

## 3. Fondamentaux de Git

### 3.1. Clonage d'un projet : SSH vs. HTTPS

Lorsque vous clonez un dépôt Git, vous pouvez utiliser deux protocoles principaux : SSH (Secure Shell) et HTTPS (Hypertext Transfer Protocol Secure). Le choix entre SSH et HTTPS influence la manière dont vous vous authentifiez auprès de GitHub pour cloner et synchroniser les dépôts.

**HTTPS** : C'est le moyen le plus simple pour les débutants de cloner un dépôt. Vous n'avez pas besoin de configurer de clés SSH, mais vous devrez entrer votre nom d'utilisateur et mot de passe GitHub (ou un token d'accès personnel pour une sécurité accrue) à chaque fois que vous poussez ou tirez des modifications.

**SSH** : Utilise une paire de clés cryptographiques pour s'authentifier. Une fois configuré, vous n'avez pas besoin de saisir votre nom d'utilisateur ou mot de passe pour les interactions avec le dépôt. SSH est plus sécurisé et pratique pour les utilisateurs avancés.

### 3.2. Publication de votre projet sur Git

Publier votre projet sur Git signifie le pousser vers un dépôt distant sur une plateforme comme GitHub. Pour cela, suivez ces étapes :

1. Créez un nouveau dépôt sur votre plateforme de gestion de version (ex. GitHub).
2. Ouvrez Git Bash et naviguez vers le dossier de votre projet local.
3. Initialisez le dossier comme un dépôt Git en utilisant : `git init`.
4. Ajoutez tous les fichiers à votre dépôt local en utilisant : `git add .`
5. Faites votre premier commit : `git commit -m "Initial commit"`.
6. Liez votre dépôt local au dépôt distant en utilisant : `git remote add origin URL_DEPOT_DISTANT`.
7. Poussez vos modifications vers le dépôt distant : `git push -u origin master`.

### 3.3. Le rôle du fichier .gitignore

Le fichier `.gitignore` indique à Git quels fichiers ou dossiers ignorer dans un projet. Typiquement, vous voudriez exclure des fichiers contenant des informations sensibles, des fichiers de configuration personnels, ou des dépendances installées localement. Pour utiliser `.gitignore`, créez simplement un fichier à la racine de votre projet et listez les chemins à ignorer.

### 3.4. Création et clonage d'un dépôt via SSH

Pour créer et cloner un dépôt via SSH, assurez-vous d'avoir configuré vos clés SSH avec GitHub. Ensuite :

1. Sur GitHub, créez un nouveau dépôt et sélectionnez "SSH" pour l'URL du dépôt.
2. Copiez l'URL SSH du dépôt.
3. Ouvrez Git Bash et exécutez : `git clone URL_SSH`.
4. Vous avez maintenant une copie locale du dépôt, prête à être utilisée et modifiée.