Les Workflows Git

Qu'est-ce que c'est ?

Présenté par: TESSI Hardson

Explication

Un Workflow Git est un ensemble instructions ou de recommandations à suivre pour utiliser Git et GitHub de façon efficace. C’est, en quelque sorte, une ligne de conduite pour tous les collaborateurs. Les workflows Git encouragent les développeurs et les équipes DevOps à exploiter Git de façon efficace et cohérente.

DevOps: Combinaison de Développement et Operations

DevOps vise à automatiser et intégrer les processus entre les équipes de développement et informatiques, en utilisant un ensemble de pratiques et d'outils. Son objectif est d'améliorer la collaboration et l'efficacité en favorisant l'automatisation des tâches et l'intégration transparente entre ces deux équipes.

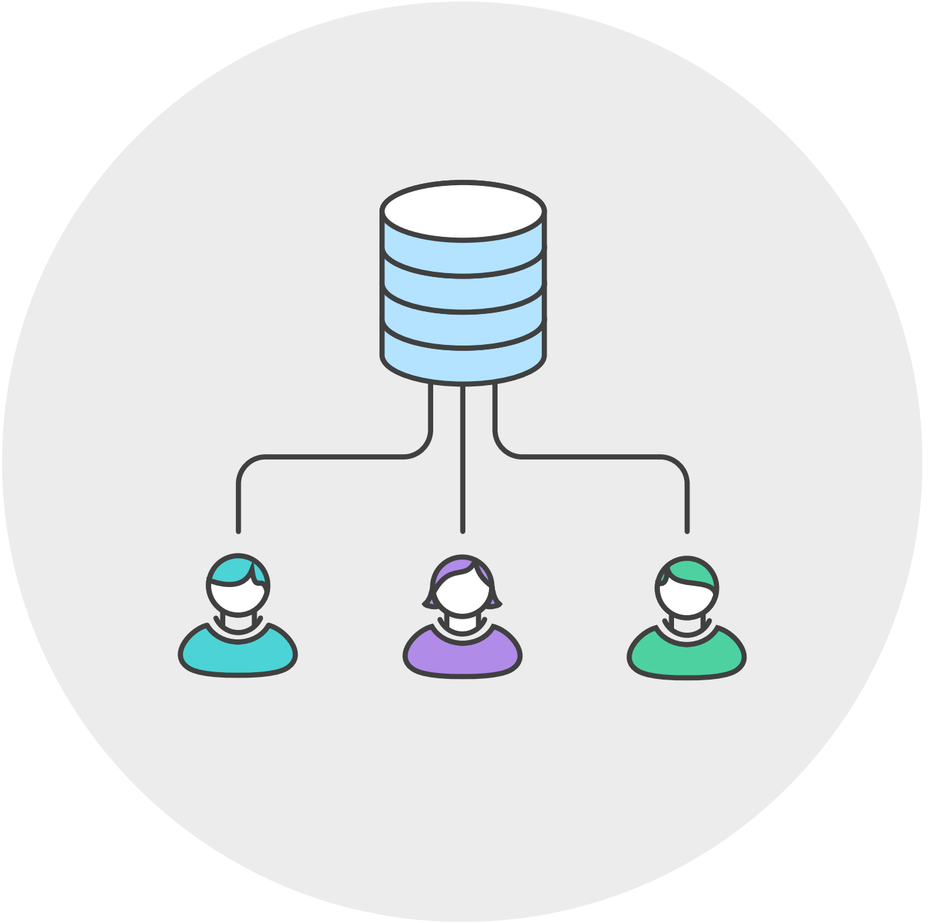
Avantages des Workflows Git ?

- **Collaboration efficace**
- **Gestion des versions**
- **Réduction des conflits**
- **Historique clair et traçabilité**
- **Automatisation et intégration continue**
- **Gestion des problèmes et des tâches**
- **Meilleure qualité du code**
- **Documentation et partage des connaissances**

Un workflow Git permet aux équipes de développement de bénéficier d'une collaboration harmonieuse, d'une gestion efficace des versions, d'une traçabilité claire et d'un processus de développement plus fluide.

Êtes-vous prêts à découvrir les Workflows Git les plus connus ?

