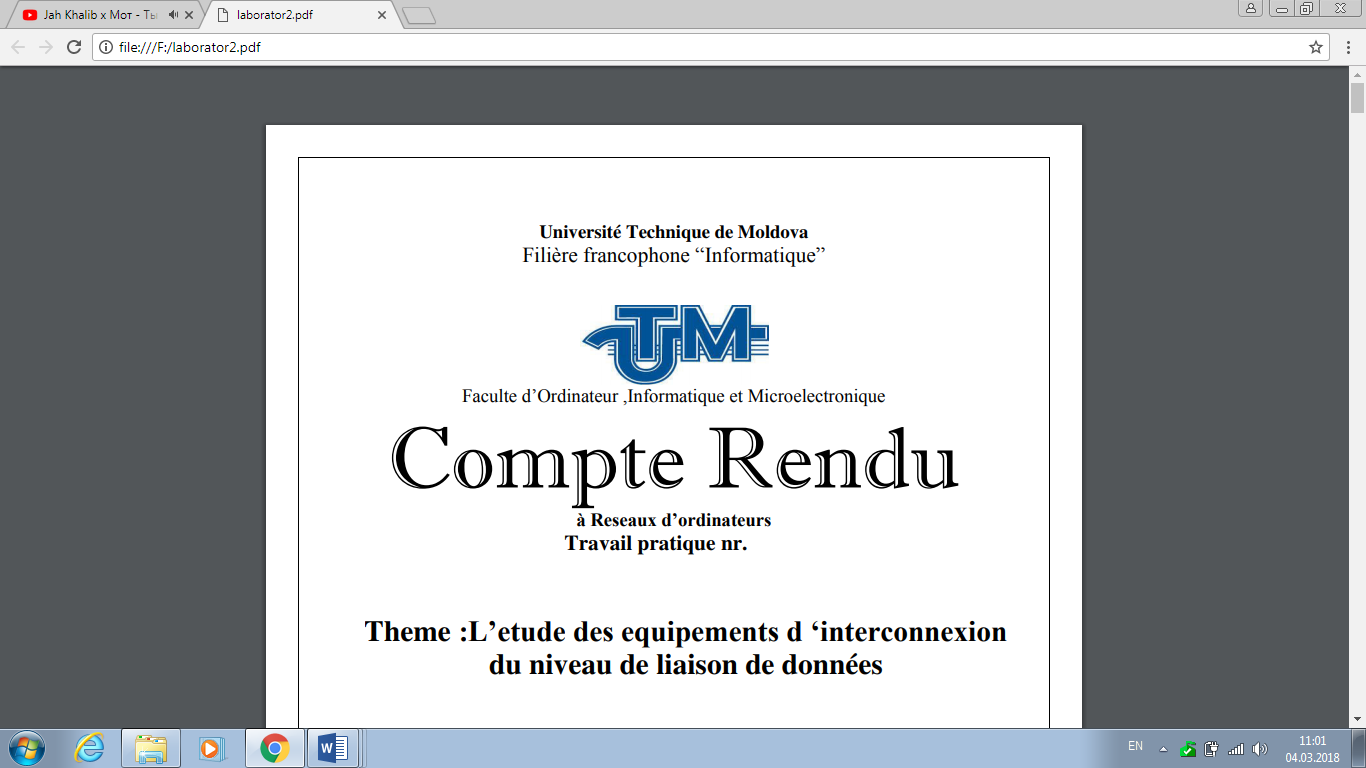
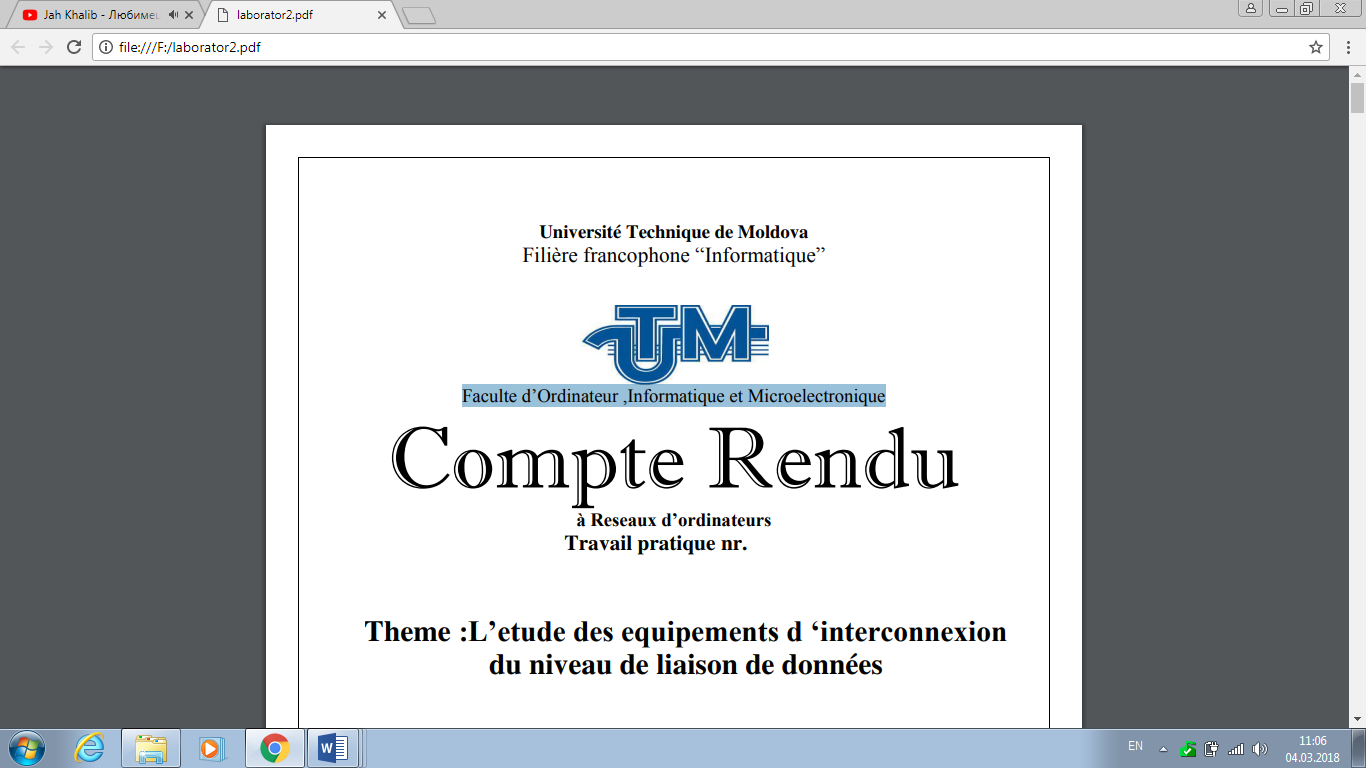
Ministère de l'éducation de la République de Moldova

