

CreerSonPropreGithub

1-explication de git et github

2-différence entre git et github

3-creation de notre github

1-explication de git et github

git

git est ce qu'on appelle (*VLC*) ou *version control system* ou encore système de contrôle de version en français.

C'est un outil qui va nous permettre versionner notre code. Il est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux.

Github

par contre ce qu'on appelle remote repository (comme: gitlab, bitbucket, aws Codecommit, google cloud source repository). Il est précisément un logiciel de contrôle de version car il utilise git en interne.

Avant github, l'utilisation de git se faisait en local, on pouvait d'ores et déjà créer, versionner notre code, revenir à la version précédente...(bref y appliquer toutes les commandes possibles) sauf que le travail fait restait en local.

Puis est survenue le problème de la disponibilité (car le code écrit n'était accessible qu'uniquement sur notre machine). Avec l'avènement des solutions cloud, GITHUB est donc née, ce logiciel va nous permettre de faire tout ce que git tout seul nous permettait de faire avant. Mais sauf qu'il vient résoudre le problème de disponibilité.

Grâce à github on peut donc aussi versionner notre code, et de plus il peut être accessible de toutes les machines connectées à internet. C'est donc pour cela qu'on dit que github utilise github.

2-différence entre git et github

Pour résumer la **différence entre** les deux :

Git est un logiciel local qui permet aux développeurs de sauvegarder instantanément leurs projets au fil

du temps. **GitHub** est une plateforme web qui intègre les fonctionnalités de contrôle de version de **Git** afin de pouvoir les utiliser en collaboration.

3-creation de notre github

Ici nous n'allons pas créer un logiciel complet comme github, ce que nous allons faire c'est de créer un depot distant, pour cela nous allons utiliser un conteneur docker accessible en ssh.

Ne soyez pas frustré par l'utilisation des mots, ces notions en elles-mêmes sont faciles à prendre en main.

Prerequis : -installer docker, ssh et git sur sa machine.

Etape1 : écrire un Dockerfile pour créer une image accessible en ssh

pour cela vous pouvez partir de l'image robertolandry/hello_ssh, elle a pour utilisateur hello_user.

Le principe c'est de générer une paire de clef en local et copier le contenu de celle avec l'extension .pub dans le fichier authorized_keys situé dans le home directory de l'utilisateur en question (hello_user pour notre cas). Ici le chemin sera donc home/hello_user/.ssh/authorized_keys.

Exemple : `COPY cle.pub /home/hello_user/.ssh/authorized_keys`

Ensuite,

etape2:installer git dans le dockerfile

cela se fait en deux commandes :

`RUN sudo apt update`

`RUN sudo apt install git.`

Enfin,

etape3 : exposer le port 22, et créer un depot local grace au git init, démarrer le service ssh

les commandes sont respectivement les suivantes :

`RUN git init --bare /home/hello_user/test/`

`RUN sudo chmod 666 -R /home/hello_user/test`

`EXPOSE 22`

`CMD ["/usr/sbin/sshd", "-D"]`

etape3 : ensuite créer l'image à partir du dockerfile

ceci se fait à partir de la commande : `docker build -t nomimage CheminVersLeDockerfile`

Exemple : `docker build -t imagegithub ./Dockerfile`

etape4 : créer le conteneur à partir de l'image

ceci a partir de la commande : `docker run -d --name nomconteneur nomimage`

Exple : `docker run -d conteneurdepotgithub1 imagegithub`

etape5 : l'heure de tester la connectivite en ssh

-il faut d'abord avoir l'adresse ip du conteneurdegithub1

pour se faire la commande est assez simple : `docker inspect conteneurdegithub1`

plus bas il y'aura un champ adresse ip, copier l'adresse en question

-il faut maintenant tester la connection en ssh du conteneurdegithub1

pour se faire la commande est : `ssh -i nomcleprivegenereploshaut
nonutilisateurduconteneur@adresseip`

Exple : `ssh -i key hello_user@127.17.0.2`

si c'est bon vous aurez surement un truc comme ceci

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

* Documentation: <https://help.ubuntu.com>

* Management: <https://landscape.canonical.com>

* Support: <https://ubuntu.com/advantage>

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

hello_user@b6c50ea9b223:~\$

-si votre resultat est le meme vous avez presque finit, on vas maintenant créer un depot local et envoyer nos travaux dans le depot distant qui est notre conteneurdegithub1.

1- créer un dossier dans votre machine et y entrer

2- faire un git init

3- produire un travail : ca peut etre nimporte qui, créer meme un fichier a.txt dans le meme dossier

4- faire un git add : git add .

5- faire un commit : git commit -m "version1"

-maintenant vient la partie la plus crutiale : la liaison des deux depots

ceci est aussi simple , elle se fait grace a la commade :

git remote add origin ssh://hello_user@172.17.0.2/home/hello_user/test

-maintenant on peut faire un push :

NB : il est a noter qu'avant de faire le push, comme sur github et vu qu'on fait un push en ssh, il faut copier le contenu de la cle publique id_rsa.pub situe dans le dossier **.ssh** de votre home directory et le coller a la suite dans le fichier **authorized_keys** situe dans le dossier **.ssh** du conteneur.

Si toute fois sur votre machine , il n'y a pas cette cle il faut soit il faut faire un **ssh-keygen** sur le terminal de votre machine et taper a chaque fois sur la touche entrer.

Ou bien copier la cle prive de la paire de cle genere plus pour l'accès en ssh au conteneur, et la coller dans le dossier **.ssh** du home directory de votre machine locale.

:)

ci c'est fait , on peut maintenant faire un push. Ceci grace a la commande :

git push -u origin master

Merci pour votre temps et votre attention !!!

pour ceux qui n'ont pas pu arriver a ce niveau, j'ai mis dans mon repository github un script qui vous permettra de faire cela automatiquement.

PREREQUIS POUR UTILISER LE SCRIPT :

- installer github sur votre machine locale**
- installer ssh sur votre machine locale**
- installer docker sur votre machine locale**