Le workflow centralisé (Centralized Workflow)

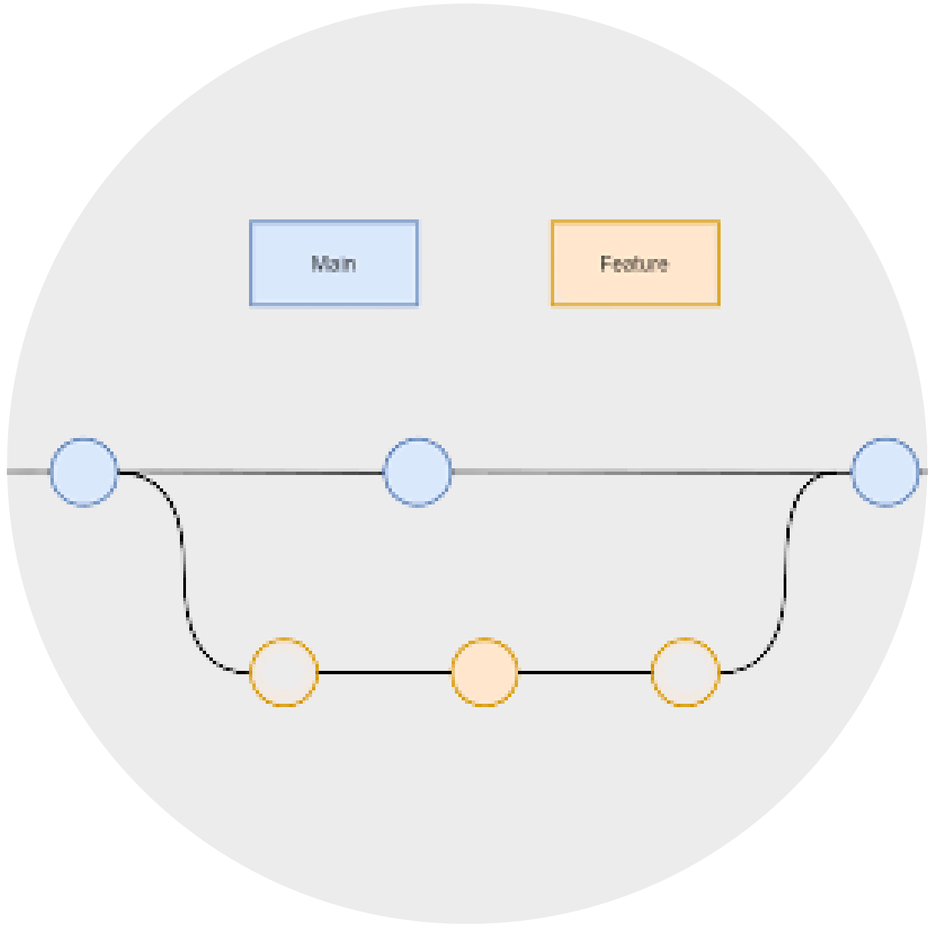


Dans ce modèle, il existe une seule branche principale qui contient le code stable. Les développeurs effectuent leurs modifications directement sur cette branche principale ou sur des branches de fonctionnalités spécifiques, puis fusionnent leurs modifications dans la branche principale une fois qu'elles sont prêtes.

Le workflow centralisé utilise un dépôt centralisé comme point d'entrée unique pour tous les changements apportés au projet. Tous les collaborateurs travaillent sur cette même branche. Un repository centralisé est créé, cloné par tous les collaborateurs, et les modifications sont ensuite synchronisées avec des commandes "git pull" et "git push".

Le workflow centralisé est la méthode la plus simple pour utiliser Git, favorisant ainsi une collaboration efficace avec une branche principale partagée par tous les contributeurs.