Université technique de Moldavie

Faculté d’Ordinateur, Informatique et Microélectronique

Filière francophone “Informatique”





**TIDPP**

**Travail pratique nr.1**

**Thème:   
Utilisation du GIT pour suivi des modifications et de la contrôle de version**

Effectué par l’étudiant(e) de gr FI-181 : Damean Alexandra

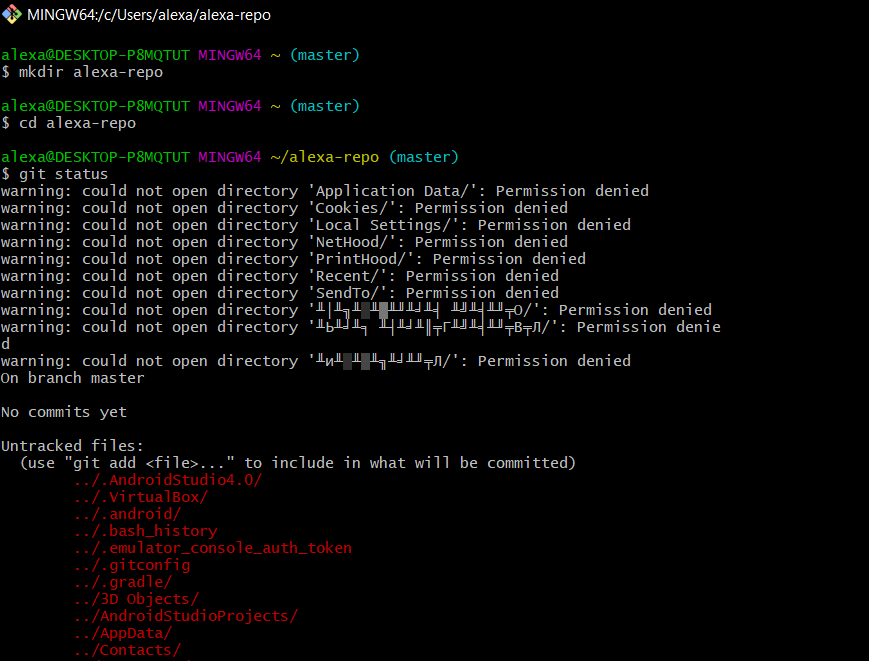
Vérifié par le professeur : Rusu Viorel

Chișinau 2020

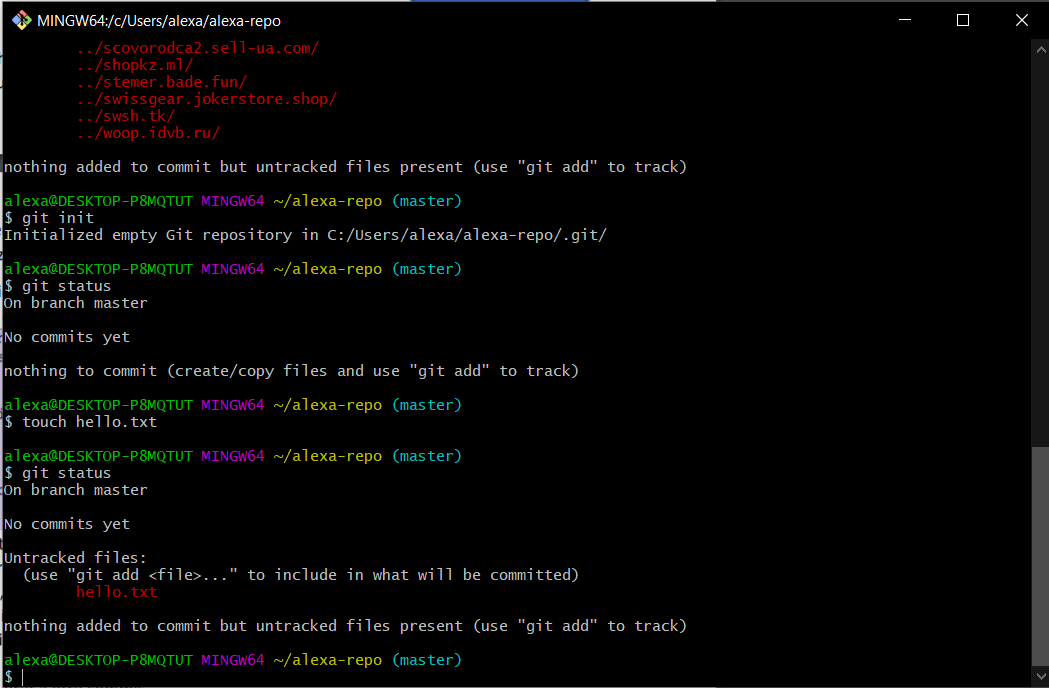
**But : Travailler avec Git Localement. Travailler avec les branches Git localement. Travailler avec Git et un référentiel distant**



**I. Travailler avec Git Localement**



L'erreur s'est produite car je n'ai pas initialisé le référentiel.



Du point de vue de Git, les fichiers peuvent se trouver dans l'un de ces états :

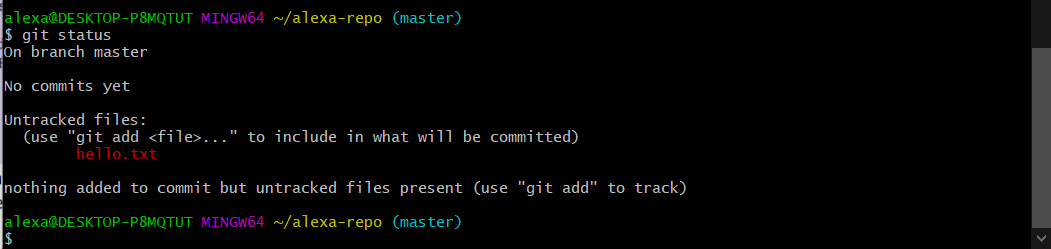
**untracked**: Git ne suit pas encore ce fichier, indiquant généralement que le fichier est nouveau.

