6. Configurer GIT

Cette étape est utile pour les développeurs ou pour les administrateurs qui souhaite utiliser des projets provenant de repository Git (Github, Gitlab, etc..).
Pour les autres utilisateurs il n'est pas nécessaire de réaliser cette étape.

- Mise en place de GIT
- Les commandes GIT
- Migration d'un Repository Git vers Gitea
- Mise en place clé SSH Git

Mise en place de GIT

Installation de GIT

```
sudo apt install git
```

Configurer GIT

Pour configurer Git il suffit de paramétrer un utilisateur avec les 2 lignes de commandes suivantes.

```
git config --global user.name "Sammy"
git config --global user.email "sammy@domain.com"
```

Les informations que vous saisissez sont stockées dans votre fichier de configuration Git, que vous pouvez éventuellement modifier à la main avec un éditeur de texte comme celui-ci :

```
nano ~/.gitconfig
```

Lister les configurations

Pour afficher toutes les configurations git existantes sur le serveur :

```
git config --list
```

Les commandes GIT

Liste des commandes GIT utiles

Commande sur un répertoire de travail

Lister tous les commit existant :

```
git log --oneline
```

Pour avoir plus d'information :

```
git log
```

Pour voir l’état du répertoire, entre autre termes les fichiers qui sont supprimé, modifié ou a ajouter :

```
git status
```

Migration d'un Repository Git vers Gitea

Nous allons partir du principe que nous avons un repository Git existant avec des branches, commits etc..

Notre repository est "DemoRepoGit" se trouvant sur un serveur Linux via le chemin d'accès suivant `/path/to/git/repo/DemoRepoGit.git`.

> Avant de commencer la migration il est nécessaire de créer le repository **DemoRepoGit** vide sur Gitea.

Préparation de l'environnement de migration

Pour ce faire nous allons d'abord créer un clone du repo afin d'éviter de perdre les données en cas de mauvaise manipulation lors de la migration.

Se placer dans un dossier temporaire, par exemple `/tmp`.

Cloner le repo dans l'environnement de migration

`--mirror` = branches + tags + refs → migration complète

```
# Cloner en mirror (IMPORTANT)
git clone --mirror /path/to/git/repo/DemoRepoGit.git
```

Puis se placer dans le repo cloné (`cd DemoRepoGit.git`).

> Il est possible que la commande retourne l'erreur `fatal : propriétaire douteux détecté dans le dépôt...`. Cette erreur corresponds à un mécanisme de sécurité récent de Git.
> Maintenant Git bloque l'accès au repo si on est "root" et qu'il appartient à un autre utilisateur.
> Pour corriger exécuter la commande suivante : `git config --global --add safe.directory /path/to/git/repo/DemoRepoGit.git`.

Ajouter le nouveau remote

L'ajout du nouveau remote permet de configurer la destination lors de l'envoi des commits, tags...

```
# Ajouter le remote Gitea
git remote add gitea http://gitea:3000/USER/DemoRepoGit.git
```

Envoi du nouveau repo

```
# Push complet
git push --mirror gitea
```

Attention si il y avait des Hooks Git il faudra les reconfigurer.

Mise en place clé SSH Git

Prérequis

Si Gestionnaire de repository Self hosted (Gitea) :

- Activer le port sur le service Gitea (2222->22)
- Ouvrir le par-feu serveur et du routeur (Box internet)
- Activer le service SSH dans le app.ini du service

Générer la clé de sécurité

Sur la machine local exécuter la commande suivante pour générer la clé SSH :

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Il n'est pas nécessaire de mettre une passphrase.

Ajouter la clé au compte

Pour ajouter la nouvelle clé au compte Gitea il faut récupérer le contenu de la clé publique :

```
cat ~/.ssh/id_ed25519.pub
```

Le contenu de la clé ressemble à ceci :

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHpRFB++5649q68s47audlJzpfdVUfAL3qJhxvaKkKSDHCsq
your_email@example.com
```

Puis sélectionner le contenu dans le presse papier.

Ouvrir les paramètres du compte puis se rendre dans l'onglet "Clés SSH / GPG".

Ajouter une nouvelle clé, donner un nom à la clé, puis coller le contenu du presse papier.

Tester la connexion avec la nouvelle clé

Pour tester la connexion il faut exécuter la commande suivante :

```
ssh -T -p <port> -i ~/.ssh/id_ed25519 git@domain.name
```

Si la connexion est time-out essayer de se connecter en local depuis le serveur hébergeant le service : `ssh -T -p <port> git@localhost` Si la connexion réussi c'est que le problème provient d'un par-feu, sinon c'est un problème de configuration du service.

Cloner le repository

Une fois la clé ajouté au compte Git il va être possible de cloner le repository via l'url SSH :

```
# Cloner un repo
git clone ssh://git@<domaine.name>:<port>/<user>/DemoRepoGit.git
# Par exemple : git clone ssh://git@git.galaxy.com:2222/john/monprojet.git

# Ajouter un nouvel origin si repo déjà existant
git remote add origin ssh://git@<domaine.name>:<port>/<user>/DemoRepoGit.git
```

Il est toujours possible de modifier un remote origin si il y a un problème.
Lister les origines `git remote -v` puis pour modifier un origin :

```
git remote set-url origin git@<domain.name>:<port>/<user>/<repo>.git
```