

Publication de site web depuis gitlab via webhook

Henri Massias
Sandrine Layrisse

27 Mars 2018



Plan

Introduction

- Le besoin
- Le contexte
- Des avantages

Le principe

La configuration

- Côté plmlab
- Côté serveur



Introduction

Le besoin

- ▶ Déployer automatiquement les modifications sur son site web depuis gitlab

Le contexte

- ▶ Les sources du site web sont dans un dépôt git
- ▶ Un push dans le repository entraîne la mise à jour du site web

Des avantages

- ▶ Avantage d'un dépôt git (gestion de version, sauvegardé dans un endroit centralisé)
- ▶ Possibilité de faire des modifications légères directement depuis l'interface de plmlab



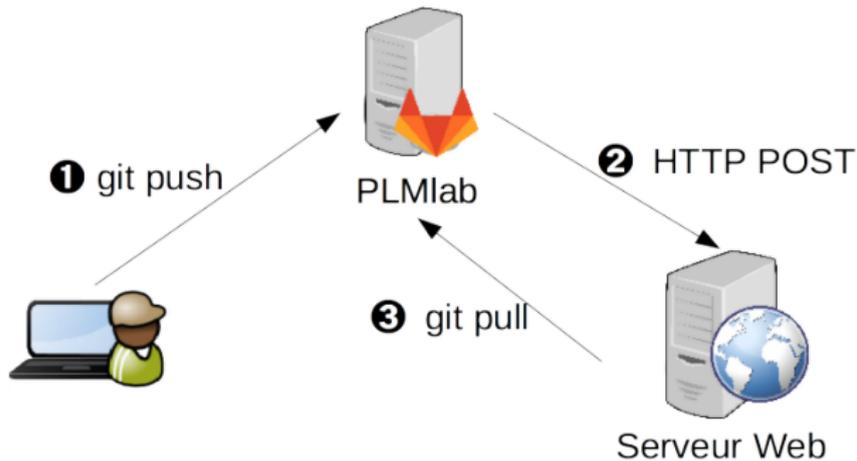
Qu'est-ce qu'un Webhook?¹

- ▶ Un Webhook est une requête POST envoyée vers une URL.
- ▶ On crée une URL `http://` ou `https://` qui accepte les données émanant du Webhook, stocke cette requête POST au format JSON ou XML, puis la traite.



¹Source : <https://www.inbenta.com/fr/blog/webhook-chatbot-api-tal/>

Le principe



- 1 Je fais un push sur le dépôt git
- 2 gitlab déclenche un webhook qui accède à l'URL `https://url-de-mon-site/maj.php`
- 3 `maj.php` lance un git pull

Configuration sur le projet PLMLab

Dans le projet PLMLab

- ▶ Aller dans le menu Settings → Integrations
- ▶ Dans le champ URL mettre `https://url-de-mon-site/maj.php`
- ▶ Sélectionner "Push events" dans "Trigger"

Webhooks (1)

`https://url-de-mon-site/maj.php`

Push Events

SSL Verification: enabled

Edit

Test ▾



Générer une clef ssh afin d'accéder au repository depuis le serveur web

Enregistrer une "Deploy Key" pour ce projet

- ▶ Aller dans le menu Settings → Repository
- ▶ Enregistrer une Deploy Key (partie publique de la clef ssh hébergée sur le serveur web)

Deploy Keys

Deploy keys allow read-only or read-write (if enabled) access to your repository. Deploy keys can be used for CI, staging or production servers. You can create a deploy key or add an existing one.

Create a new deploy key for this project

Title

Key

Paste a machine public key here. Read more about how to generate it [here](#)

Write access allowed

Allow this key to push to repository as well? (Default only allows pull access.)

Add key



Préalable sur le serveur

- ▶ Installer la bi-clef correspondant à la Deploy Key
- ▶ Faire un git clone du repository à la racine du site
`/var/www/mon-site`
- ▶ L'utilisateur sous lequel tourne le démon http doit
 - ▶ avoir les droits pour écrire à la racine du site,
 - ▶ pouvoir exécuter la commande git,
 - ▶ avoir les droits de lecture de la clef privée ssh
 - ▶ avoir les droits de lecture et écriture sur le dossier `.git`
- ▶ les connexions ssh doivent être autorisée depuis le serveur web vers plmlab

Configuration sur le serveur

- ▶ Mettre en place le script php suivant sur le serveur à l'URL `https://url-de-mon-site/maj.php`

```
<?php
echo " Update_GIT_<br>";
$lines = exec('su -& login; cd /var/www/mon-site;
GIT_SSH="/var/www/mon-site/git_ssh_command' git pull 2>&1" , $output, $res);
foreach_($output_as_$value){
echo_.$value.'<br>';
}
?>
```

- ▶ Contenu du script `git_ssh_command`

```
#!/bin/sh
exec /usr/bin/ssh -o StrictHostKeyChecking=no -i private_key_file
```



Variante

- ▶ Utiliser git en https plutôt qu'en ssh
- ▶ Créer un token http : Menu User Settings → Access Tokens → Personal Access Tokens, Scopes → api

Attention

Les Personal Access Tokens remplacent juste la saisie de votre mot de passe. Avec le Scope api , vous obtenez les mêmes droits que votre utilisateur : accès à tous vos dépôts, possibilités de faire un push, ...



Deuxième variante

- ▶ Utiliser l'intégration continue sur PLMLab afin de générer une archive du site web (artifact)
- ▶ Configurer un webhook avec le trigger "Pipeline events"
- ▶ Sur le serveur web, configurer le script maj.php qui
 - ▶ récupère l'archive générée une fois le pipeline terminé,
 - ▶ la met en place sur le serveur dans un nouveau dossier,
 - ▶ met à jour les liens de façon à ce que le site web pointe sur la nouvelle version².



² permet de revenir très facilement à une ancienne version

Autres possibilités

- ▶ Utilisation de Pages (cf. la présentation de Matthieu Boileau) : *mais ne gère pas le php*
- ▶ Configurer un gitlab runner en mode shell sur le serveur web : *mais pas très recommandé*
- ▶ Utiliser un script d'intégration continue pour faire un rsync sur le serveur : *mais nécessite de stocker une clef ssh privée sur PLMLab (utiliser une variable secrète³)*



³Menu Settings → CI/CD

Remerciements

La **PLMTeam** et en particulier Sylvain Allemand, David Delavennat et Philippe Depouilly

