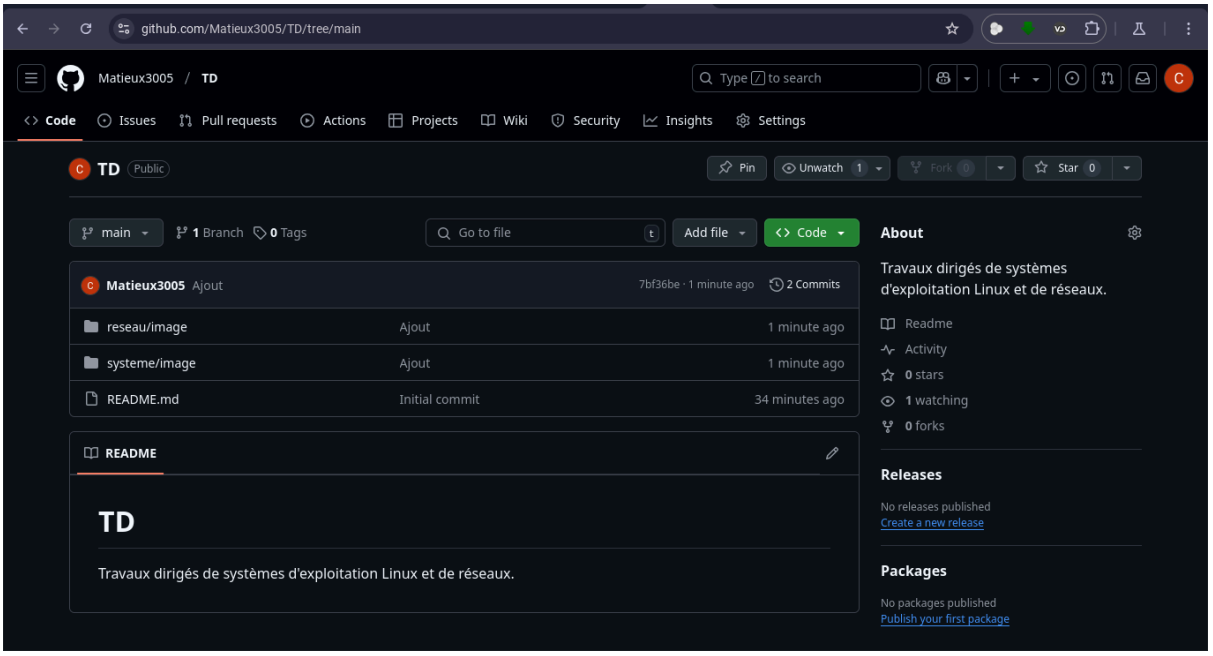
Travaux Dirigés Git et GitHub

Dans ce travail dirigé, j’ai appris à utiliser Git et GitHub pour gérer un projet et suivre mes fichiers de manière organisée. J’ai commencé par créer un dépôt sur GitHub, puis je l’ai cloné sur mon ordinateur afin de travailler dessus en local. Ensuite, j’ai structuré mon projet en créant des dossiers adaptés, ajouté mes fichiers à Git et enfin envoyé mon travail sur GitHub.

Création de mon dépôt sur GitHub

Avant de pouvoir utiliser Git sur mon ordinateur, il fallait d’abord que je crée un dépôt sur GitHub. Pour cela, je me suis rendu sur GitHub, j’ai cliqué sur "New Repository", et j’ai renseigné les informations nécessaires. J’ai choisi d’appeler mon dépôt TD, puisqu’il regroupe mes travaux dirigés sur les systèmes d’exploitation et les réseaux.



Clonage du dépôt sur mon ordinateur

Une fois mon dépôt créé sur GitHub, je devais le récupérer en local pour pouvoir y ajouter mes fichiers. J'ai donc ouvert mon terminal et exécuté la commande suivante :

Grâce à cette commande, mon ordinateur a téléchargé une copie du dépôt, créant un dossier du même nom (`TD`).

Organisation des fichiers et des dossiers

Plutôt que de laisser tous mes fichiers en vrac, j'ai préféré créer des dossiers pour mieux structurer mon projet. J'ai décidé de séparer mon travail en deux grandes parties :

- Un dossier systeme, qui contient les fichiers liés aux systèmes d'exploitation.
- Un dossier reseau, pour tout ce qui concerne les réseaux.

Dans chacun de ces dossiers, j'ai aussi créé un sous-dossier image afin d'y ranger les captures d'écran et autres illustrations.

J’ai utilisé ces commandes pour les créer directement depuis le terminal :

Ajout des fichiers au suivi Git

Une fois mes dossiers créés, j’ai voulu les enregistrer avec Git pour suivre leur évolution. Pour voir quels fichiers étaient pris en compte. Git m’a alors indiqué que mes nouveaux dossiers n’étaient pas encore suivis. J’ai donc utilisé une commande suivante pour ajouter tous mes fichiers, puis j’ai validé ces ajouts avec un **commit**, ce qui permet de sauvegarder l’état actuel du projet. Pour que mon travail soit disponible en ligne et synchronisé avec GitHub j’ai dû envoyer mes modifications avec la commande git push.