

Migration de tous mes dépôts de développement vers un Gitlab

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 décembre 2019

<https://www.bortzmeyer.org/github-to-gitlab.html>

Je viens de terminer la migration de tous mes dépôts de développement logiciel actifs (et quelques inactifs) vers un GitLab, suite au rachat de GitHub par Microsoft, en 2018.

Quand j'avais commencé à mettre des dépôts sur une forge publique, c'était sur SourceForge, en 2000. Après que SourceForge ait sombré, techniquement et politiquement, je suis passé à GitHub <<https://www.bortzmeyer.org/github.html>>, à l'époque une petite entreprise sympa et techniquement pointue. Puis GitHub a grandi, devenant le « Facebook des geeks ». Et, en juin 2018 <<https://www.theverge.com/2018/6/4/17422788/microsoft-github-acquisition-official-deal>>, Microsoft annonce le rachat de GitHub, le sort fréquent des petites entreprises cools. Il n'était évidemment pas question pour moi de rester chez Microsoft, l'entreprise symbole du logiciel privateur. Même si de nombreux Microsoft "fanboys" s'étaient à l'époque succédé sur les réseaux sociaux pour répéter les éléments de langage « Microsoft a changé », « Maintenant, c'est une entreprise sympa », « Il y a eu quelques erreurs mais maintenant, Microsoft ne traite plus le logiciel libre de cancer communiste <https://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/> », je n'ai jamais avalé ces arguments. D'ailleurs, Microsoft lui-même ne s'est jamais réclamé du logiciel libre, juste d'un vague « open source » <<https://www.bortzmeyer.org/free-software-open-source.html>>, qui veut tout et rien dire, et qu'on utilise quand le mot « liberté » fait peur.

Bon, c'est bien joli, cela, mais cela fait bien plus d'un an que le rachat a eu lieu, c'était pas rapide comme fuite hors de l'étreinte de Microsoft? C'est vrai, il m'a fallu du temps, la paresse a longtemps été la plus forte, mais j'ai fini par le faire.

Et la destination? La solution techniquement la plus proche de GitHub est un GitLab. Attention, beaucoup de gens confondent **un** GitLab (une instance spécifique d'un service utilisant le logiciel) et **le** GitLab.com géré par la société du même nom. Migrer vers GitLab.com n'aurait qu'un intérêt limité : ce serait abandonner une entreprise pour une autre, qui connaîtra peut-être le même sort si elle a du succès. Au contraire, le logiciel GitLab est libre et peut donc être installé par de nombreux acteurs. Je ne sais pas s'il existe quelque part une liste d'instances GitLab ouvertes, mais je connais Framasoft et j'ai donc choisi la leur, FramaGit <<https://framagit.org/>>. Je sais que Framasoft est lancé dans une

entreprise de « déframasofisation <<https://framablog.org/2019/09/24/deframasoftisons-internet/>> » et qu'il faudra peut-être une autre migration dans un an ou deux mais le sort de FramaGit ne semble pas clairement fixé. Si vous connaissez un GitLab (ou équivalent) chez un CHATON <<https://chatons.org/>> sympa...

Passons maintenant à la pratique. Il faut récupérer les données. L'utilisation d'un VCS décentralisé comme git résout automatiquement une grande partie du problème. Chaque copie locale contient non seulement tout le code mais également tout l'historique. Il ne reste plus qu'à récupérer les tickets et les "pull requests". Personne ne migrerait depuis GitHub si cela signifiait de perdre ces importantes informations. Pour tout récupérer, GitLab a fait une excellente documentation <<https://docs.gitlab.com/ee/user/project/import/github.html>>. J'en résume les principaux points :

- La mise en correspondance des auteurs des "commits" entre GitHub et la nouvelle forge se fait sur la base d'une connexion utilisant GitHub **ou bien** sur celle de l'adresse de courrier. Sans correspondance, "commits" et tickets seront attribués à la personne faisant l'importation.
- L'administrateur de la forge GitLab utilisée doit avoir activé l'intégration GitHub <<https://docs.gitlab.com/ee/integration/github.html>>. (C'est le cas de FramaGit.)
- Ensuite, il n'y a plus qu'à faire (vous noterez que la traduction en français de GitLab est incomplète) Nouveau projet [Caractère Unicode non montré ¹] "Import project" [Caractère Unicode non montré] [depuis] Github [Caractère Unicode non montré] [Accepter l'autorisation OAuth] "Authenticate with GitHub" [Caractère Unicode non montré] [Sélectionner le projet à importer]. Le plus long est évidemment de tout vérifier, mettre à jour ses dépôts locaux, documenter sur l'ancien dépôt (je n'ai pas détruit les anciens dépôts, juste mis à jour la documentation) et sur le nouveau.

Comme mon adresse de courrier n'était pas la même sur GitHub et sur FramaGit, je ne pouvais pas compter sur une bonne correspondance. Je me suis donc une fois connecté à FramaGit en utilisant mon compte GitHub, créant ainsi un compte « fantôme » qui allait recevoir les anciens "commits". (J'aurais pu m'en passer, ce qui aurait réaffecté ces commits au compte FramaGit habituel.) Ensuite, tout s'est bien passé, commits, "pull requests" et tickets sont bien importés. (Attention à ne pas trop en mettre, FramaGit semble avoir une limite à 50 projets.)

Notez que, politiquement, c'est évidemment une excellente idée que de migrer depuis un service centralisé comme GitHub vers plein de petits GitLab partout. Mais, techniquement, cela peut rendre la coopération difficile, puisqu'il faut un compte sur chaque instance pour participer. C'est d'autant plus absurde que git est lui-même décentralisé (et a des mécanismes pour contribuer sans compte <<https://git-scm.com/docs/git-request-pull>>.) Il faudrait donc qu'on puisse avoir un dépôt complètement décentralisé, par exemple en mettant les tickets dans git lui-même. (On me dit que ce serait possible avec le logiciel SourceHut <<https://sourcehut.org/>>, qui a déjà un service d'hébergement <<https://sr.ht/>>, mais je n'ai pas testé. Ou bien avec Fossil <<https://fossil-scm.org/>> ou encore git-bug <<https://github.com/MichaelMure/git-bug>>. Et il y a une liste de projets similaires <<https://github.com/dspinellis/git-issue/blob/master/README.md#related-work>>.) Une autre approche (qui ne me convainc pas, mais qui peut être utile pour certains services) serait de fédérer les GitLab, par exemple avec ActivityPub. Voir par exemple le ticket #4013 de GitLab <<https://gitlab.com/gitlab-org/gitlab-foss/issues/4013>>.

1. Car trop difficile à faire afficher par \LaTeX