**tracked**: le fichier fait partie de l'instantané Git et Git suit ses modifications.

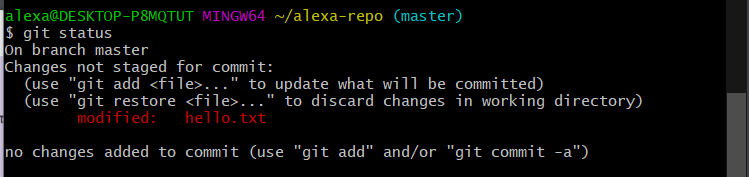
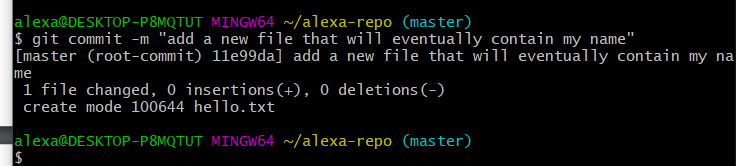
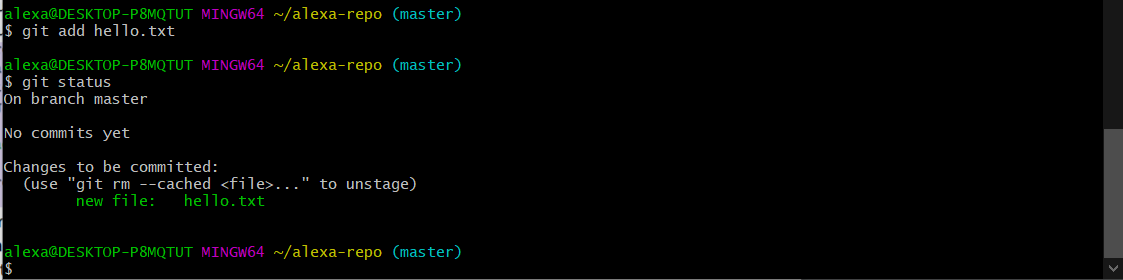
**unstaged**: un fichier suivi a des modifications mais il n'a pas été mis en scène pour une validation.

**staged**: le fichier est prêt à être commis.

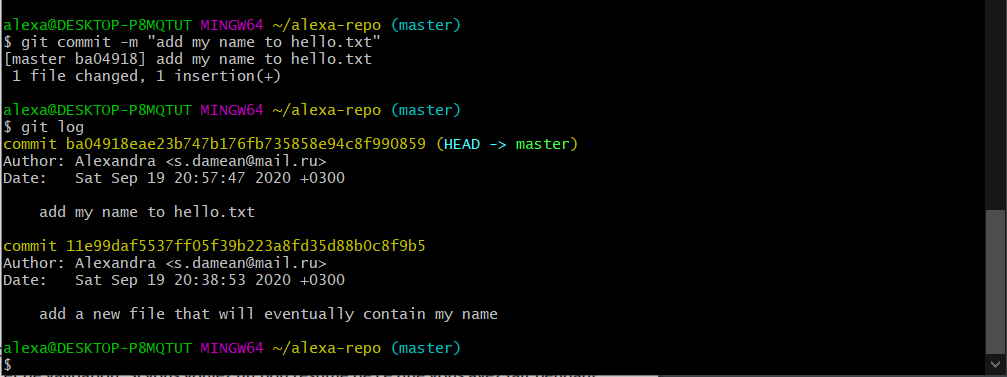
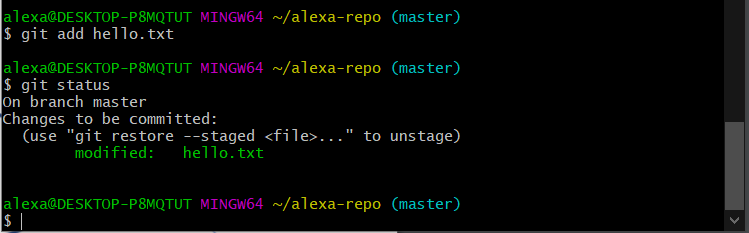
**deleted**: le fichier a été supprimé et doit être supprimé de Git.



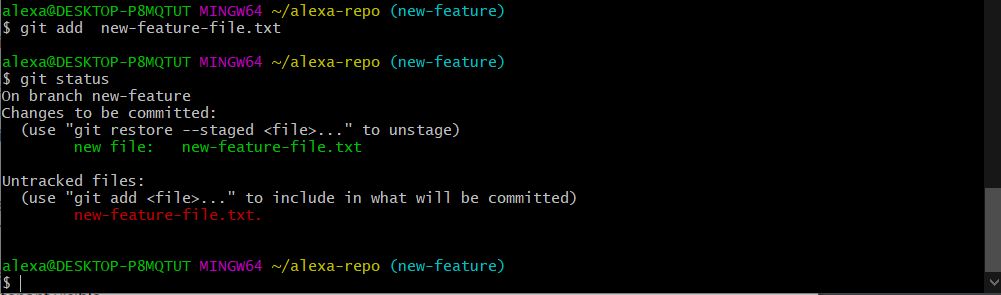
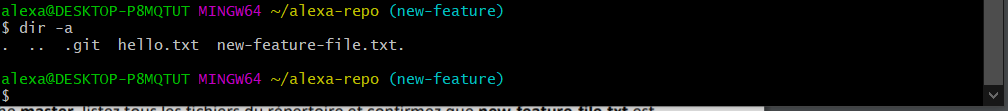
Pour vérifier l'état du référentiel, nous utilisons la commande *git status*.

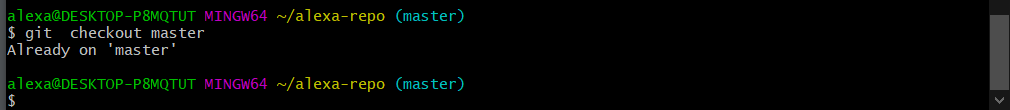


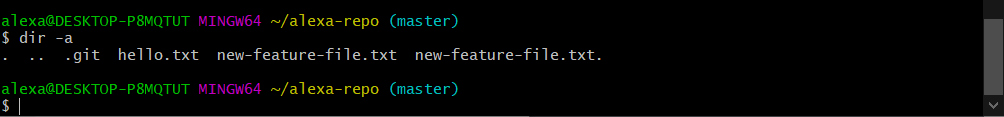
Le format de la commande pour valider les modifications apportées à une branche est *git commit -m "<message de validation>".*



