

TP GIT TAG :

Les tags dans Git (et donc aussi sur GitLab) sont un moyen de pointer vers un commit spécifique dans l'historique de votre dépôt. Les tags sont souvent utilisés pour marquer certaines versions ou états importants du code. Contrairement aux branches, les tags ne bougent pas lorsque de nouveaux commits sont ajoutés ; ils pointent toujours vers le même commit une fois qu'ils sont créés.

À quoi servent les tags Git sur GitLab ?

Marquer des versions spécifiques :

La plupart du temps, les tags sont utilisés pour marquer des versions de releases. Par exemple, si vous avez une version 1.0 de votre logiciel, vous pouvez créer un tag v1.0 qui pointe vers le commit de cette release.

Faciliter la recherche :

Les tags peuvent être utilisés pour faciliter la recherche de commits spécifiques qui représentent des étapes importantes du projet.

Documentation et référence :

Ils peuvent servir de références dans la documentation, les notes de version ou les tickets de support pour indiquer des points spécifiques dans l'historique du code.

Cas d'utilisation réels :

Gestion des releases :

Imaginons que vous développiez une application. À chaque fois que vous sortez une nouvelle version, vous pouvez utiliser un tag pour marquer cette version. Plus tard, si vous devez revenir en arrière pour corriger un bug sur cette version spécifique, vous pouvez facilement trouver le code exact de cette version grâce au tag.

Hotfixes :

Si un problème critique est découvert dans une version précédente, vous pouvez facilement naviguer vers cette version à l'aide du tag, y apporter des correctifs et éventuellement créer un nouveau tag pour cette version corrigée.

Historique de développement :

Si vous avez des étapes importantes dans le développement, comme des jalons ou des étapes de prototype, vous pouvez utiliser des tags pour marquer ces points. Cela peut être utile pour la revue de code ou pour retracer l'évolution du projet.

Référencer dans des tickets ou des issues :

Lorsque vous discutez d'un problème ou d'une fonctionnalité dans un ticket ou une issue, vous pouvez faire référence à un tag pour montrer exactement quelle version du code vous discutez.

Automatisation :

Dans les pipelines CI/CD, vous pouvez avoir des étapes qui se déclenchent spécifiquement sur des tags. Par exemple, vous pourriez avoir une étape qui déploie automatiquement votre application chaque fois qu'un nouveau tag est créé.

Mise en œuvre :

1^{ère} étape : tagger sur gitlab

- Se connecter à gitlab sur l'interface web
- Créer un tag v1.1 de votre branche develop (**pour ceux qui sont sur le serveur faites-le sur la branche avec votre prénom**). Mettez-lui une description

New Tag

Do you want to create a release with the new tag? You can do that in the [New release page](#).

Tag name

Create from

develop

Existing branch name, tag, or commit SHA

- Consulter le tag



- Allez sur votre dépôt local dans VSC et faites un git fetch pour synchroniser les changements.
- Faites un git tag pour afficher les tags présents sur votre gitlab

2^{ème} étape, tagger un commit et le push

- Toujours dans visual studio code, modifier ou créer un fichier dans la branche develop
- Commiter le
- Taguer le avec votre prénom
- Pusher votre commit et votre tag
- Aller voir sur votre gitlab si votre branche est à jour
- Aller voir votre tag sur gitlab