Git premiers pas

Git - Découverte et initialisation

Ressources documentaires

- Wikipédia
- Site officiel
- OpenClassroom
- Git petit guide

Utilisation générale

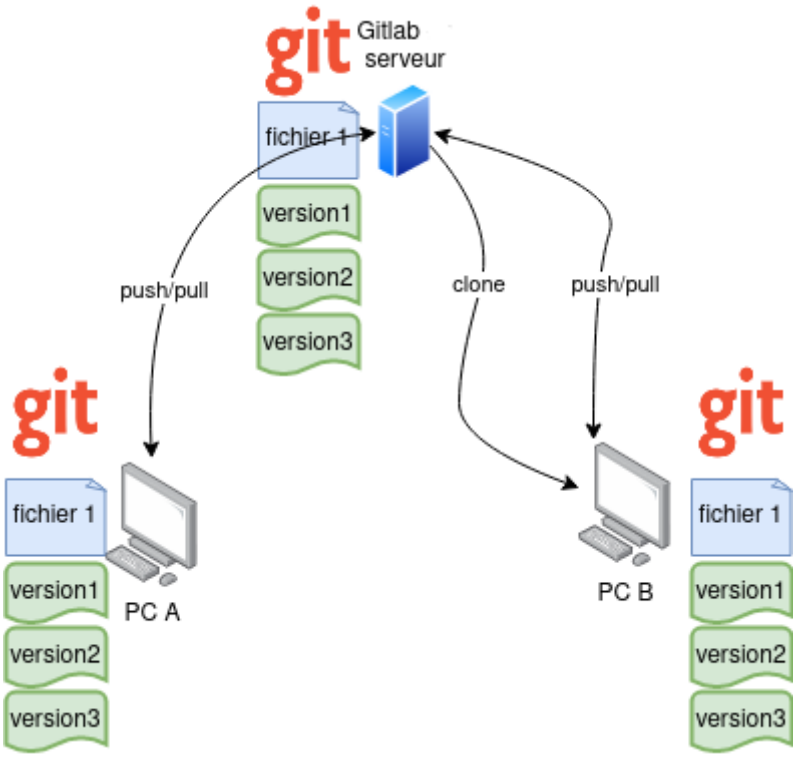
Git est un système de gestion de version **décentralisé**.

Cela signifie qu'il peut fonctionner sans serveur distant, uniquement sur votre machine de développement.

Ceci-dit, nous l'utiliserons toujours avec un serveur gitlab, qui contiendra la copie de votre travail local.

Cela vous permettra notamment :

- De travailler le même projet sur plusieurs ordinateurs
- De travailler à plusieurs sur le même projet.



Initialisation de Git sur votre ordinateur la première fois

Avant de pouvoir utiliser git sur votre ordinateur, il faut l'installer et le configurer

1. Installation

```
sudo apt install git
```

2. Configurer le nom d'utilisateur et le mail
   1. Configuration de votre nom

```
git config --global user.name "Votre nom"
```

   2. Configuration de votre mail

```
git config --global user.email "votre mail"
```

3. Configurer la sauvegarde de votre identifiant et mot de passe

> **Attention,** le mot de passe sera enregistré en clair sur le PC !

```
git config --global credential.helper store
```

Configuration d’un nouveau dépôt GIT (graphiquement depuis le serveur Gitlab)

La procédure décrite permet de :

- créer un dépôt sur le serveur,
- d’initialiser un dépôt sur votre PC,
- de sauvegarder (pousser) vos modifications locales sur le serveur.

