycle complet de versioning

GIT HOOKS & SUB-REPOS SUBTREE VS SUBMODULES

- Dans quel cas utiliser un subtree ou un submodule
- Utiliser la commande native git submodule
- Faire un sub-tree à la main
- Utiliser un script : git-subrepo

GIT ATTRIBUTES

- Traitement des fichiers binaires
- Filtres bi-directionnels (clean & smudge)
- Attributs d'archivage
- Stratégie de fusion

GIT HOOKS

- Environnements des hooks
- Installer les hooks côté client
- Installer les hooks côté serveur

Sociétés concernées

Cette formation s'adresse aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiple permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.