**II. Travailler avec les branches Git localement**



****

****



**III. Travailler avec Git et un référentiel distant**

 **git status** - checks the state of a repo

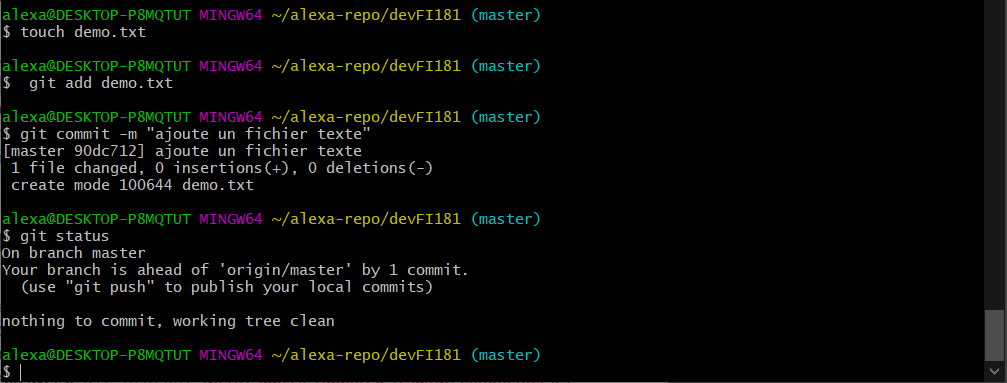
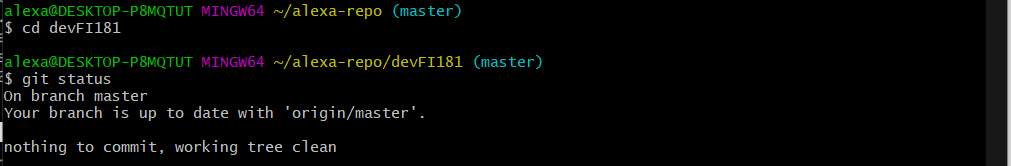
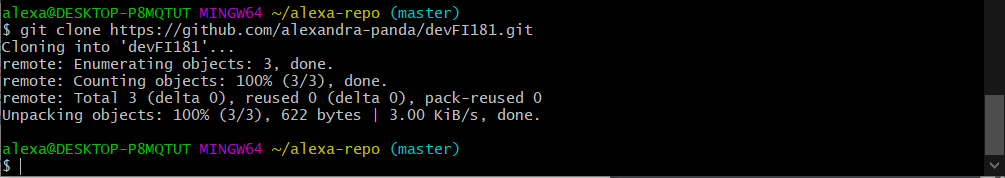
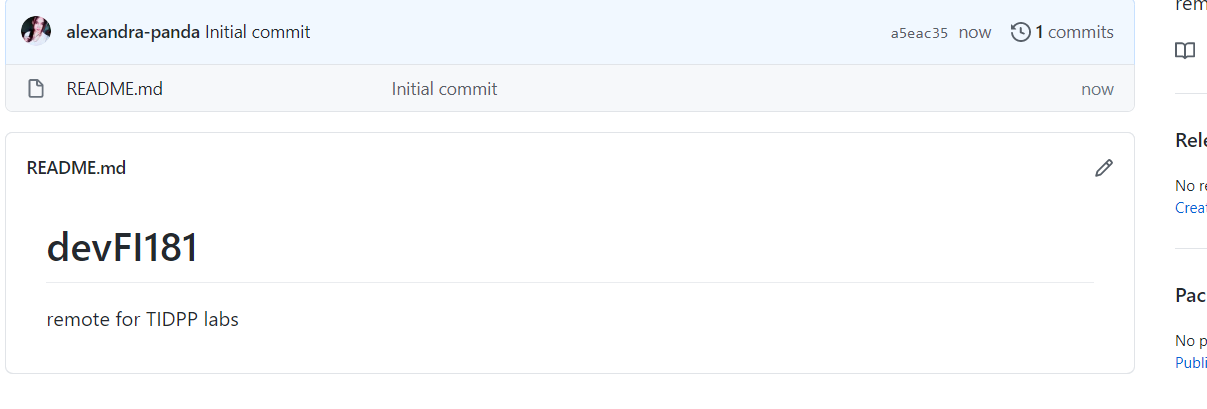
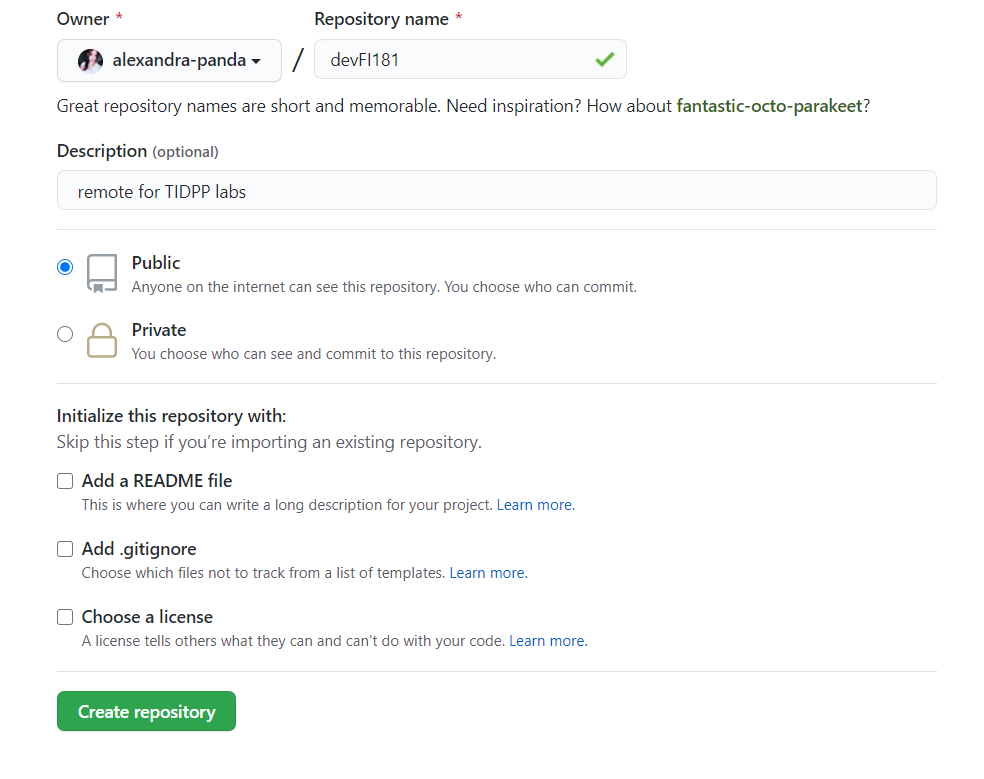
 **git add** - stages updates

