

TP GIT

1)

Dans les 3 premières parties je m'étais trompé et avait nommé le dossier avec notre prénom « prénom » mais l'ai renommé par la suite.

On se connecte avec un compte utilisateur.

Avec les commandes linux :

On se rend dans le dossier www à l'aide de `cd : /var/www/`

Création du dossier prénom avec `sudo mkdir prénom-corentin`

On se rend dedans.

Installation de git avec `sudo apt install git`.

Maintenant, on utilise les commandes git

Clonage : `sudo git clone https://github.com/Uneip/CS2I-B2023.git`

```
@debian:/var/www/prenom$ sudo git clone https://github.com/Uneip/CS2I-B2023.git
Clonage dans 'CS2I-B2023'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (16/16), done.
Réception d'objets: 100% (30/30), 4.05 Kio | 4.05 Mio/s, fait.
remote: Total 30 (delta 0), reused 24 (delta 0), pack-reused 0
@debian:/var/www/prenom$
```

On se rend dans la bonne branche :

```
@debian:/var/www/prenom/CS2I-B2023$ sudo git checkout Corentin_Achard
La branche 'Corentin_Achard' est paramétrée pour suivre la branche distante 'Corentin_Achard' de
'origin'.
Passer sur la nouvelle branche 'Corentin_Achard'
@debian:/var/www/prenom/CS2I-B2023$ ls
dossier_achard  README.md  toto.txt
@debian:/var/www/prenom/CS2I-B2023$ cd dossier_achard
@debian:/var/www/prenom/CS2I-B2023/dossier_achard$
```

2)

On récupère directement le wordpress utilisé lors d'un précédent tp.

On le copie-colle à l'aide de la commande linux adaptée

On change de répertoire pour se rendre dans le dossier peugeot.fr où se trouve le wordpress.

```
cp -r /var/www/peugeot.fr/wordpress/ /var/www/prenom/
```

```
cp -r /var/www/peugeot.fr/wordpress/ /var/www/corentin/CS2I-B2023/dossier_achard
```

Désormais si on se rend dans corentin et qu'on fait un ls, on retrouve le wordpress.

TP GIT

Il faut désormais commit avec les commandes git :

Sudo git add wordpress

Puis on s'identifie

```
b@debian:/var/www/prenom$ sudo git config --global user.email cocolanoix@gmail.com
b@debian:/var/www/prenom$ sudo git config --global user.name "corentin"
```

Sudo git commit wordpress

On rajoute un commentaire en bas du vi qui vient de s'ouvrir et on quitte.

3)

On configure le wordpress :

Lamp est déjà installé

J'ai dû réinstaller apache2 et le reconfigurer :

Avec vi de linux :

```
<VirtualHost *:80>
    ServerAdmin gittip@localhost
    DocumentRoot /var/www/corentin/CS2I-B2013/dossier_achard

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Dans /etc/apache2/

Vi sites_available/000-default.conf

On doit configurer une base de données.

Connexion sur mariadb :

Mysql -u root -p

On s'identifie

Ensuite on passe en sql :

CREATE DATABASE gittip ;

GRANT ALL ON gittip.* TO 'gittip'@'localhost' IDENTIFIED BY 'gittip' ;

FLUSH PRIVILEGES ;

Pour partir, EXIT ;

Ensuite, on refait une commande linux, cette fois pour attribuer les droits :

TP GIT

Chown -R www-data :www-data /var/www/corentin/wordpress

On fait un `sudo systemctl restart apache2` (commande linux)

Ne fonctionne pas mais je suis très en retard donc je passe à la suite.

(le serveur fonctionne mais je ne peux accéder qu'à la page de base de debian)

On va maintenant commit le wordpress sur github

On doit exclure wp-config.php

Pour cela on utilise .gitignore.

Je l'avais précédemment stupidement retiré.

Désormais :

`Sudo git add wordpress`

`Sudo git commit wordpress`

`Sudo git push origin Corentin_Achard`

On nous demande des identifiants.

Il faut rentrer notre nom d'utilisateur GITHUB ainsi qu'un token préalablement généré dans nos paramètres.

Le push est réalisé.

4)

On restreint l'accès au .htaccess :

On se rend dans `/etc/apache2/site-available` et on ouvre `000-default.conf` avec vi

Les modifications suivantes sont réalisées dans VirtualHost :

