



Collaborer avec Git

1. Les dépôts distants

Un dépôt distant est une version du projet hébergée sur une plateforme comme GitHub, GitLab ou Bitbucket. Les dépôts distants permettent de partager du code avec d'autres développeurs.

1.1. Ajouter un dépôt distant

Pour connecter un dépôt local à un dépôt distant :

```
git remote add origin REPOSITORY_URL
```

Exemple :

```
git remote add origin https://github.com/user/project.git
```

1.2. Vérifier les dépôts distants

```
git remote -v
```

1.3. Changer l'URL d'un dépôt distant

Si l'URL d'un dépôt change :

```
git remote set-url origin NEW_URL
```

2. Pousser des modifications

Pour envoyer les modifications locales vers un dépôt distant :

```
git push origin branch_name
```

Options courantes :

- Ajouter `u` pour définir la branche distante par défaut pour les pushes futurs :

```
git push -u origin branch_name
```

- Pousser toutes les branches :

```
git push --all
```

3. Récupérer des modifications

3.1. Récupérer les données sans fusionner

Pour télécharger les mises à jour du dépôt distant sans les intégrer immédiatement dans la branche :

```
git fetch
```

- Cette commande met à jour les références locales pour les branches distantes.

3.2. Récupérer et fusionner les modifications

Pour synchroniser une branche locale avec les dernières modifications du dépôt distant :

```
git pull
```

| *N.B : Cela combine git fetch et git merge.*

Exemple :

```
git pull origin main
```

4. Résolution des conflits en collaboration

Lorsque plusieurs développeurs modifient le même fichier, Git peut signaler des conflits.

Étapes pour résoudre un conflit :

1. Identifier les fichiers en conflit avec `git status`.
2. Ouvrir les fichiers concernés et modifier manuellement les sections marquées :

```
<<<<< HEAD
Votre version
=====
Version du dépôt distant
>>>>> origin/main
```

3. Une fois les conflits résolus, ajouter les fichiers corrigés à la staging area :

```
git add conflicted_file
```

4. Terminer le processus avec un commit :

```
git commit
```

5. Revue de code et pull requests

5.1. Qu'est-ce qu'une pull request ?

Une pull request (PR) est une demande pour fusionner les modifications d'une branche dans une autre (souvent sur une plateforme comme GitHub ou GitLab).

5.2. Étapes pour créer une pull request :

1. Pousser la branche sur le dépôt distant :

```
git push origin branch_name
```

2. Sur la plateforme distante, ouvrir une PR.

3. Demander une revue pour obtenir des commentaires de coéquipiers.
4. Une fois approuvée, fusionner la PR.

6. Travailler avec des branches distantes

6.1. Suivre une branche distante

Pour créer une branche locale à partir d'une branche distante :

```
git checkout -b branch_name origin/branch_name
```

ou

```
git switch -c branch_name origin/branch_name
```

6.2. Supprimer une branche distante

Pour supprimer une branche sur le dépôt distant :

```
git push origin --delete branch_name
```

7. Les bonnes pratiques en équipe

7.1. Ne pas utiliser `main` pour le développement direct

Toujours travailler sur des branches spécifiques pour les nouvelles fonctionnalités ou corrections.

7.2. Régulièrement synchroniser les données

Avant de commencer à travailler, on récupère les dernières mises à jour :

```
git pull origin main
```

7.3. Divisez le travail

Il faut créer des commits fréquents et bien décrits pour faciliter le suivi des modifications.

7.4. Utiliser des messages de commit clairs

Un bon message de commit doit :

- Être descriptif : "Ajoute la fonctionnalité de connexion utilisateur".
- Utiliser l'impératif : "Corrige un bug dans l'authentification".

7.5. Résoudre les conflits rapidement

Identifier et résoudre les conflits dès qu'ils apparaissent pour éviter d'entraver le travail d'équipe.

8. Résumé des commandes principales

Action	Commande
Ajouter un dépôt distant	<code>git remote add origin URL_DU_DEPOT</code>
Vérifier les dépôts distants	<code>git remote -v</code>

Pousser une branche	<code>git push origin nom_branche</code>
Récupérer les mises à jour	<code>git fetch</code>
Synchroniser avec le dépôt distant	<code>git pull</code>
Résoudre les conflits	Résolution manuelle + <code>git add</code> + <code>git commit</code>
Supprimer une branche distante	<code>git push origin --delete nom_branche</code>