 **git commit** - commits updates

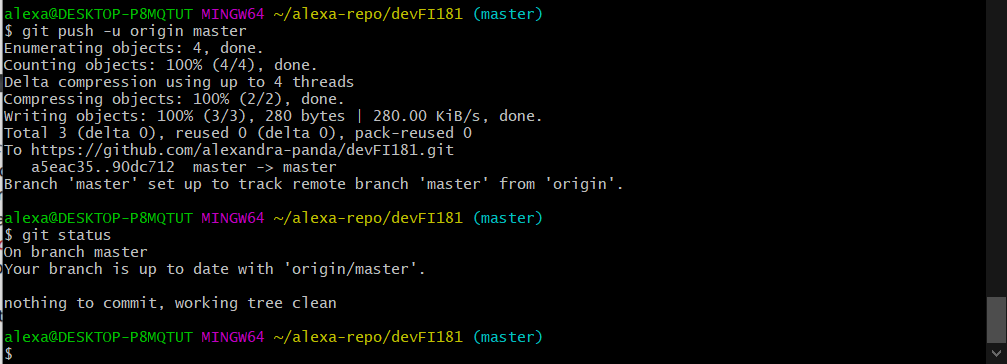
 **git checkout** - switches branch

 **git branch** - lists existing branches

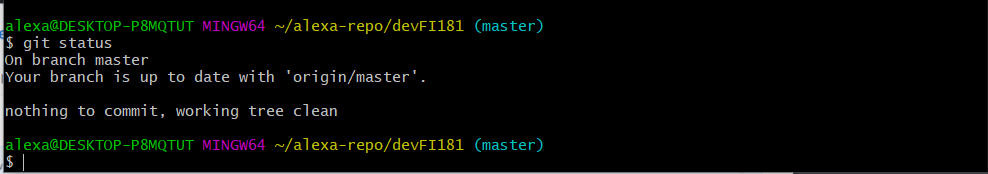
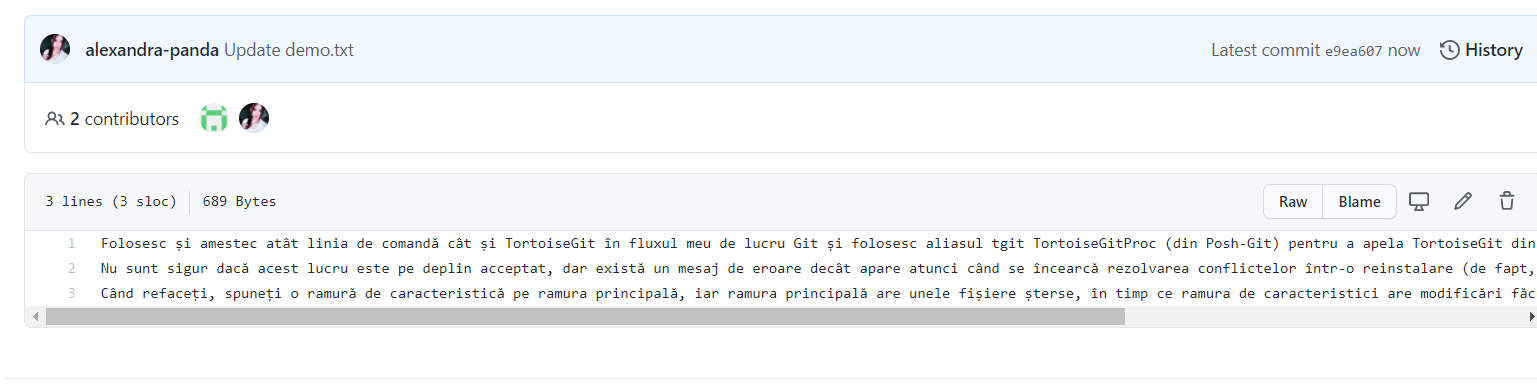
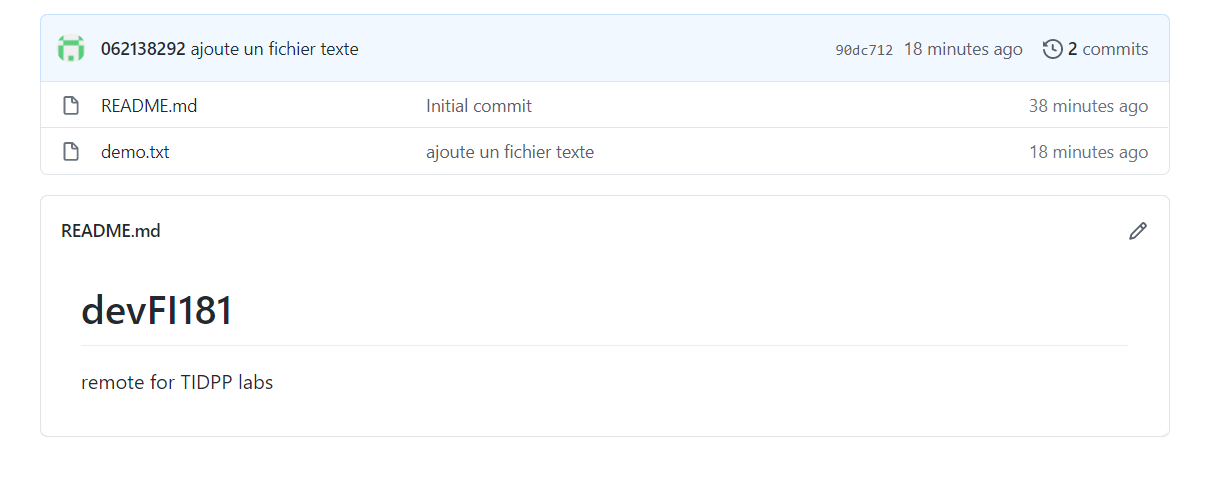
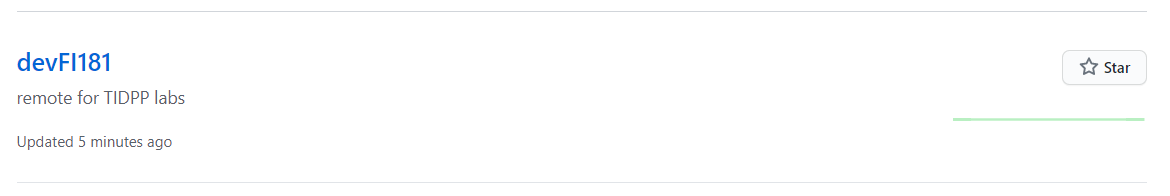
 **git log** - shows commit history



