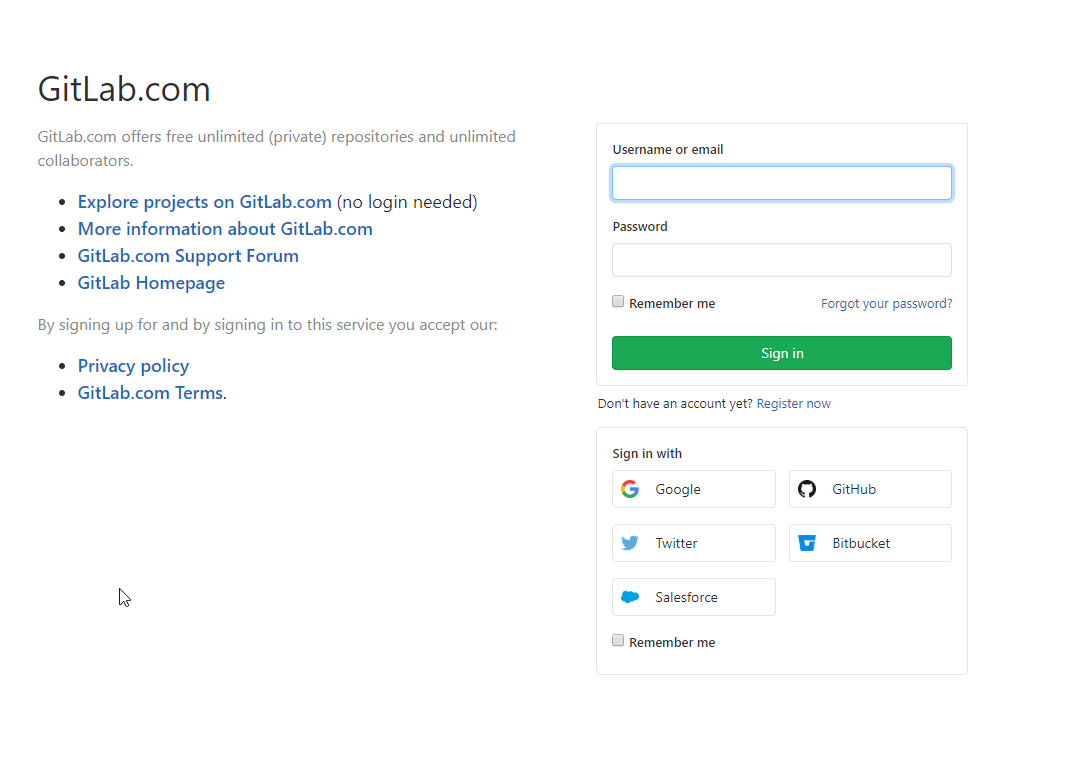
**Réalisez l’ensemble des étapes ci-dessous avec des captures d’écran de chaque étape**

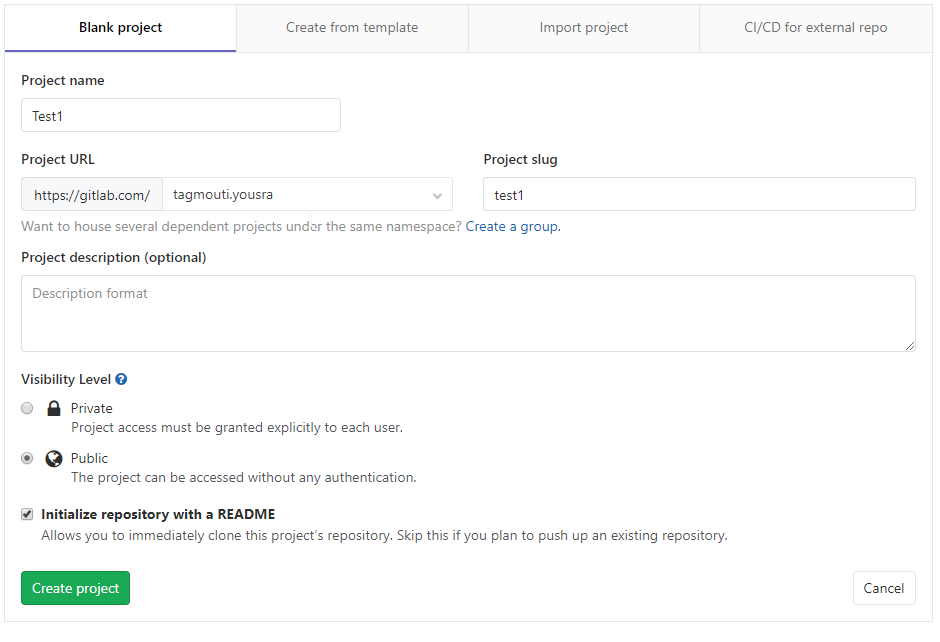
1. Il faut tout d’abord commencer par créer un compte utilisateur sur la plateforme **GitLab**



