Un turbo dans ton workflow GitHub

Speaker : Alain Hélaïli (GitHub)
Format : Conférence
Date : 18 avril 2019

Ce talk été construit autour des questions posées à Alain sur les stands GitHub de différentes conférences.
Questions récurrentes :

- On utilise Jira pour collaborer
- On a déployé Bitbucket parce qu'il est compatible avec Jira
- J'utilise GitHub, enfin … Git, quoi …

Alain souhaite nous montrer comment utiliser GitHub pour être beaucoup plus productif.
Tous les participants (sauf 1) ont un compte GitHub.
Les entreprises se mettent de plus en plus à utiliser de l'OpenSource. Même si en interne elles n'utilisent pas GitHub, pour l'OpenSource elles viennent à l'utiliser.
Exemple de la MAIF, de Décathlon, de la Société Générale ou bien encore d'AXA.
Le Gouvernement s'y est même mis avec Etatlab est très en avance.

Terme de plus en plus usité : Open Source Program Office (OSPO). Beaucoup de boites se dotent de ce département. Les grandes banques américaines s'y mettent.

Il y'a également de l'InnerSource : faire de l'OpenSource à l'intérieur de l'entreprise, sans le rendre public. Le code de Jean-Marc peut être mis à disposition de l'ensemble des employés : permet de réutiliser, mutualiser, s'entre-aider …. Permet d'avoir des contributions de développeurs qu'on ne connaît pas.

En 2019, l'automatisation ne se résume plus au TU et à la CI (un acquis depuis 10 ans).
Alain cite l'exemple amusant d'une Pull Request de Facebook dans laquelle des bots échangent des commentaires.

Le projet GitHub Learning Lab a été lancé récemment. L'idée est d'avoir un robot dans son repo GitHub pour apprendre quelque chose. Cela permet d'accueillir plus facilement de nouveaux développeurs qui se font aider par un Bot. On peut créer son propre cours / apprentissage.

2 autres robots sont déjà proposés par GitHub : Hubot et Probot

Hubot a été développé il y'a 7-8 ans. On peut dialoguer avec le bot via Slack (ou autre) en lui envoyant des commandes. Par exemple pour déployer une Pull Request dans l'environnement prod/canary : hubot va merger, attendre que la CI soit verte puis déployer.
En une seule commande : .qmtd https://github.com/github/github/pull/112545
GitHub utilise Hubot pour manager leur cluster Kubernetes. Il existe plus de 2000 packages prêts à l'emploi.
Lors d'incident de Prod, Hubot permet d'accéder aux graphiques issus de la supervision.

En résumé, Hubot permet d'interagir avec un robot depuis Slack.

Probot permet d'automatiser des interactions avec GitHub. C'est un framework basé sur GitHub Apps (Endpoint REST). Probot encapsule ces API dans Node.JS. Cela permet de démarrer rapidement une nouvelle App.

GitHub Actions : nouveau moyen plus moderne d'interagir avec GitHub.

Démo avec 2 comptes GitHub : halain987 et helaili
Repo octodemo/MySampleExpressAppOnAzure

On peut créer un board à partir d'un template (ex : Automated kanban with reviews).
Ces boards peuvent être privatisés, par exemple pour détecter toutes les issues sur lesquelles on doit travailler.
Alain nous montre la fonction de status (Busy pendant 30 minutes) pour ne pas être notifié lorsqu'on est cité dans des issues GitHub.
Dans GitHub, on peut créer des templates d'issues. Exemples : Bug report, Feature request.
Exemple d'un repo qui détecte les « it should be easy » afin d'éduquer le contributeurs.

Autre exemple dans lequel Alain crée une branche, ajoute du code avec un :
@todo parler de cette feature
@body c'est génial
Alain commit et recommande de pousser via Atom. En pratique, il signe ses commits et passe donc par la ligne de commande.
Depuis GitHub, il crée la Pull Request.
Le robot se réveille et affiche le todo et le body

Trouver la bonne personne pour effectuer une revue de code est toujours problématique. Le fichier codeowners ? permet de définir des rôles (ex : HTML géré par tel intégrateur web).
On peut désormais flagger une Pull Request comme Work in Progress pour ne pas qu'une revue se déclenche.

Autre exemple de GitHub Apps pour se simplifier la vie lors des déploiements. Alain nous présente un système utilisant l'API de déploiement de GitHub (qui ne fait pas de déploiement, mais envoie des ordres de déploiement) avec des labels GitHub pour faire des demandes de déploiement en Test, en Production … Le robot réagit au label puis envoie cet ordre de déploiement à n'importe quoi. Dans l'exemple d'Alain, ce n'importe quoi est une GitHub Actions : construction de l'image, déploiement dans Azure.
Le bouton « View Deployment » permet d'accéder directement à l'URL provisionnée par Azure.

Fonctionnement des GitHub Actions
Le fichier .github/main.workflow peut être consulté et édité depuis GitHub via une IHM. 261 actions existent déjà. Exemple : « GitHub Action for npm » exécute une commande NPM. On

peut configurer la commande et les secrets.
Une GitHub Action est facile à développer : elle se base sur des conteneurs Docker.

Lors du merge d’une branche dans develop, Alain a un bot qui supprime la branche.