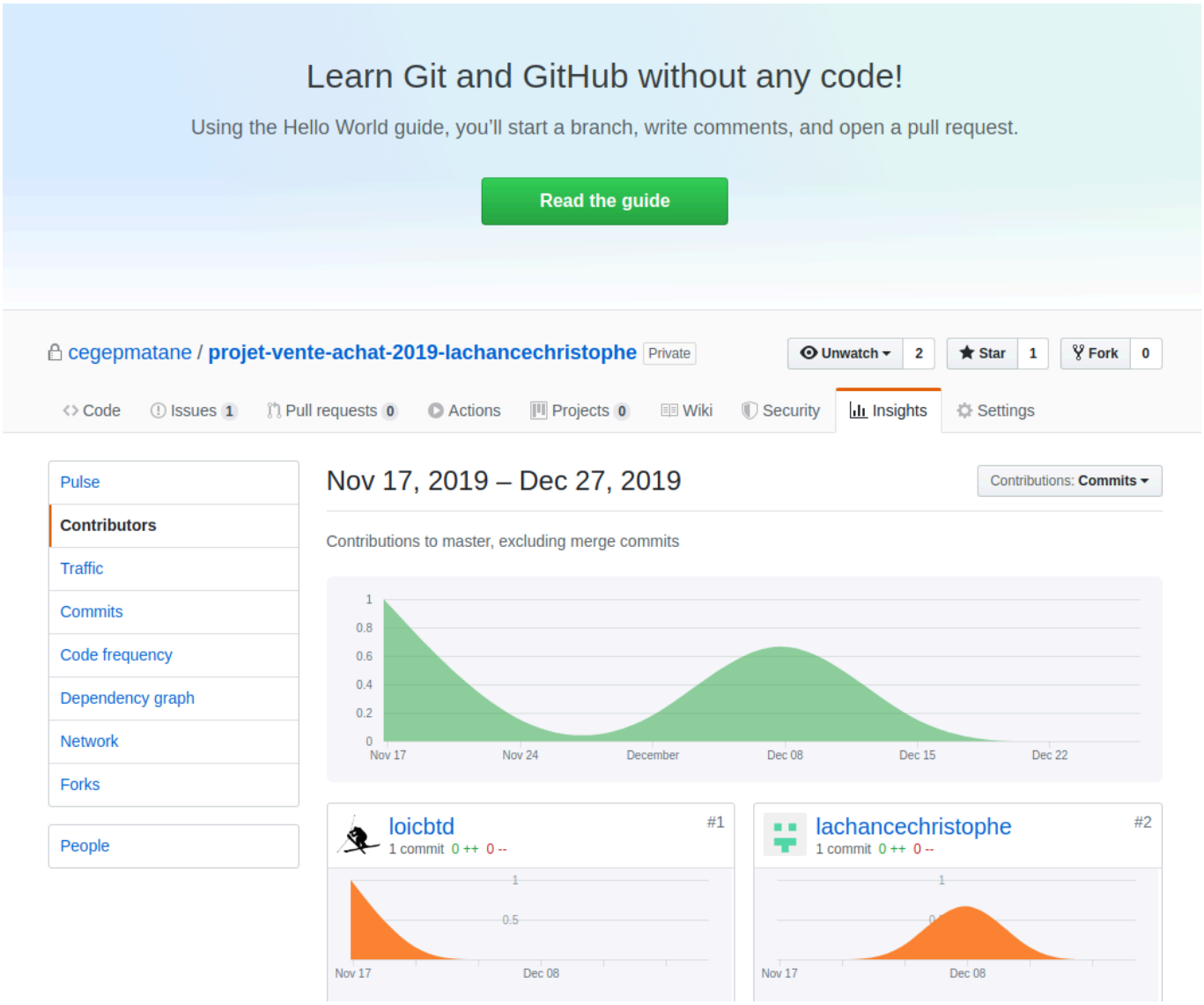
Normalement les branches servent à tester des alternatives au code actuel dans le but de peut-être l'intégrer au projet plus tard. On peut y tester des refactoring, des techniques alternatives d'accès aux données, des librairies graphiques différentes. Je vous assure que de diviser vos projets en branches, cela ne se fait pas dans l'industrie, ce ne sont pas des répertoires mais des alternatives - des diff existent entre les branches vous savez et les branches sont destinées à se merger de par la structure de l'outil…

Si vous vouliez absolument vos petits projets séparés, vous auriez pu accepter l'assigment plusieurs fois (une fois par participant) et produire vos applications dans des projets github séparés ce qui n'aurait cependant pas arrangé le versionnement des versions compatibles. Sinon il faut utiliser les sous-répertoires - avec l'aide d'un client GitHub intégré au système d'exploitation (Tortoise, ligne de commande). Mais jamais les branches lol. Une branche c'est une copie du projet existant avec des modifications alternatives, non-approuvées. De par les normes du cours fournies avec l'évaluation, les branches étaient proscrites !

