

## Utilisation de Git et Github

Nous voilà à une étape clé du module.

Afin d'assurer une sauvegarde de votre travail, nous allons apprendre à utiliser très rapidement Git avec Github.

### À savoir que :

Git est un outil de gestion de version qui permet de stocker un ensemble de fichiers de manière à suivre chronologiquement les modifications apportées à un projet.

GitHub est quant à elle une plateforme permettant d'héberger les répertoires Git.

L'avantage avec le Nuage Pédagogique, c'est que nous allons pouvoir le configurer une bonne fois pour toutes.

### 1. Création de la connexion

Dirigez-vous sur l'éditeur de code (VS Code) de votre serveur Nuage-Pédagogique et procédez aux étapes suivantes :

- Placez-vous dans le répertoire racine de votre serveur web grâce à la commande « `cd /var/www/html` »
- Générez une nouvelle clé d'authentification à l'aide de « `ssh-keygen -t ed25519 -C "your_email@example.com" -f /var/www/html/.ssh/npgithub` ». Cette clé doit rester secrète et ne doit jamais être divulguée. Elle permettra à GitHub d'autoriser les échanges avec votre espace Nuage-Pédagogique.

-t est utilisé pour indiquer l'algorithme de chiffrement

-f est utilisé pour indiquer le chemin et le nom de la clé

L'exécution de cette commande vous demandera d'entrer une « passphrase ». Elle n'est pas obligatoire, mais fortement recommandée. Si vous souhaitez la renseigner, ne perdez pas cette phrase. Elle ajoute une surcouches de sécurité en chiffrant votre clé.

- Pour que la clé soit fonctionnelle, vous devez l'ajouter à l'agent d'authentification. Vous pouvez y parvenir en lançant la commande « `ssh-add /var/www/html/.ssh/npgithub` ».
- En faisant la commande « `ls -l /var/www/html/.ssh` » vous pourrez remarquer que deux fichiers ont été créés : « `npgithub` » et « `npgithub.pub` ». Le premier fichier est

votre clé privée et ne doit jamais être divulgué. Le deuxième fichier est votre clé publique, c'est celle que vous pouvez partager pour autoriser des connexions.

Votre jeu de clé est maintenant prêt à être utilisé. Nous allons autoriser l'échange entre GitHub et notre instance du Nuage-Pédagogique.

Si vous n'avez pas de compte GitHub, je vous invite à bien vouloir en créer un. Vous pouvez d'ailleurs utiliser votre adresse mail EPSI. Vous pourrez ainsi réutiliser ce compte Github pour tous vos projets de formation.

Afin d'autoriser la connexion, nous allons devoir ajouter notre clé publique dans les paramètres de notre compte Github. Pour cela vous devez accéder à la zone de paramétrage. Celle-ci est accessible depuis le menu déroulant en haut à droite, notamment représenté par votre photo de profil. Dans ce menu vous choisirez le lien avec le libellé « Settings ».

À partir de cette nouvelle page, choisissez dans le menu vertical de gauche le lien « SSH and GPG keys ». Vous devriez voir apparaître une section nommée « SSH keys » avec notamment un bouton « New SSH key » qui vous permet d'ajouter la clé publique. Cliquez dessus et remplissez les champs.

Dans le champ « Title » vous pouvez mettre l'intitulé que vous souhaitez. L'idéal c'est de bien nommer vos connexions afin de mieux repérer les services qui accèdent à votre compte. Pour le « Key type », l'option « Authentication Key » doit être sélectionnée.

Enfin dans la zone « Key » vous devez insérer votre clé publique contenue dans le fichier « npgithub.pub ». Pour afficher la clé de ce fichier, retournez dans la console de votre IDE dans le Nuage Pédagogique et exécutez la commande « cat /var/www/html/.ssh/npgithub.pub ».

Copiez le texte, collez-le dans la zone « Key » et validez l'ajout.

Vous pouvez tester la connexion avec le site Github en lançant la commande « ssh -T git@github.com ». Si vous obtenez un message de réussite, vous pouvez à la suite. Dans le cas contraire, ressayer la procédure.