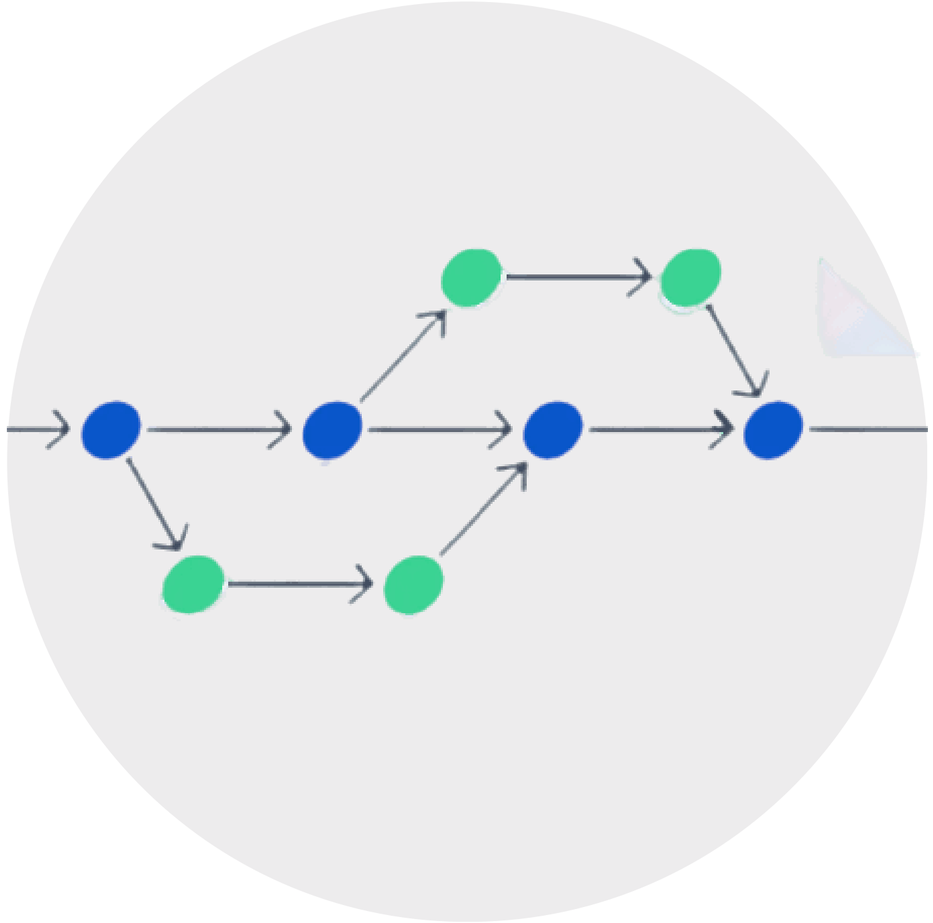
Le workflow de création de branches de fonctionnalités (Feature Branch Workflow)



Le workflow de création de branches de fonctionnalités vise à faciliter le développement de nouvelles fonctionnalités dans un projet logiciel tout en préservant l'intégrité de la branche principale (main). Un dépôt centralisé est créé et cloné par tous les collaborateurs, avec une branche dédiée à chaque fonctionnalité. Une fois achevée, la fonctionnalité est fusionnée avec la branche principale via une pull request.

Ce workflow permet à plusieurs développeurs de travailler simultanément sur différentes fonctionnalités sans altérer la base de code principale, assurant ainsi la stabilité de celle-ci tout en facilitant la collaboration et le développement parallèle.

Le workflow basé sur le tronc (Trunk Based Development)

