

Notre gitlab évolue en Framagit. C'est très efficace !

Warning : cet article parle de forge logicielle qui sert à développer collaborativement du code. Il est donc un peu velu et technique, mais il fera plaisir aux plus « barbu-e-s » d'entre vous !

Préviousselaid, chez Framasoft : nous avons besoin d'une forge logicielle comme outil interne à l'asso... parce que même si nous ne développons pas (ou exceptionnellement) de logiciel libre ; les mettre en avant, les améliorer (parfois), les promouvoir et ouvrir des services au monde, ben ça demande de créer, maintenir, échanger et améliorer du code !

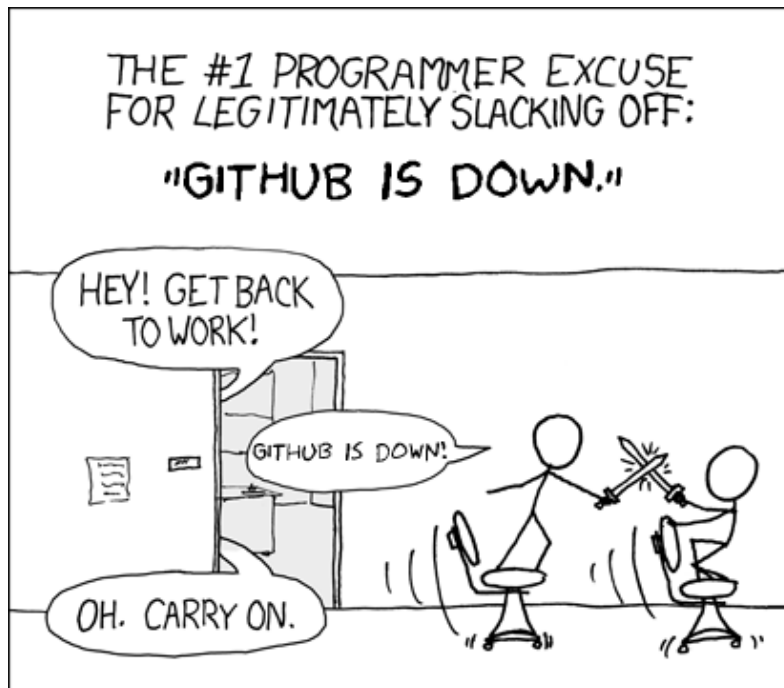
Nous nous étions donc installé [Gitlab](#) à la main, sur un coin de serveur, juste pour nous... Étant les seuls utilisateurs, on s'est dit que ce ne serait pas grave s'il n'était pas toujours à jour, à traquer la dernière version... (oui : nous sommes moins exigeants sur nos outils internes que pour les services que nous ouvrons au grand public ^^).

Franchement, merci Google !

Merci, parce qu'à chaque fois que vous prenez des décisions unilatérales aux dépens de vos utilisateurs-produits, vous nous offrez l'occasion de prouver que le Libre offre des alternatives bien plus respectueuses des personnes qui vous ont confié leur vie numérique (et leur code).

Le jour où nous avons appris que [Google Code fermait ses portes, nous avons donc décidé d'ouvrir les nôtres](#). Cela nous a aussi permis de sensibiliser au fait que, dans le mode des

codeurs et développeuses, [GitHub est devenu un point central et monopolistique assez inquiétant.](#)



L'excuse n°1 des programmeurs pour se lâcher sans scrupules :
« GitHub est en panne »
– Hé, au boulot les gars ! – Github est en panne !
– Ah bon, continuez alors.

Forcément, l'ouverture à tous de notre *git* et les nouvelles fonctionnalités des nouvelles versions de Gitlab (une nouvelle version tous les 22 du mois) nous ont incités à mettre à jour plus régulièrement, ce qui prend plusieurs heures à chaque fois... et plusieurs fois par mois, car des versions correctives sont régulièrement publiées.

Améliorer le Framagit... une priorité

Ceci, ajouté à l'utilisation grandissante de notre forge qui allait bientôt poser des problèmes de taille de disques, nous a amenés à migrer (le 17 mars dernier) notre Gitlab vers une

machine avec plus de disque et surtout avec une installation utilisant les paquets dits « omnibus ».

Ces paquets omnibus nous ont permis d'installer Gitlab à l'aide d'un simple `apt-get install gitlab-ce` plutôt que de suivre la longue procédure d'installation manuelle. Non seulement l'installation est simplifiée, mais – et c'est surtout là la plus-value que nous en attendions – mettre à jour Gitlab devient tout aussi simple avec une seule commande `apt-get dist-upgrade`.

Résultat : notre Gitlab suit scrupuleusement la publication des nouvelles versions, avec leur lot de nouvelles fonctionnalités !

*Alors, il paraît
que tu es passé
de GitHub à Gitlab ?*



Pour fêter cela, nous avons étreigné un nouveau nom de domaine.. inspiré par vous ! Avouons-le, «Git point Framasoft point orrrrrrrrghueh », ça accroche un peu en bouche. De partout, nous avons entendu parler du « [Framagit](#) » : alors tant qu'à faire, autant l'appeler comme vous le faites déjà. Bien entendu, il n'est nul besoin de modifier vos URL, elles restent valides..

mais la nouvelle est à votre disposition !

Et si on ajoutait de l'intégration continue ?

Derrière ce terme barbare se cache une fonctionnalité très pratique : on crée une « recette » qui sera exécutée dans une machine virtuelle à chaque *push*. Cela peut par exemple permettre de lancer une suite de tests pour vérifier que l'on n'a rien cassé. □

Pour utiliser cette fonctionnalité, il faut disposer de ce que l'on appelle un *runner*, c'est à dire un [logiciel](#) qui va récupérer la recette et l'exécuter. Il est possible d'installer un *runner* sur n'importe quel ordinateur, même votre ordinateur de bureau.

Pour ceux qui ne souhaitent pas gérer leur *runner* eux-mêmes, Framasoft met à disposition deux *runners* partagés entre tous les utilisateurs de Framagit, que vous pouvez utiliser comme bon vous semble. Notez toutefois que Gitlab indique que quiconque utilise un *runner* partagé peut accéder aux informations des projets utilisant ce *runner* : il vaut mieux monter votre propre *runner* pour vos projets sensibles.

De plus, en utilisant les *runners* partagés de Framasoft, il est possible que votre projet soit mis en file d'attente, en attendant que les recettes précédentes aient fini de s'exécuter... à vous de voir !



Pouhiou-le-moldu-du-code lisant cet article,
allégorie.

Vous voulez des pages Gitlab ? Nous aussi !

Github permet à tout un chacun d'héberger un site statique. Gitlab propose une fonctionnalité similaire mais hélas, uniquement dans sa version entreprise... Nous utilisons pour notre part la version communautaire qui est la version libre

de Gitlab... donc *sans* les pages Gitlab.

Nous avons donc ouvert un ticket pour demander que cette fonctionnalité soit incluse dans la version communautaire. Si vous aussi vous aimeriez voir cela arriver, aidez-nous tout simplement en votant sur <https://gitlab.com/gitlab-org/gitlab-ce/issues/14605>.

En attendant, profitez d'une forge logicielle à jour et libre sur [Framagit.org](https://framagit.org) !

Mise à jour du 5/08/2016 :

Le tutoriel d'installation de Gitlab est -enfin- disponible [sur le FramacLOUD](#).

Notez que cette installation est conjointe à celle de Mattermost (Framateam) puisque c'est ainsi que nous avons procédé ☐