Serveur Gitlab Perso

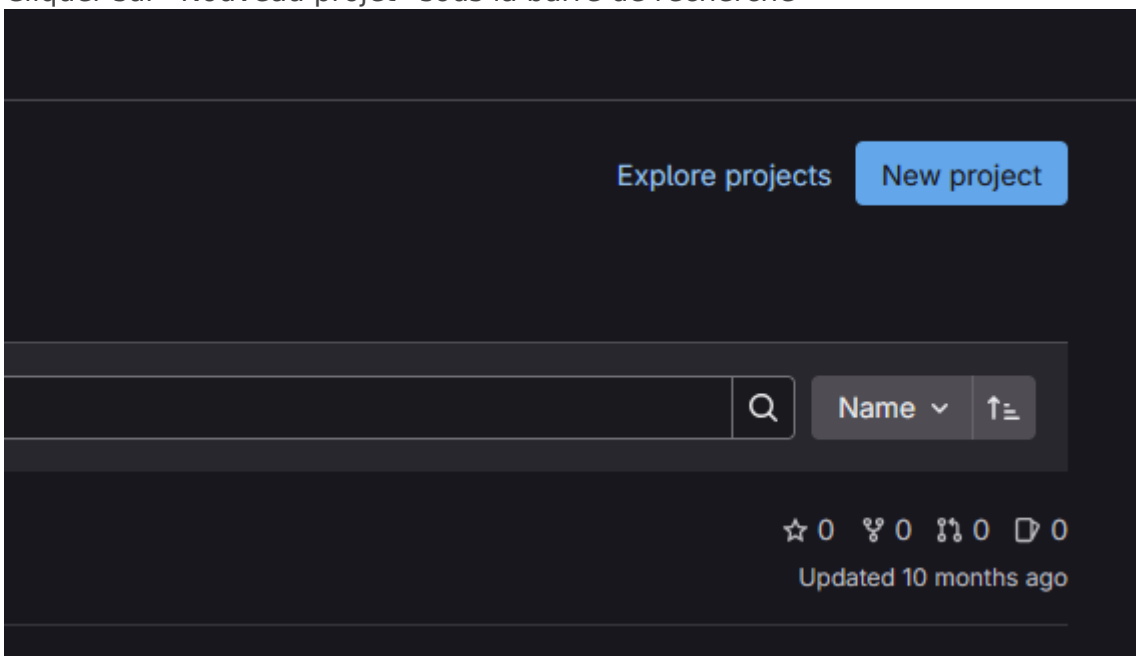
Gitlab Perso

Pour avoir un dépôt, il faut créer un nouveau projet. La création d’un dépôt vous permettra de sauvegarder votre travail lors des TP et projets.