Vous pouvez utiliser plusieurs connecteurs externes, GitHub, Google, etc.

**Créer un premier projet sur GitLab**

1. Cliquez sur le bouton 
2. Vous pouvez créer un projet public ou privé. Nous allons choisir de créer un projet **public**
3. Initialisez votre projet avec un **README** (l’utilité est d’avoir un premier fichier dans votre répertoire, si vous voulez le cloner sur votre machine locale)

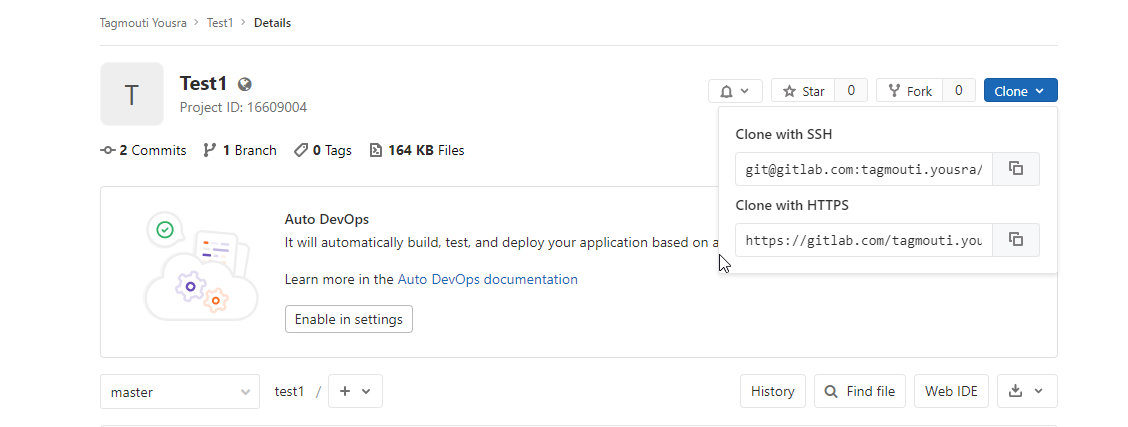


*Bravo vous venez de créer un premier projet GitLab !!!*

**Récupérer le contenu d’un projet GitLab sur votre machine locale**

Une fois que vous avez créé un projet ou ciblé un projet auquel ça vous intéresse de collaborer, vous devez récupérer l’URL du projet afin de le cloner

1. Connectez-vous à GitLab pour récupérer l’URL du projet qui vous intéresse, avec l’objectif de créer une première copie de ce projet sur votre machine (**clonage**).

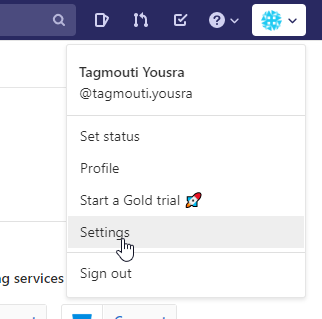


1. Récupérer l’url de votre projet disponible sur la section **Clone with HTTPS**

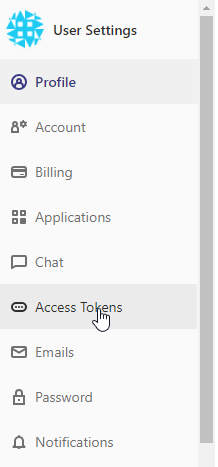
**Important :**

Pour cloner un projet sur votre machine locale, vous devez vous authentifier. Si vous ne voulez pas utiliser l’authentification standard vous avez la possibilité **d’utiliser un jeton (token) pour l’authentification**, pour cela vous devez **générer votre token** selon la procédure ci-dessous :