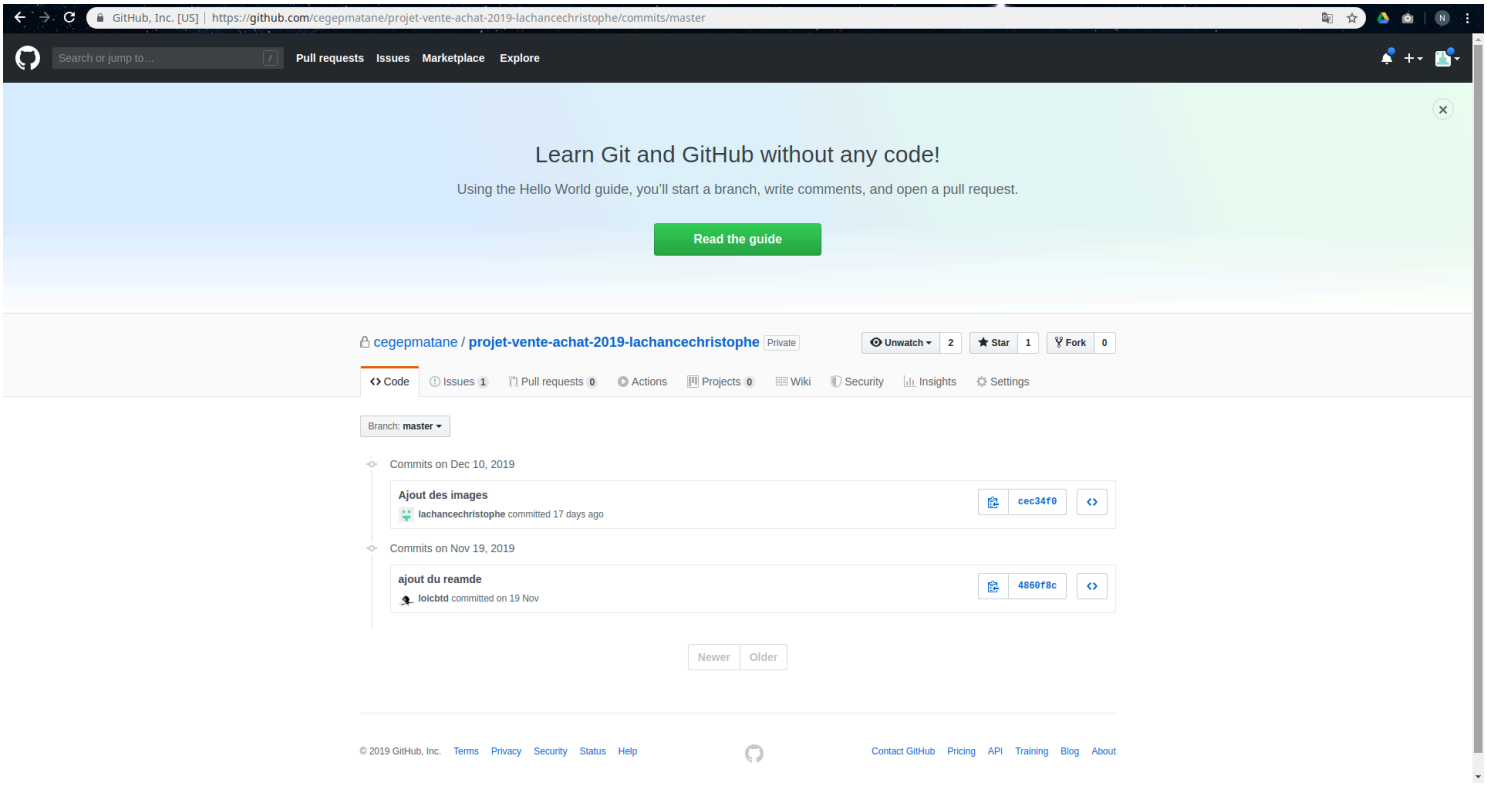


Voici les problèmes qui existent pour moi avec votre organisation en branches.

1 - Je ne peux pas voir le nombre de commit total du projet ni la liste des commits
2 - Les graphiques de contribution des participants ne fonctionnent plus, les commits ne sont pas encore intégrés au projet principal (trunk) donc ils ne comptent pas. Je ne sais même pas qui a travaillé sur ce projet !!!

3 - Je ne peux naviguer le projet en entièreté dans sa révision pré-cloud (tag LIVRAISON_MAGASIN ou LIVRAISON_BOUTIQUE demandée)
4 - Quand on fait une RELEASE, seulement le trunk est zippé et taggé, les branches n'existent pas.
5 - Enfin avant-dernier problème et le plus grave de tous, la synchronicité (compatibilité) des vos applications entre elles n'est plus maintenue. Autrement dit, lorsqu'on programme un client et un serveur, ils sont compatibles au moment ou ils sont versionnés. Quelques versions plus tard on les modifie et ils sont encore compatibles entre eux mais ils ne sont plus compatibles avec les anciennes versions.. Je dois pouvoir reculer à tout moment du projet et pouvoir tester les applications entre elles, c'est ce à quoi sert un outil de versionnement - à maintenir des versions de fichiers (ou de sous-projets) qui marchent entre eux.
6 - Enfin dernier problème, les branches (qui durent plus de quelques secondes) n'encourage pas l'intégration continue d'une manière générale, c'est une mauvaise pratique qu'il faut réserver à des cas d'exception approuvé par l'architecte (ou le professeur). C'est souvent utilisé par des participants tentant d'éviter le travail d'équipe.

Lorsque j'ai regardé votre projet j'ai failli mettre zéro (yay pour moi c'est pas long à corriger) , car j'ai vu 2 commits et 2 contributeurs (Loic et Christophe) mais les remises des améliorations (projet 4) vous ont sauvés de justesse.



Donc les branches sont une manière inappropriée de diviser les sous-projets !

Alors à quoi servent les branches ?

- Un exemple idéal d'utilisation de la branche aurait pu être le DAO alternatif vers l'infonuagique. Dans cet exemple le même DAO est en NoSQL dans le trunk et en Firebase dans la branche.

- Certaines compagnies imposent aux junior de faire leurs modifs dans des branches et de merger après approbation du code proposé, généralement en lien avec un ticket.