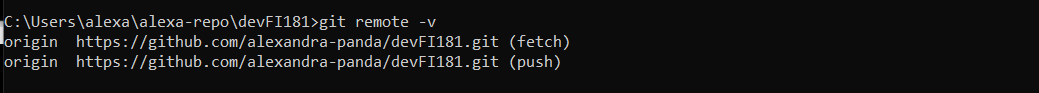
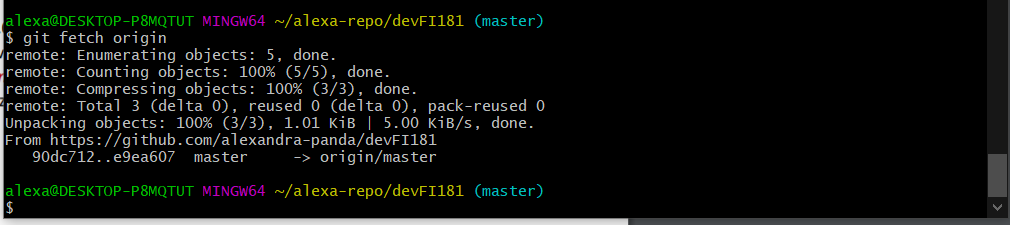
Je dois utiliser la commande *git push* pour apporter des modifications au référentiel.



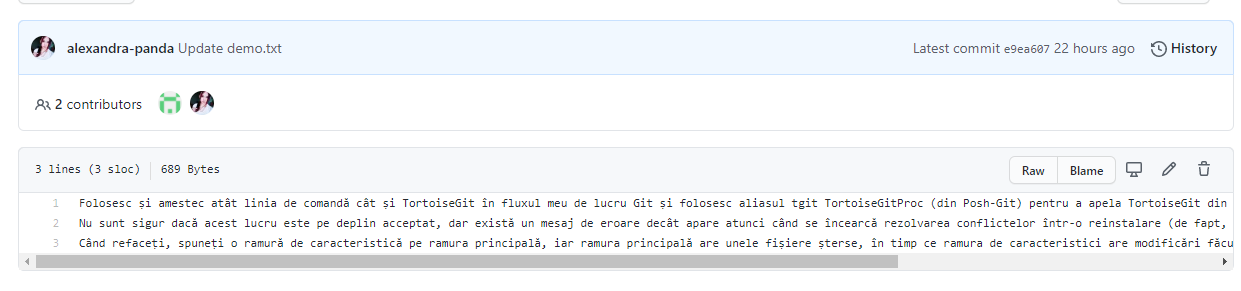
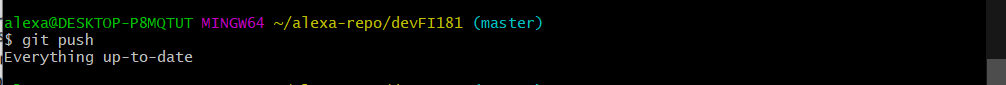
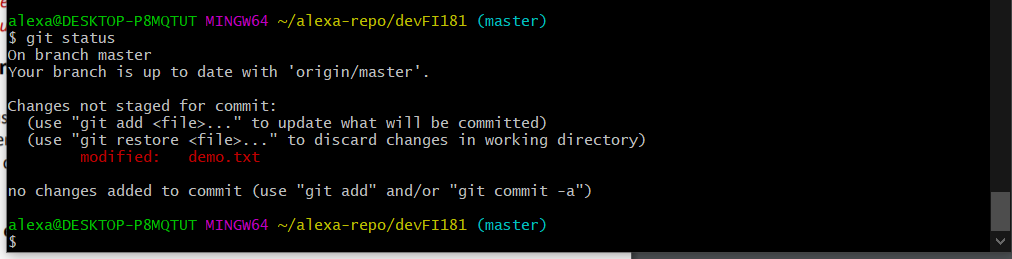
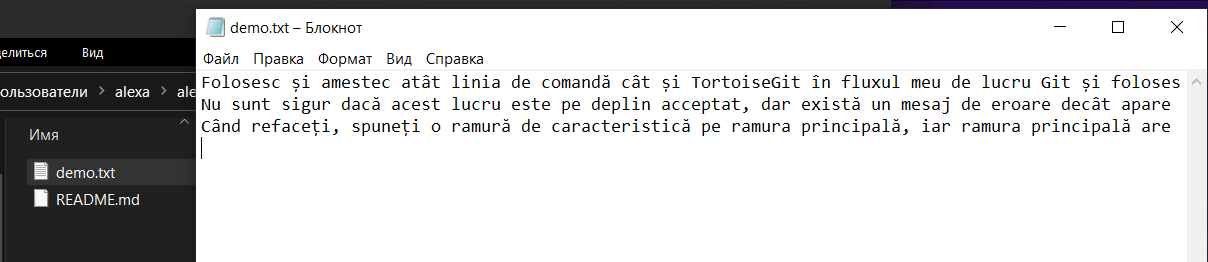
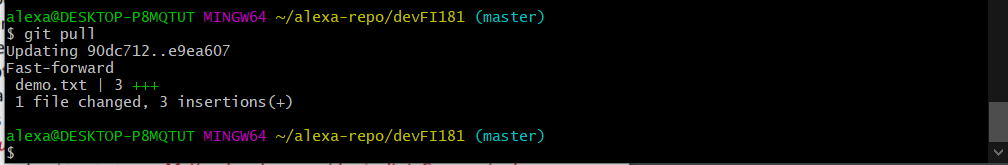
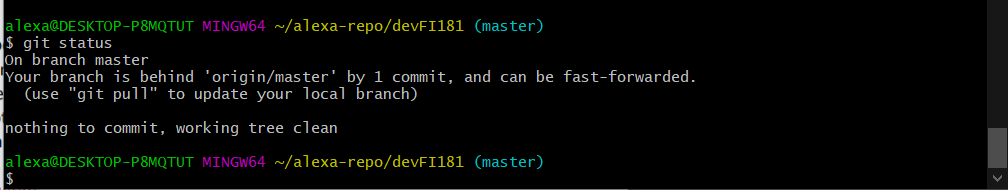
Les modifications sont transmises à *origin master*.