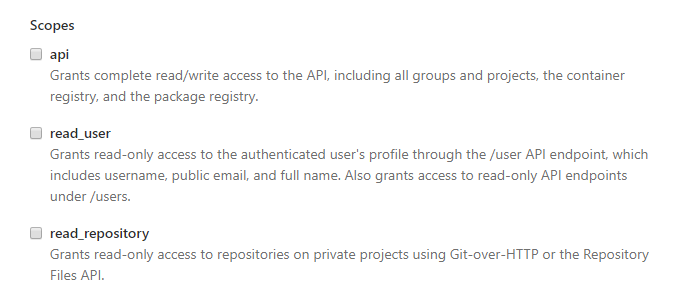
1. Connecter sur GitLab
2. Dérouler le menu sur votre avatar en haut à droite et cliquer sur l’option **Settings**

****

1. Choisir ensuite l’option **Access Tokens** sur le menu à gauche

****

1. Choisir le **nom** et la **date d’expiration** de votre token
2. Choisir ensuite les options d’accès désirées sous le menu **Scopes**

****

1. Cliquez ensuite sur le bouton **Create personal access token**
2. Enregistrer votre token à un endroit sûr, car une fois la page actualisée vous ne pouvez plus le visualiser.
3. Rendez-vous sur votre terminal Git et tapez la commande

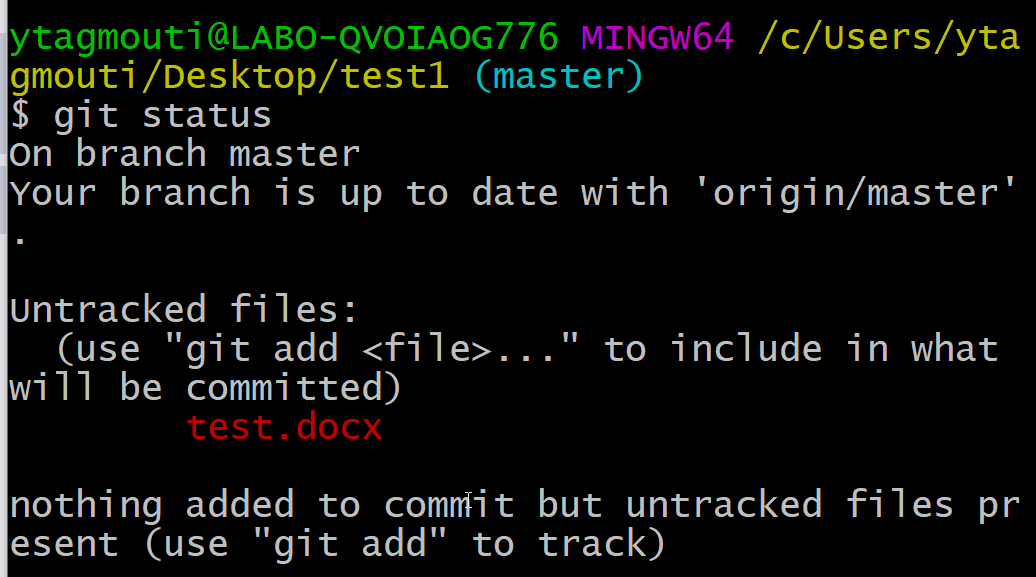
**git clone https://gitlab-ci-token:VotreToken@gitlab.com/tagmouti.yousra/test1.git**