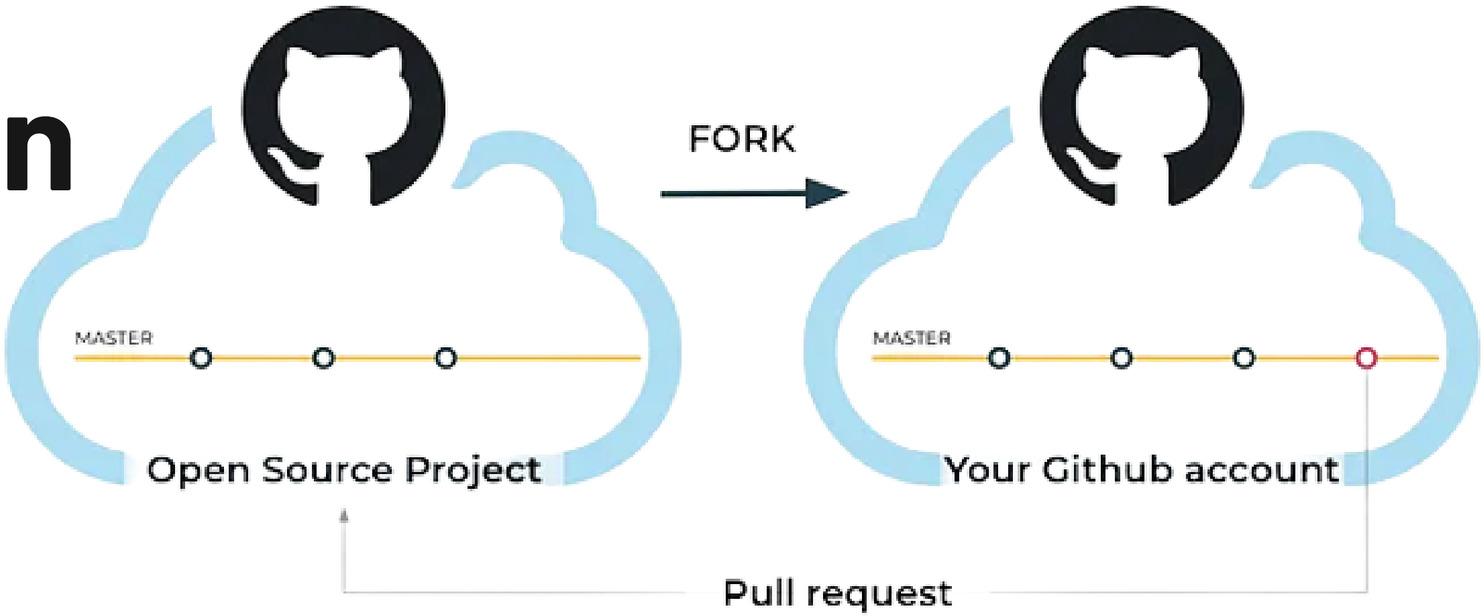


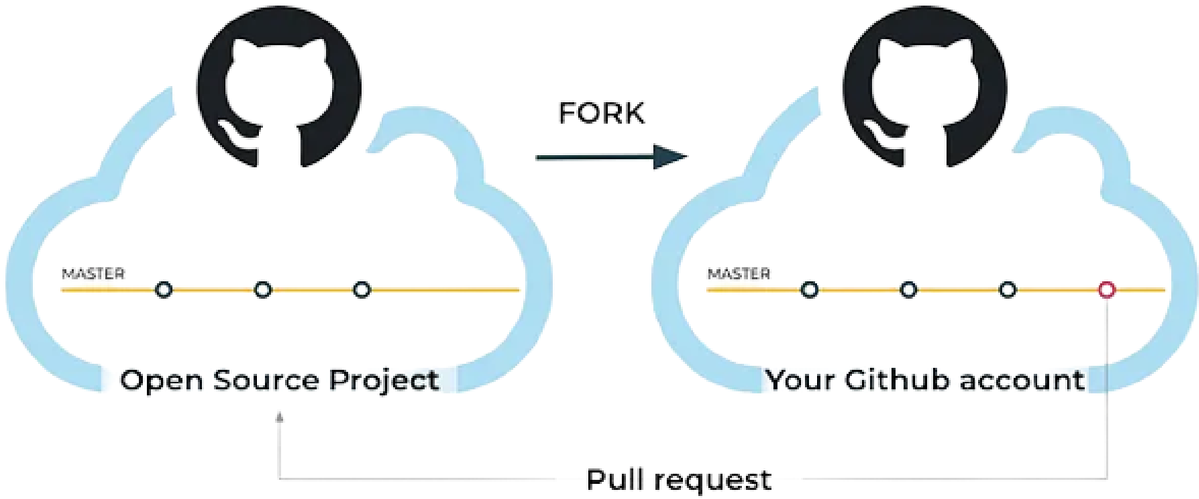
Le développement basé sur le tronc est une approche agile de plus en plus populaire en réponse à l'intégration continue et au développement continu. Il résout les problèmes liés aux branches de fonctionnalités prolongées et à leur complexe fusion.

Les opérations de fusion régulières sont effectuées sur la branche principale (Trunk), facilitant ainsi l'itération rapide et simplifiant les phases de fusion et d'intégration.

Cette méthode est essentielle pour la CI/CD et largement utilisée par les équipes DevOps, favorisant la fluidité des déploiements en production malgré la complexité croissante du code et l'agrandissement de l'équipe.