La commande *fetch* met à jour essentiellement votre branche de suivi à distance, dans ce cas d'origine.

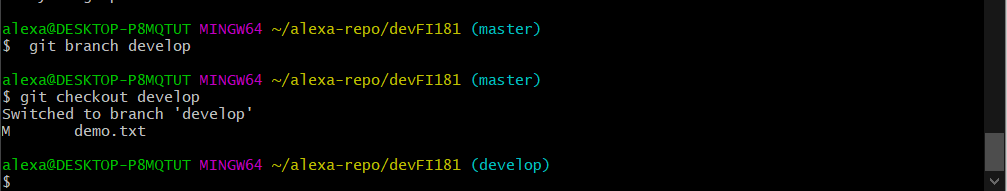


Cette commande fournit des informations sur notre référentiel.

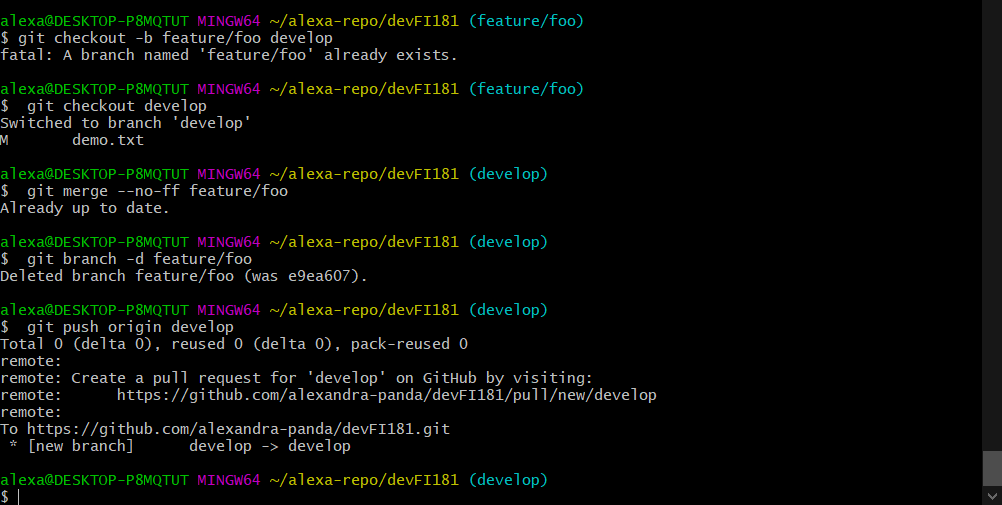
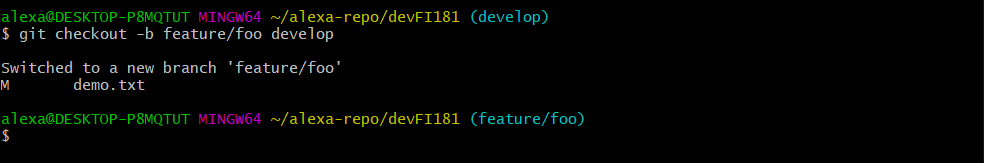


**IV. Adopter un modèle de versionnement efficace avec Git**

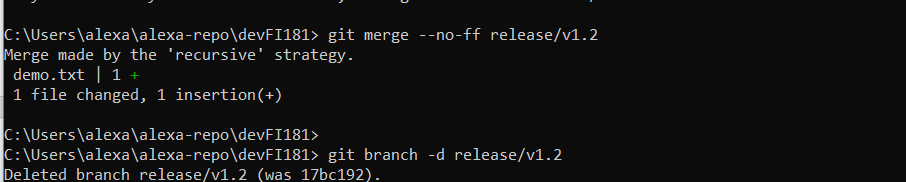
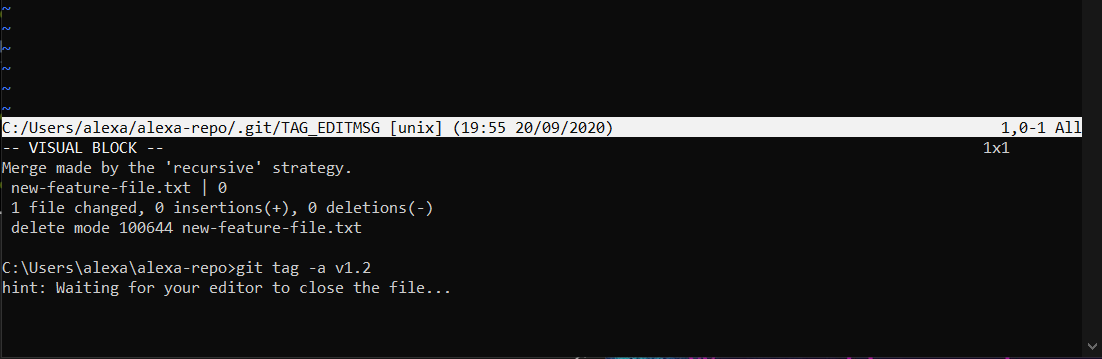
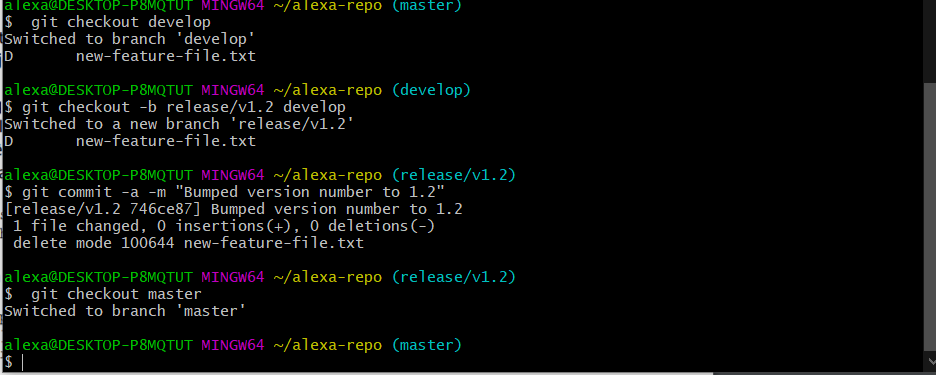
1. **Une branche de production saine et cloisonnée**