**L’adresse de votre projet**

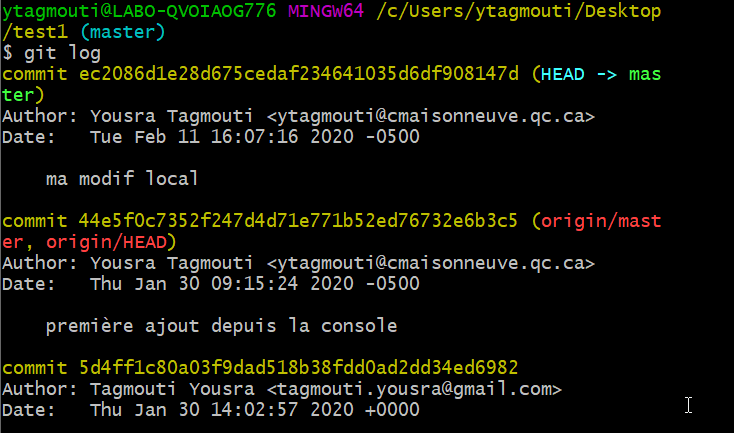
**Contribuer à un projet sur GitLab**

Une fois que vous avez cloner un projet de GitLab sur votre machine locale, vous avez la possibilité de travailler dessus, ensuite l’idée est de partager votre travail sur le Web … non pas sur Facebook ! mais sur **GitLab**.

1. Ajoutez au répertoire créé à l’étape précédente un fichier de votre choix, dans mon cas c’est le fichier Word **test.docx** tel que ci-dessous :
2. Si vous regardez maintenant le statut de votre projet, vous pouvez voir un changement, tel que ci-dessous :



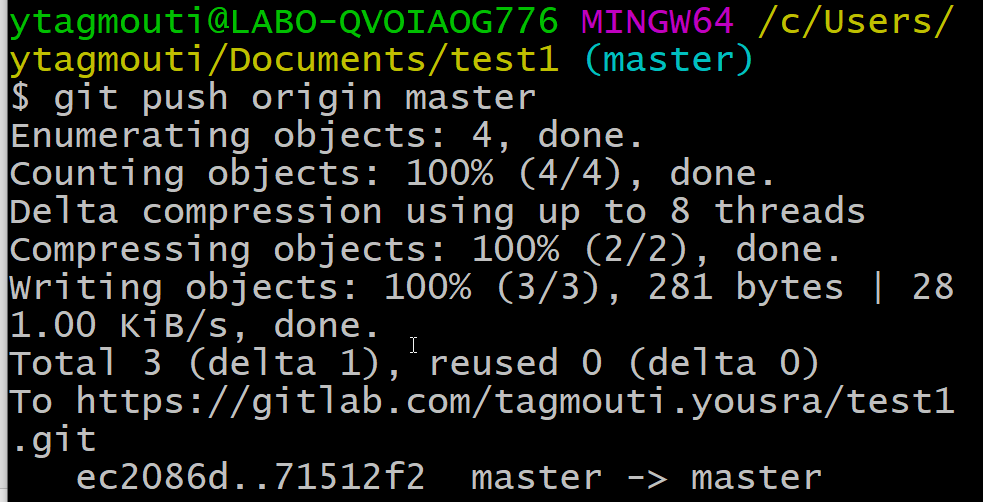
1. Tel que vous devez le savoir à ce niveau, vous devez faire un **git add** et ensuite un **commit**
2. Vous pouvez voir vos commits avec la commande **git log**, tel que ci-dessous :



1. Vous pouvez envoyer vos changements sur GitLab avec la commande ci-dessous :

**git push origin master**

* **origin**: est le nom donné au remote par défaut, ceci dit nous pouvons créer d’autres remote si nous voulons envoyer nos changements sur d’autres destinations
* **master**: est le nom de la branche par défaut sur laquelle vous travaillez

