

Guide d'installation de GitLab



GitLab est développé pour les OS UNIX (sauf OS X, Mac OS, Arch Linux, Fedora, Gentoo et FreeBSD), il n'est donc pas utilisable sur un OS Windows (aucun Windows) et il n'y a aucune envie de la part de l'équipe de développement de développer une solution Windows dans un futur proche. Si toutefois vous souhaitez rester sur votre OS Windows il faudra passer par une machine virtuelle.

Le matériel requis pour utiliser GitLab dépend de l'utilisation que vous comptez en faire :

1) Stockage

L'espace nécessaire sur votre disque dur dépend de la taille des dépôts GIT que vous souhaitez stocker sur GitLab. Logiquement vous devrez avoir autant d'espace libre que la somme des dépôts GIT que vous souhaitez y mettre. Si vous avez un disque SSD vous augmenterez les performances de GitLab.

2) CPU

1 core : supporte jusqu'à 100 utilisateurs mais la plateforme GitLab sera ralentie dû au fait que les workers et les jobs seront tous lancés sur un seul et même core.

2 cores : largement recommandé et suffisant selon GitLab, cela vous permet d'avoir jusqu'à 500 utilisateurs sur la plateforme.

3) Mémoire

Il vous faudra au minimum 4GB de mémoire (RAM + SWAP) pour installer et utiliser GitLab.

Installation sur Ubuntu 14.04

Installation et configuration des dépendances nécessaire

Si vous avez installé « Postfix » afin d'envoyer des e-mail veuillez à sélectionner « Internet Site » pendant l'installation. À la place de « Postfix » vous pouvez également utiliser « Sendmail ».

Sur Centos 6 et 7 les commandes ci-dessous vont ouvrir des accès HTTP et SSH sur le pare-feu du système.

```
sudo apt-get install curl openssh-server ca-certificates postfix
```

Ajout du paquet serveur GitLab et installation du paquet

Nous allons à présent télécharger le paquet serveur GitLab et l'installer grâce aux commandes ci-dessous.

```
curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash  
sudo apt-get install gitlab-ce
```

Installation sur Raspberry PI 2 sous Raspbian

Installation et configuration des dépendances nécessaire

Si vous avez installé « Postfix » afin d'envoyer des e-mail veuillez à sélectionner « Internet Site » pendant l'installation. À la place de « Postfix » vous pouvez également utiliser « Sendmail ».

```
sudo apt-get install curl openssh-server ca-certificates postfix apt-transport-https  
curl https://packages.gitlab.com/gpg.key | sudo apt-key add -
```

Ajout du paquet serveur GitLab et installation du paquet

Nous allons à présent télécharger le paquet serveur GitLab et l'installer grâce aux commandes ci-dessous.

```
sudo curl -sS https://packages.gitlab.com/install/repositories/gitlab/raspberry-pi2/script.deb.sh | sudo bash  
sudo apt-get install gitlab-ce
```

Configuration et démarrage de GitLab

Maintenant que GitLab est installé sur la machine nous allons pouvoir configurer les premiers paramètres qui permettront de démarrer le service GitLab.

Pour ce faire, lancez la commande suivante :

```
sudo gitlab-ctl reconfigure
```

Rendez vous sur la page de connexion et connectez vous

Lors de votre première visite vous serez redirigé vers une page vous permettant de créer le mot de passe de votre compte administrateur. Entrez le mot de passe que vous souhaitez et vous serez redirigé vers la page de connexion.

Le compte par défaut est « root ». Entrez le mot de passe que vous avez saisi précédemment et connectez vous. Après cette connexion vous pourrez changer le nom du compte à votre souhait.

Les petits plus

Si vous souhaitez aller encore plus loin GitLab vous permet d'utiliser un système d'intégration continue puisque cette option est intégrée dans GitLab. GitLab CI fait parti de GitLab, l'application utilise une API qui permet de communiquer avec GitLab.

Après chaque push l'application va exécuter les tests de votre projet et vous indiquer quels sont les erreurs qui ont été rencontrées grâce à un tableau de bord contenant tous les projets que vous avez décidé d'utiliser sur GitLab CI.

Si vous avez envie de vous faire un tableau de bord de votre activité sur le GitLab vous pouvez également le faire, l'application GitLab regorge d'API et est libre, cela vous permet de développer des outils comme un tableau de bord retraçant les tickets clos dernièrement etc. Un outil comme celui-ci a été développé dans mon entreprise en utilisant Symfony2 et nous permet ainsi d'être très souple sur le planning en sachant combien d'issues était attribué à chacun et en indiquant un niveau de complexité sur chaque issue nous sommes capable de donner du travail pour la semaine, de suivre l'avancement en milieu de journée, de venir en aide quand nous voyons qu'un collaborateur est coincé sur une issue.

La plupart des IDE (NetBeans, JetBrains, Eclipse ...) possèdent des plugins permettant l'utilisation des fonctionnalités de Git plus simplement comme par exemple revenir à une version précédente d'un fichier, « pusher » du code, voir l'historique d'un fichier, etc. directement dans l'IDE.