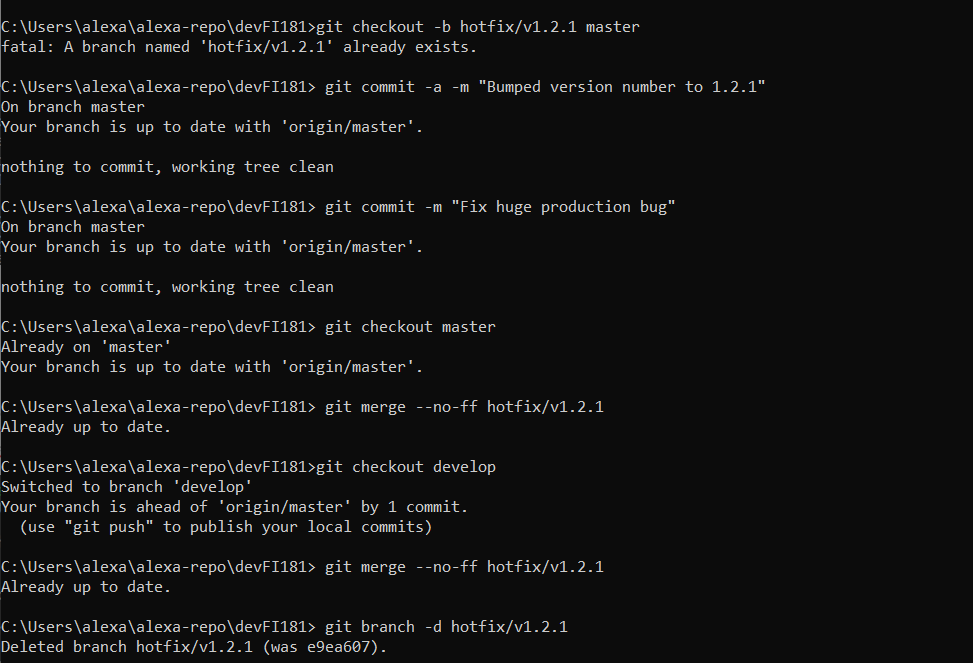
1. **Des branches spécifiques pour les développements lourds**



1. **Passage en production**

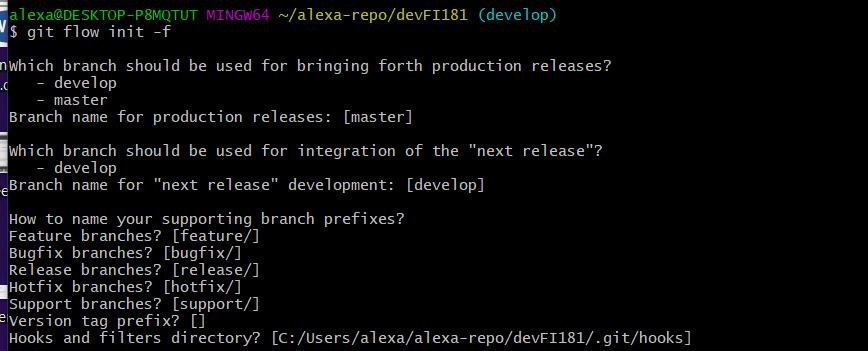


1. **Dépanner les bugs critiques en production**

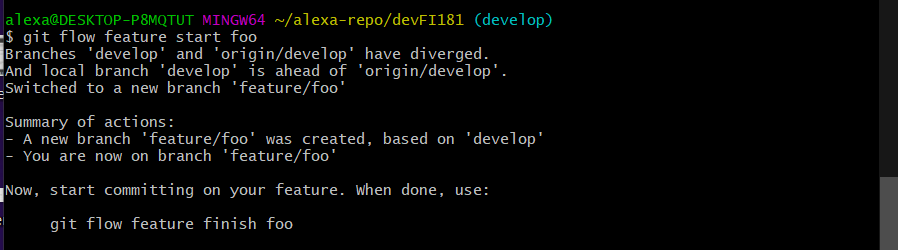


1. **Git-flow**

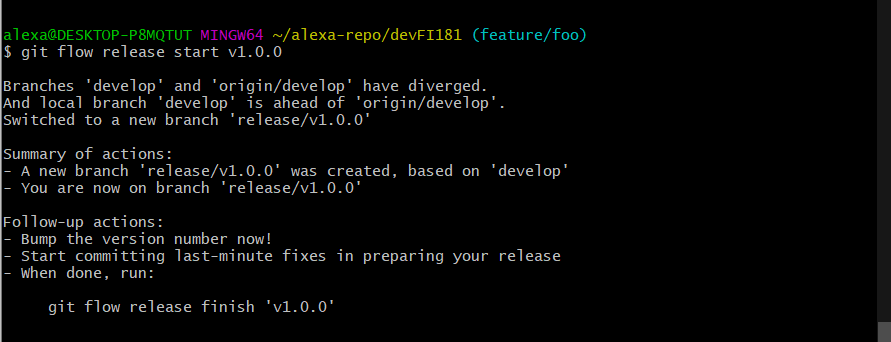
**Git-flow** est un ensemble d’extensions pour Git livré sous forme de shell-scripts très simples d’accès. Git-flow va permettre de suivre le workflow présenté précédemment sans avoir à tout retenir, surtout si vous êtes un peu perdu avec toutes les commandes et options de Git.



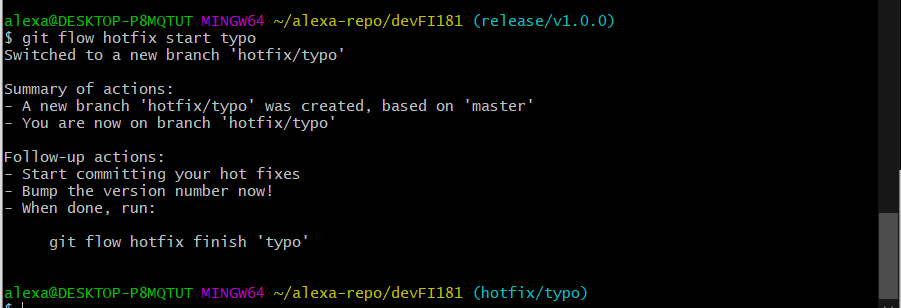
1. **Créer une branche de feature**



1. **Mise en production**



**9. Hotfix**



**Conclusion:** Dans ce laboratoire, j'ai étudié Git Localement, le branch Localement et une référence distante. J'ai appris de nombreuses commandes de Git et étudié les principes du fonctionnement du référentiel. J'ai eu des problèmes avec certaines commandes, mais j'ai trouvé les réponses dans le guide fourni par le professeur.