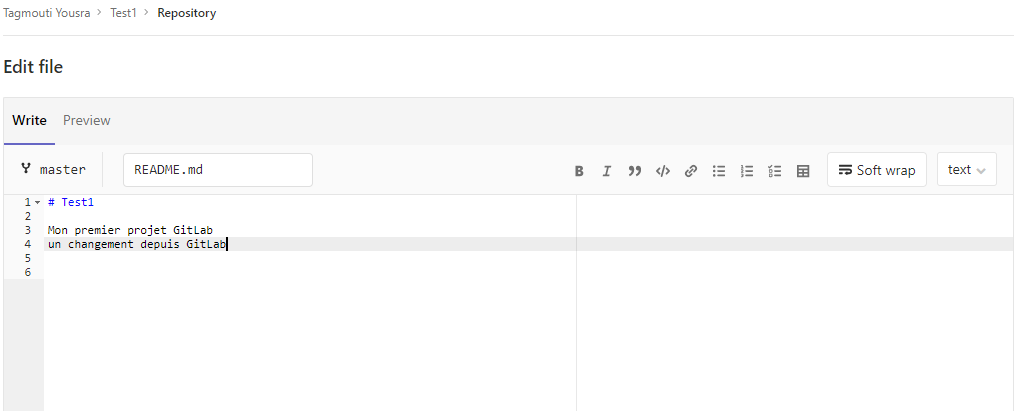


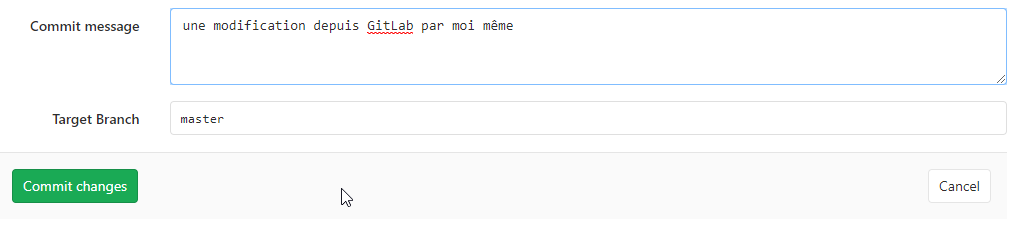
1. Maintenant vous pouvez voir vos contributions sur votre projet GitLab

**Récupérer les changements sur GitLab**

1. Vous pouvez récupérer sur votre dépôt local les changements faits par exemple par une autre personne sur le projet
2. Vous pouvez faire une simulation de ce scénario en modifiant par vous-même un document du projet directement sur GitLab, tel que le fichier **README** par exemple.
   1. Cliquer sur le fichier **README** ensuite sur le bouton **Edit**
   2. Écrire un texte de votre choix et remplir la section commit

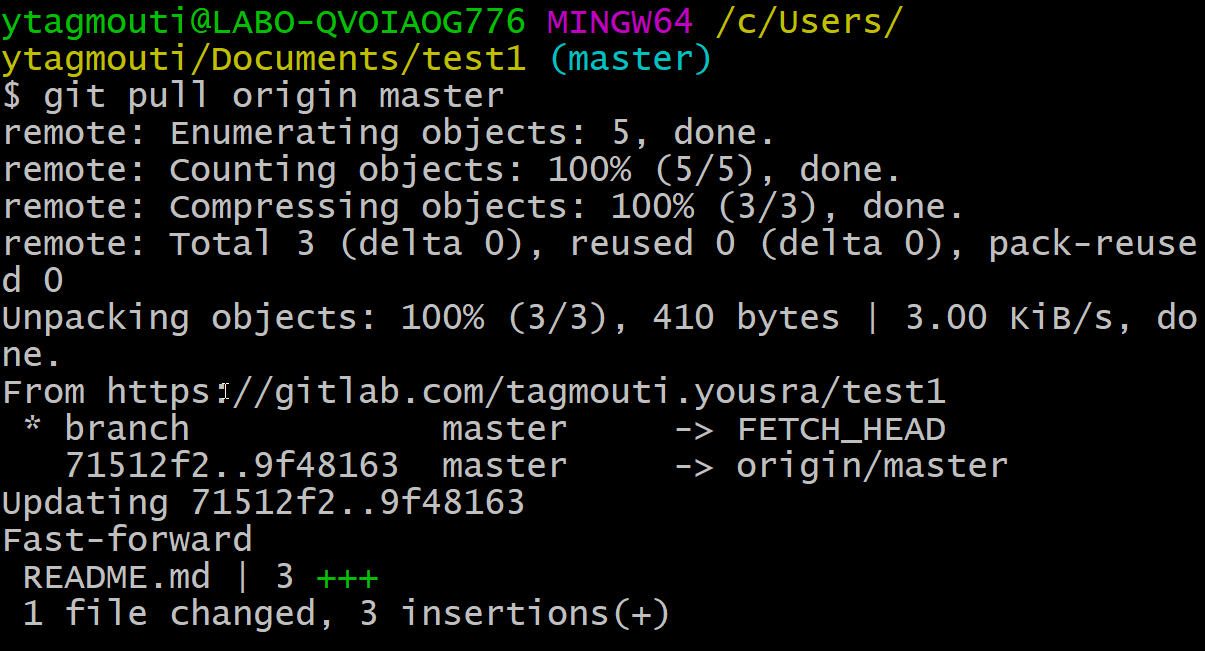
*Et oui ce que vous faites c’est bien un commit !!!*