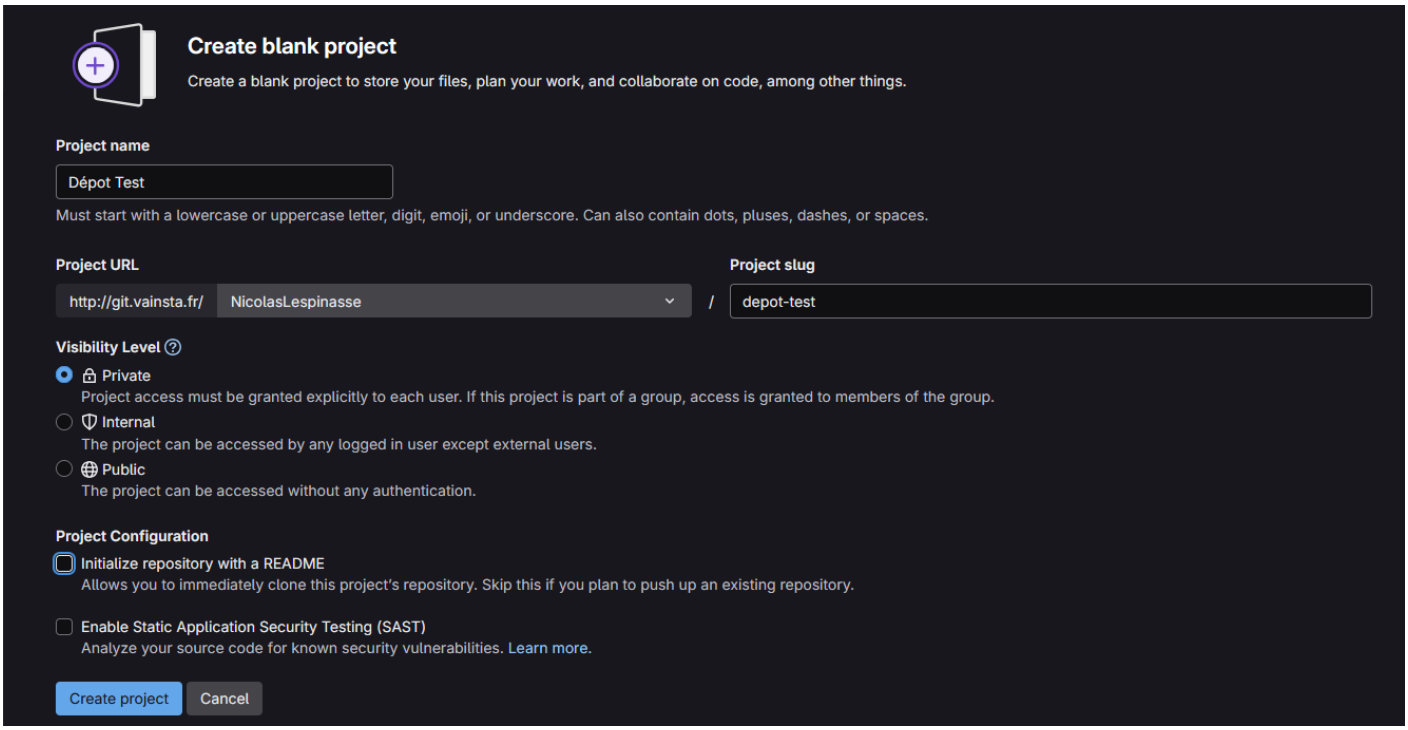
- Connectez-vous sur le serveur GIT

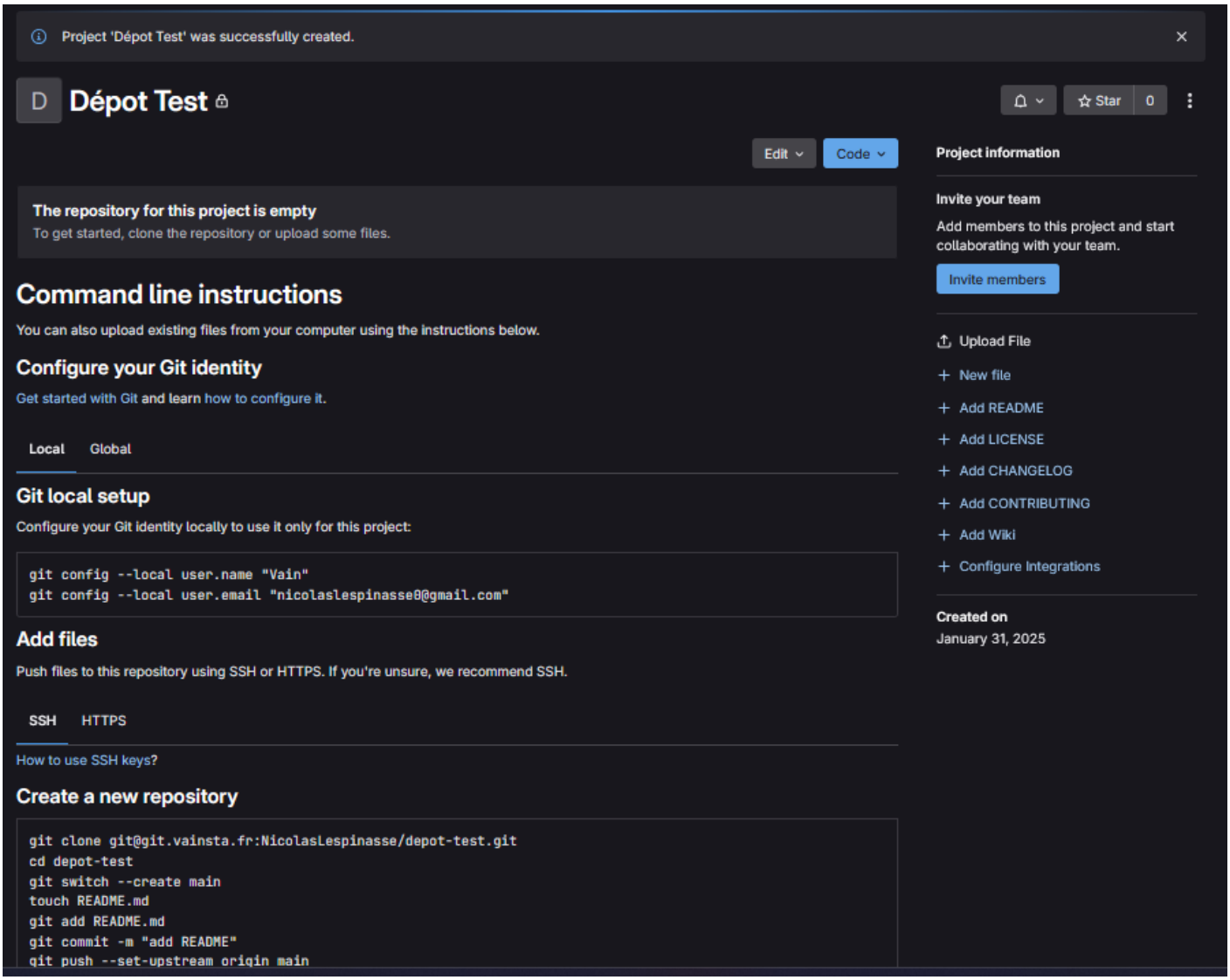
Création d’un nouveau projet

- Cliquer sur “Nouveau projet” sous la barre de recherche



- Choisir de créer un projet vide (Blank project)
  - Choisir un nom de projet
  - Choisir Visibility Level “Private”
  - **Décocher la case “Initialize repository with a README”**
  - Cliquer sur “Create project”





Configuration de votre dépôt

Les informations suivantes vous expliquent comment configurer et synchroniser votre dépôt distant et le travail sur votre ordinateur

> **Attention**, elles sont à personnaliser en fonction de vos besoins et des instructions du professeur

Nous allons détailler, les deux cas :

1. Cas 1 : Vous venez de cloner un dépôt qui ne vous appartient pas, il faudra modifier la configuration
2. Cas 2 : Vous voulez créer un nouveau dépôt

> **DANGER** : Pour toutes les manipulations, toujours se positionner dans le répertoire du projet.

CAS 1 POUR UN DÉPÔT CLONÉ

Dans le cas d'un dépôt cloné, votre dossier de projet est déjà initialisé. C'est à dire qu'il contient **déjà** un dossier `.git` et suit déjà les versions des fichiers que vous venez de cloner.

Par contre, l'adresse du dépôt distant est celle que vous venez d'utiliser pour cloner le projet. Il faut donc modifier cette adresse du dépôt distant pour sauvegarder votre travail.

1. Modifier l'url du dépôt distant
   Attention : Modifiez `votre_nom` et `votre_depot` par leurs valeurs réelles dans la ligne de code ci-dessous

   ```
   git remote set-url origin https://git.vainsta.fr/votre_nom/votre_depot.git
   ```

CAS2 : CRÉATION D'UN NOUVEAU DÉPÔT

Si le projet dans lequel vous travaillez est un nouveau projet (pas cloné depuis Internet), vous devez initialiser le dépôt, c'est à dire préciser que vous voulez suivre les versions des fichiers du répertoire courant.