## 2. Configuration du répertoire de travail

Votre connexion avec Github est maintenant fonctionnelle. Nous allons voir comment configurer notre premier répertoire de travail.

Pour ce projet de site e-commerce, vous aller vous voir assigner un répertoire spécifique grâce à GitHub Classroom. L'idée étant de faciliter le suivi de votre code source par l'intervenant.



Dans un projet classique, il vous suffirait juste de créer votre propre répertoire.

Rendez-vous donc sur le lien : <https://classroom.github.com/a/AEtEZ-9T>

Une page va s'ouvrir et vous demander d'autoriser l'accès à votre compte GitHub, acceptez.

Vous allez normalement arriver sur une page contenant le message :

*« You accepted the assignment, php-shop . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates. »*

Veuillez patienter quelques instants et rafraîchissez votre page.

Lorsque la création est terminée et après le rafraîchissement de votre page, un lien devrait apparaître. Ce lien permet d'accéder au répertoire qui vous a été assigné, cliquez donc sur ce lien.

Nous allons maintenant lier le répertoire GitHub avec notre répertoire « shop ». Pour cela vous allez devoir suivre la procédure ci-après exposée.

Avant de commencer la procédure, vous allez créer dans la racine du dossier du projet « shop » un nouveau fichier « .gitignore ». Ce fichier va permettre d'ignorer des fichiers et dossiers. Dans notre projet, nous devons simplement empêcher l'envoi du dossier « vendor ». En effet, il possède beaucoup trop de fichiers. Grâce à Composer, il nous sera très facile de réinstaller ces librairies.

Dans le fichier « .gitignore » ajouter le texte suivant :

`/vendor/`

À partir de la console de votre éditeur de code, accédez au dossier du site e-commerce. Il devrait normalement s'appeler « shop ».

Pour rappel la commande permettant de se placer dans un dossier est la suivante : « cd shop »

Maintenant, veuillez exécuter ces quelques commandes :

- Initialiser le répertoire avec « git init »
- Ajouter des fichiers pour créer une version avec « git add . ». Le point signifie que nous allons ajouter l'ensemble des fichiers de notre projet mis à part les fichiers et dossiers qui doivent être ignorés.
- Créer un commit (une version du code) grâce à « git commit -m « Premier commit » ». Le « -m » permet de définir le libellé de la version. Il est fort recommandé de le renseigner.
- Créer la branche principale à l'aide de « git branch -M main »
- Connecter le répertoire GitHub avec la commande « git remote add origin <repository\_link> ». Vous pouvez récupérer le lien du répertoire depuis la page GitHub. Attention, choisissez bien le lien « SSH ». Le lien devrait ressembler à cela : « git@github.com:epsi-arras/php-shop-votreNomGit.git ».

- Envoyer le code sur le répertoire GitHub en exécutant « `git push -u origin main` »

Voilà, vous venez d'envoyer votre première version de votre site e-commerce. Pour vérifier que tout a bien fonctionné, vous pouvez retourner sur votre répertoire. Vous devriez voir apparaître les dossiers et fichiers de votre site. Vous verrez également le message que vous avez défini pour le commit.

### 3. Procédure de création de versions

Nous venons de voir comment initialiser un répertoire GitHub, mais sachez que toutes les opérations ne seront plus nécessaires par la suite.

Voici donc la procédure que vous pourrez utiliser pour la suite du projet.

#### Ajouter des fichiers pour la future version

```
git add <fichier|dossier> // ajouter un fichier ou un dossier spécifique  
git add *.<extension> // ajouter tous les fichiers avec une extension en particulier  
git add . // ajouter tous les fichiers
```

#### Créer une version ou un commit

```
git commit -m « Votre message de version »
```

#### Envoyer une version sur le répertoire distant

```
git push --set-upstream origin main // Main correspond à la branche dans laquelle envoyer  
les sources
```

À partir de maintenant vous êtes capable de créer des versions de votre site e-commerce.