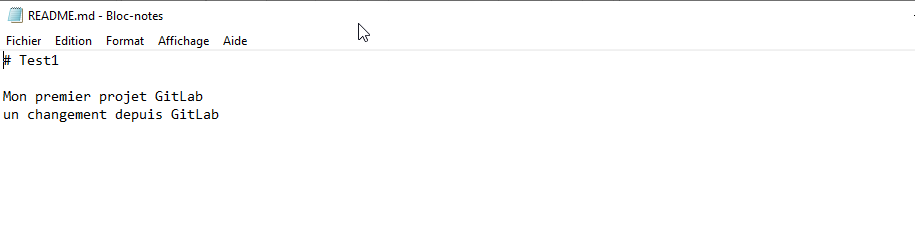


1. Pour pouvoir récupérer les modifications du fichier README sur votre dépôt, vous devez utiliser la commande **pull** :

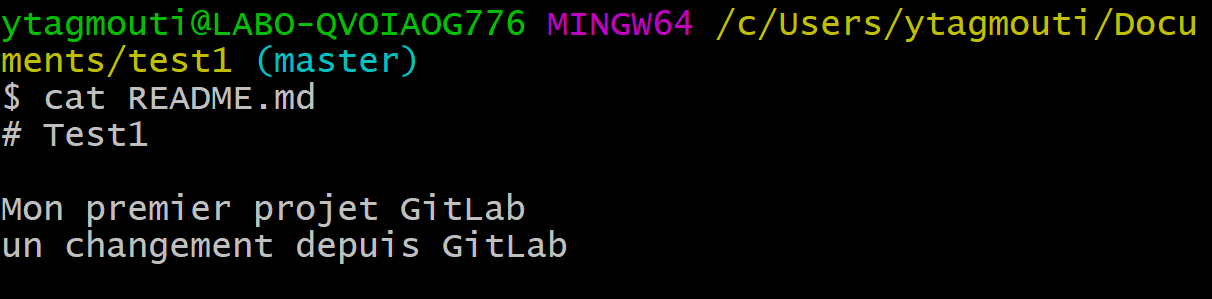
**git pull origin master**



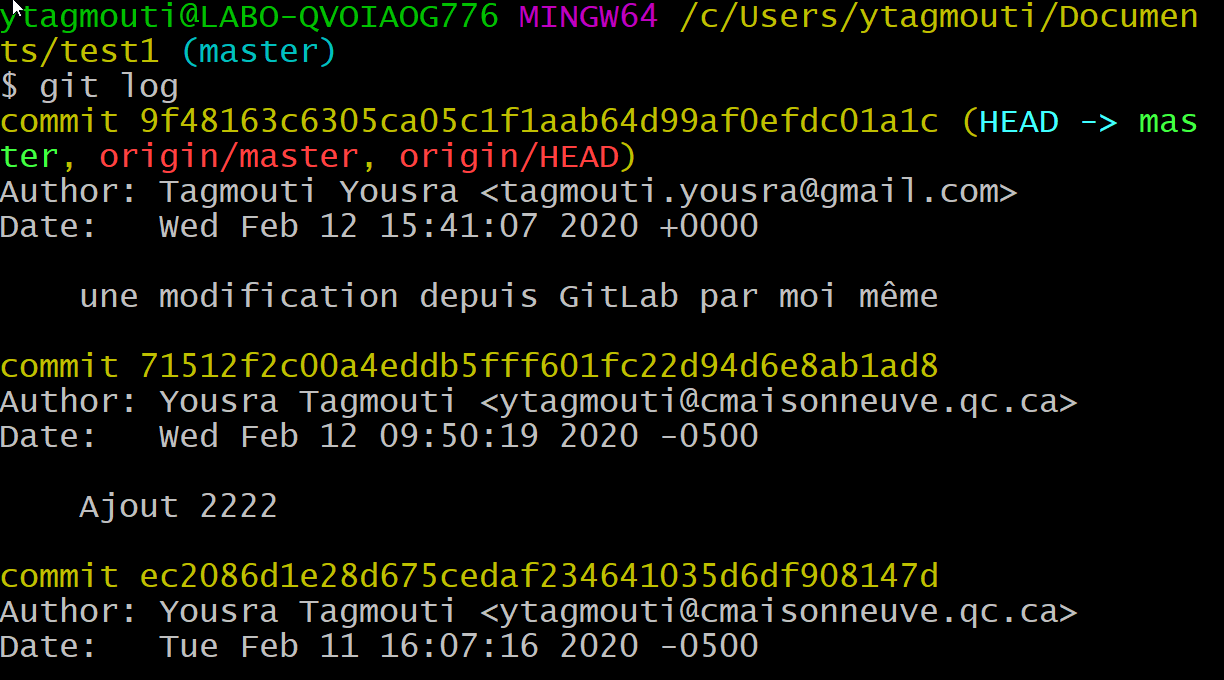
1. Vous pouvez voir les changements sur le fichier README de votre dépôt local



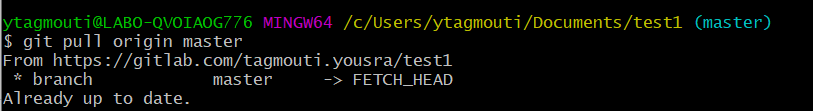
1. Vous pouvez faire la même vérification avec la commande **cat README.md**



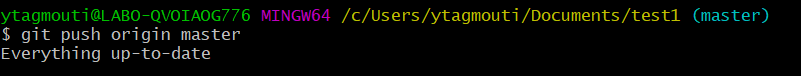
1. Vous pouvez aussi voir que le nombre de commits a également changé avec la commande **git log**



1. Si maintenant vous faites un **git pull**, vous avez un message qui vous informe que vous êtes à jour, tel que ci-dessous :



Vous avez la même information si vous faites un **git push**, tel que ci-dessous :