```
ServerAdmin git@localhost
DocumentRoot /var/www/corentin/CS2I-B2013/dossier_achard
    <Directory /var/www/corentin/CS2I-B2013/dossier_achard
        allowoverride all
    </Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
```

On vient maintenant compléter le fichier .htaccess du wordpress avec les lignes suivantes :

TP GIT

```
AuthType Basic
AuthName "Accès protégé"
AuthUserFile /usr/home/login/www/.htpasswd
require valid-user_
```

On crée désormais .htpasswd :

```
b@debian:/etc/apache2$ sudo htpasswd
[sudo] Mot de passe de b :
Usage:
    htpasswd [-cimBdpsDv] [-C cost] passwordfile username
    htpasswd -b[cmBdpsDv] [-C cost] passwordfile username password

    htpasswd -n[imBdps] [-C cost] username
    htpasswd -nb[cmBdpsDv] [-C cost] username password
-c Create a new file.
-n Don't update file; display results on stdout.
-b Use the password from the command line rather than prompting for it.
-i Read password from stdin without verification (for script usage).
-m Force MD5 encryption of the password (default).
-B Force bcrypt encryption of the password (very secure).
-C Set the computing time used for the bcrypt algorithm
  (higher is more secure but slower, default: 5, valid: 4 to 17).
-d Force CRYPT encryption of the password (8 chars max, insecure).
-s Force SHA encryption of the password (insecure).
-p Do not encrypt the password (plaintext, insecure).
-D Delete the specified user.
-v Verify password for the specified user.
On other systems than Windows and NetWare the '-p' flag will probably not work.
The SHA algorithm does not use a salt and is less secure than the MD5 algorithm.
b@debian:/etc/apache2$ _
```

5)

On crée maintenant un fichier avec l'historique des versions :

Pour cela, on va utiliser la commande suivante :

Sudo git log > logs.txt

Si l'on fait un vi, on obtient :

```
commit b5cc65bde92f425d780cc315c79ea1803fc9ab17
Author: corentin <cocolanoix@gmail.com>
Date: Tue Feb 14 16:44:41 2023 +0100

    Commentaire wordpress

commit f66bb059b2ddda8688f19e3d7721f0e8711c65dc
Author: ACHARD Corentin <cocolanoix1234@gmail.com>
Date: Tue Feb 14 12:46:18 2023 +0100

    Bonjour, admirez mon magnifique travail

commit 7510cf9bd5da8f74099d47a205ac50b8b50ec5af
Author: Uneip <55940964+Uneip@users.noreply.github.com>
Date: Tue Feb 14 11:03:10 2023 +0100

    Create toto.txt

commit dc1bb7f45362eaf61419a282c660f84f09a06804
Author: Uneip <55940964+Uneip@users.noreply.github.com>
Date: Tue Feb 14 10:50:48 2023 +0100

    Initial commit
```

6) Nous allons maintenant restaurer la version avant le htaccess.

Il faut utiliser la commande git :

Sudo git restore htaccess

7)

On fait un push vers github avec toutes les modifications apportées.

Cela fait, on fait une commande git pour retourner dans le main :

Git checkout main

On peut maintenant faire un

Git merge Achard_Corentin

```
root@debian:/var/www/corentin/CS2I-B2023# ls
dossier_achard Entwanne Evan FournierCharliTpGIT.pdf README.md Romain_Bernard toto.txt
root@debian:/var/www/corentin/CS2I-B2023#
root@debian:/var/www/corentin/CS2I-B2023# _
```

Cela fait, on fait une commande git pour retourner dans le main :

On peut désormais push de nouveau.

Cela ne fonctionne pas donc je fais un pull comme conseillé.

J'ai maintenant une erreur plus problématique : la vm est pleine.

TP GIT

```
root@debian:/var/www/corentin/CS2I-B2023#  
root@debian:/var/www/corentin/CS2I-B2023# git pull  
remote: Enumerating objects: 51, done.  
remote: Counting objects: 100% (15/15), done.  
remote: Compressing objects: 100% (6/6), done.  
remote: Total 51 (delta 8), reused 14 (delta 8), pack-reused 36  
error: erreur d'écriture d'un fichier: Aucun espace disponible sur le périphérique  
fatal: impossible d'écrire le fichier d'objet esseulé  
fatal: échec de unpack-objects  
root@debian:/var/www/corentin/CS2I-B2023# _  
On peut désormais push de nouveau.
```

J'ai essayé de passer mon répertoire dans home qui n'était pas plein mais toujours impossible de pull.