

Structure Github

Le dossier envoyé est le dossier 'projet'.

Il contiendra les documents, les fichiers scripts pour configurer les différentes machines et les fichiers sources des programmes avec leur dockerfile et leur test

Nous aurons trois branches dans le répertoire git :

- branche majeur
- branche mineur
- branche commit

Chaque commit aura comme tag les numéros dans cette forme-ci : xx.xx.xx
Ce qui correspond à majeur.mineur.commit.

La branche 'commit' nous permet d'envoyer de manière régulière l'avancement du projet. Lorsque les fichiers seront arrivés, des tests unitaires et des tests d'intégrations avec Travis. Mais, même si les tests passent, il n'y aura pas la création d'image docker, ni de changements aux niveaux des machines d'hébergement. La version de l'application en production ne change pas.

La branche mineur nous permet d'envoyer le projet lorsqu'il y aura eu un avancement important. Dans ce cas, en plus des tests, l'application en production sera mis à jour.

La branche majeur sera comme la mineur mais nous ferons le commit que lorsqu'il y aura assez de changement important pour considérer que l'application est passé à une version supérieure. Par exemple, le premier commit sera pour la version 1.0.0 quand on considérera que le développement de l'application est terminé. Pour tout ce qui est de la correction des bugs et de la maintenance, les commits se feront dans 'mineur' avec des versions 1.2.0 ou bien 1.11.0.

Lors d'apparition de bugs constaté sur l'application en production, afin de les corriger, on crée une nouvelle branche 'debug' et on y met les fichiers de la branche mineur. Dès que le bug est réglé et que les tests passent, on fusionne la branche dans 'commit' et ensuite dans 'mineur' en y incrémentant le dernier nombre. On ne mélange pas 'commit' et 'mineur' car on ne souhaite pas introduire tout de suite les futurs bugs de la prochaine version 'mineur' à notre application en production. Après cela, on donne un nouveau tag pour la branche mineur.