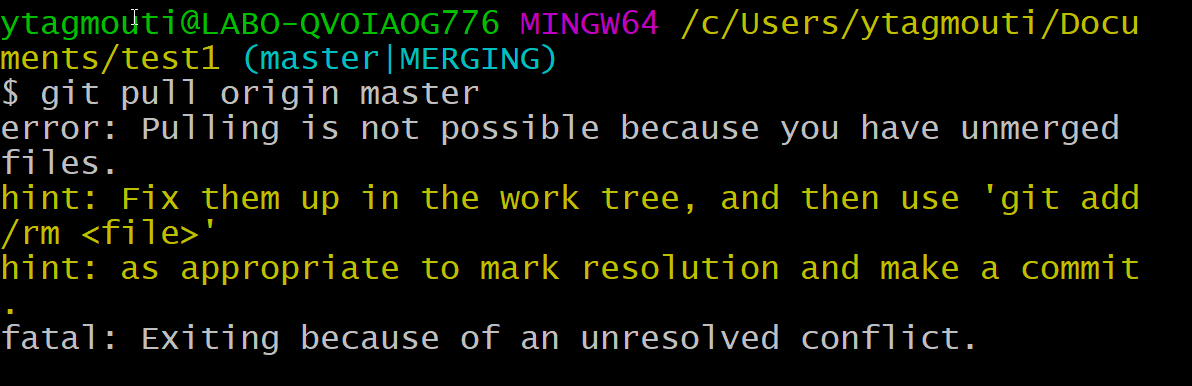


**Résolution de conflit**

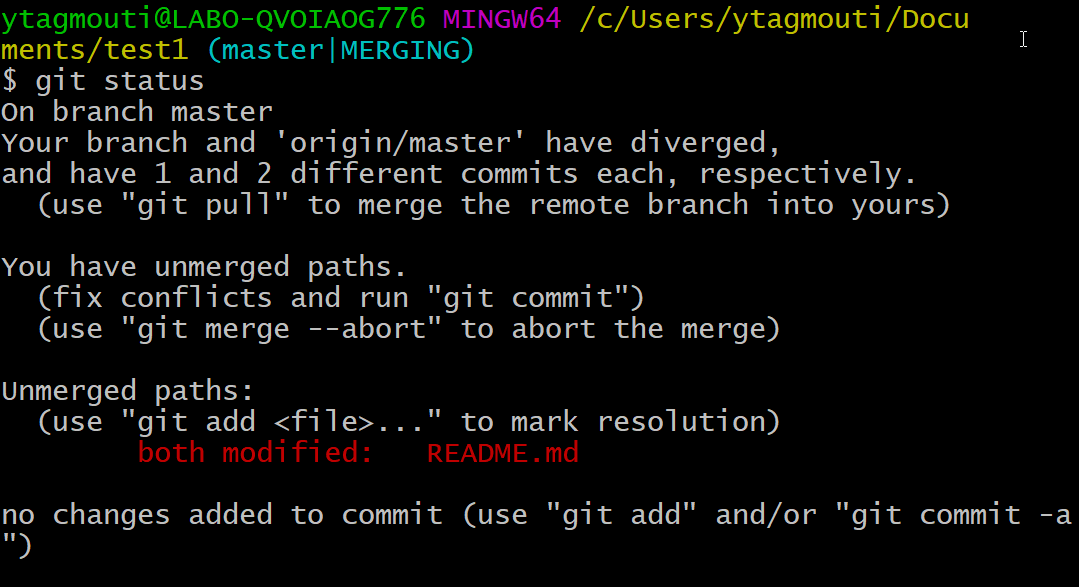
Si vous travaillez à plusieurs sur le même fichier et les mêmes lignes et que vous voulez faire une fusion (merge) via le **pull**, il y a un risque de conflit au moment de la fusion.

**Simulation d’une situation de conflit**

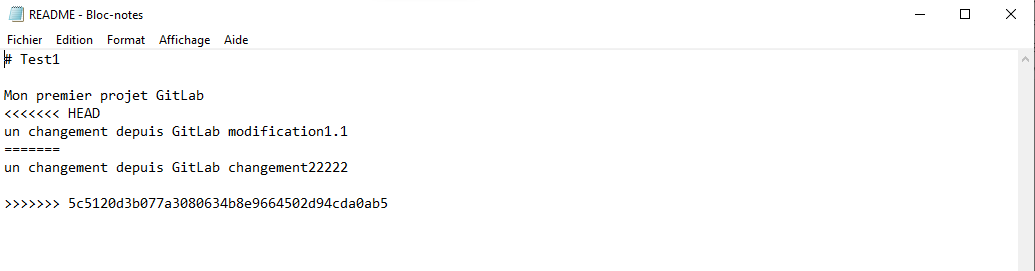
1. Sur le fichier **README.md** du répertoire de votre projet local, modifiez la dernière ligne du fichier
2. Mettre votre statut à jour avec un **add** et **commit**
3. Faire la modification sur la même ligne depuis GitLab et faire un commit de la modification
4. Essayez de faire un **pull** du contenu de GitLab avant de faire un **push** pour partager votre travail sur GitLab. Ceci affiche un message de conflit indiquant qu’il ne peut faire le merge à cause d’un conflit sur un fichier, tel que ci-dessous par exemple :



1. Vous pouvez avoir plus d’informations sur le conflit avec la commande git status



1. Maintenant si vous ouvrez le fichier **README.md vous allez voir que Git vous a laissé un petit message.** Tel que ci-dessous par exemple :



1. Il faut donc éditer manuellement le fichier à la cause du conflit. Une fois le conflit résolu il faut faire un **git add** et un **git commit**.
2. À ce niveau un **git pull** vous indique que tout est à jour