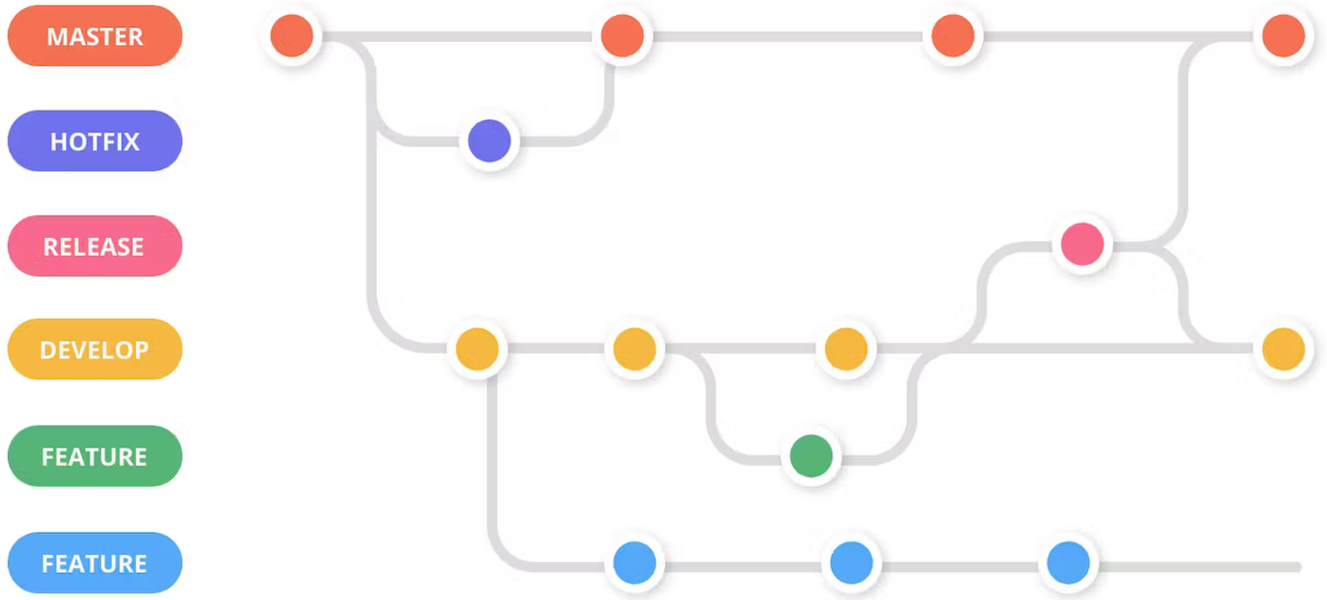
Le workflow de duplication (Forking Workflow)





Le workflow de duplication, également appelé Forking Workflow, est largement utilisé dans les projets open source. Le gestionnaire de projet crée un dépôt officiel du projet, tandis que chaque collaborateur crée une copie (fork) de ce dépôt sur le serveur. Les développeurs apportent leurs modifications dans leurs forks respectifs et soumettent des pull requests pour proposer leurs contributions. Le mainteneur du projet examine ces pull requests et décide d'accepter ou de refuser les modifications. Cette approche encourage une collaboration efficace et assure un contrôle adéquat des contributions au sein des projets open source.

Le workflow GitFlow



GitFlow est un modèle de gestion des branches qui facilite le développement logiciel. Il se concentre sur la création de branches de fonctionnalités dédiées pour développer chaque nouvelle fonctionnalité de manière isolée. En plus de ces branches de fonctionnalités, GitFlow propose d'autres branches clés pour simplifier la gestion du processus de développement. La branche "develop" est centrale car elle sert de point d'intégration où les fonctionnalités en cours de développement sont progressivement fusionnées. Les branches "release" regroupent les fonctionnalités finalisées en vue de préparer une future version. La branche principale, généralement appelée "main" , contient le code stable provenant des branches de release. Les branches "hotfix" sont utilisées pour résoudre rapidement les problèmes identifiés dans la branche principale. GitFlow présente de nombreux avantages dans des situations spécifiques. Par exemple, il facilite la gestion des contributions dans les projets open source en permettant aux collaborateurs de travailler sur des fonctionnalités spécifiques sans perturber le code principal. Il est également bénéfique pour les équipes comprenant des développeurs juniors qui ont besoin d'une structure claire pour les guider. De plus, pour les projets de grande envergure, GitFlow contribue à minimiser les risques de corruption du code en assurant une gestion rigoureuse des branches.

Comment choisir ou définir son Workflow Git ?

Le choix d'un Workflow Git dépend de plusieurs facteurs telsque:

- Nature du projet
- Délai du projet
- Objectifs de livraison et de collaboration
- Technologies et architecture
- Taille de l'équipe
- Niveau d'expérience de l'équipe

Merci !

Références

- OpenClassrooms
- Atlassian
- NicoEspeon