Nous allons détailler ci-après la procédure pour initialiser votre dépôt, le connecter au dépôt distant et sauvegarder de votre travail

1. Se déplacer dans le dossier du projet

   ```
   cd Dossier_du_projet
   ```

2. Initialiser le dépôt local

   ```
   git init
   ```

3. Connecter votre dépôt local au dépôt distante
   Attention : Modifiez `votre_nom` et `votre_depot` par leurs valeurs réelles dans la ligne de code ci-dessous

   ```
   git remote add origin https://git.vainsta.fr/votre_nom/votre_depot.git
   ```

Sauvegarder votre travail

C'est la dernière étape, la plus importante. Il s'agit de sauvegarder régulièrement les différentes versions des fichiers qui composent votre projet.

Pour sauvegarder un ou plusieurs fichiers, vous devez effectuer successivement les commandes

- `git add` : Ajouter un fichier à la liste des fichiers suivis
- `git commit` : Enregistrer (localement) les modifications sur un ou plusieurs fichiers
- `git push` : Pousser les nouvelles versions sur le serveur distant

1. Ajouter un fichier à GIT

```
git add nom_fichier
```

2. Commiter le(s) fichier(s)

```
git commit -am "Feature1 : Ajout du if pour gérer les différents cas"
```

3. Pousser vos modifications sur le serveur

```
git push -u origin master
```

Remarque : Une fois qu'un fichier a été ajouté une fois, il est connu de `git`. L'option `-am` de `git commit` précise d'enregistrer toutes les modifications des fichiers suivis.

Attention : Après un `git push` Vérifier **TOUJOURS** que les fichiers sur le serveur sont bien ceux que vous voulez

Configuration d'un nouveau dépôt GIT (en ligne de commande depuis votre poste)

Lorsque vous serez plus aguerri, vous pourrez utiliser cette procédure qui vous permet de directement créer le dépôt distant depuis votre PC.

- Initialiser le dépôt dans le dossier du projet.

```
git init
```

- Ajouter l'adresse de l'origin depuis votre pc

```
git remote add origin https://git.vainsta.fr/votre_nom/votre_depot.git
```

- Ajouter vos fichiers

```
git add nom_fichier
```

- Commiter vos changements

```
git commit -m "mon super message"
```

- Pousser vos modifications. Le dépôt distant sera automatiquement créé.

```
git push origin master
```

Commandes utiles de git

Git est un logiciel très complet, mais qui aussi peut s’avérer complexe à prendre en main et à manipuler. Il faut donc acquérir les bons reflexes en cas d’erreurs.

Cloner un dépôt

Cloner un dépôt vous permet de récupérer tout l’historique (fichiers et répertoires) du dépôt en une seule commande

- exemple avec le dépôt de vlc

```
git clone https://github.com/videolan/vlc.git
```

Commandes indispensables de git

Le premier des reflexes est de savoir si vous êtes dans le bon répertoire et à quel serveur votre dépôt local est relié.

- Connaître les fichiers modifiées et/ou à pousser sur le serveur

```
git status
```

- Connaître le dépôt distant auquel je suis connecté

```
git remote -v
```

- Connaître l’historique des commits

```
git log
```

- Connaître la branche sur laquelle je travaille

```
git branch
```

Gestion des identifiants !

Par défaut, git ne se souvient pas de vos identifiants. Vous pouvez modifier le comportement par défaut, soit en lui précisant de les sauvegarder 15min, soit en dur sur l'ordinateur.

- Pendant 15 minutes

```
git config --global credential.helper cache
```

- Tout le temps (Attention, le mot de passe (en clair) est enregistré sur le PC !)

```
git config --global credential.helper store
```

Fichiers de configurations

.gitconfig

La configuration de git est par défaut sous Linux dans le fichier .gitconfig dans le répertoire de l'utilisateur. vous pouvez l'éditer à la main ou utiliser la commande `git config`

Vous pouvez définir des alias dans .gitconfig comme celui-ci qui présente de manière plus agréable les log de commit.

```
lg = log --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit
```

Une fois l'alias défini, il est utilisable avec git

```
git lg
```

Revision #5
Created 23 May 2024 23:35:25 by Nicolas

Updated 9 February 2025 14:11:07 by Nicolas