

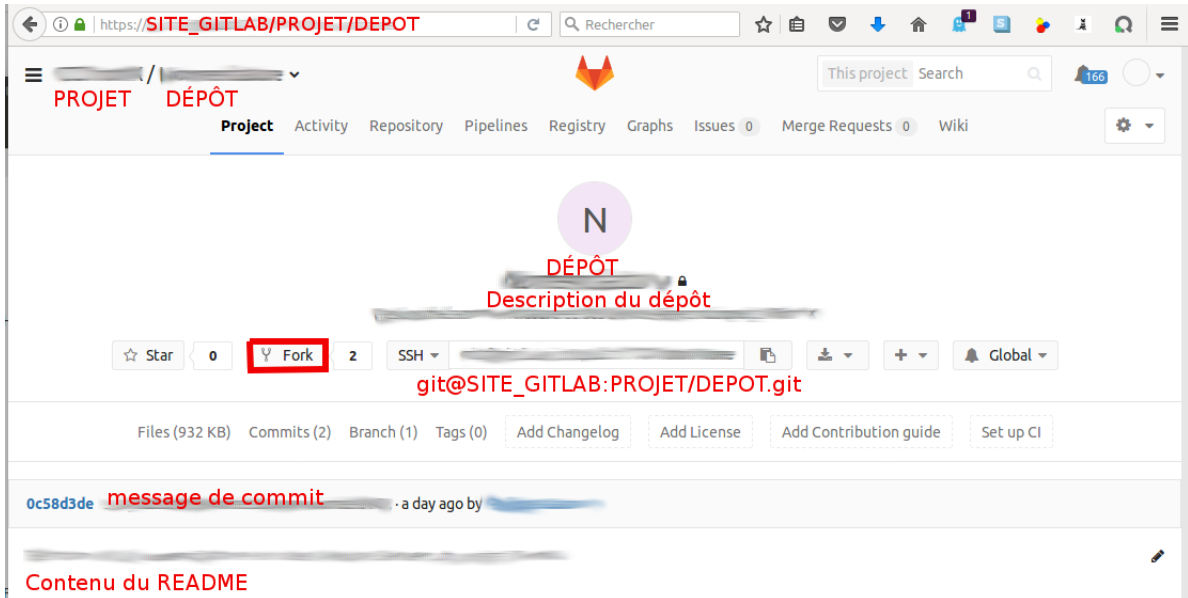
Pour cloner un dépôt :

1. aller sur le projet https://SITE_GITLAB/PROJET/DEPOT

cliquer sur le bouton fork

sélectionner son compte (celui proposé par défaut)

Ce bouton est apparu à partir du moment où un fichier a été ajouté (les bonnes pratiques recommandent l'ajout d'un README)



2. Configurer son compte SSH pour qu'il utilise sa clé SSH de manière transparente : dans son fichier de configuration SSH (sous linux, ~/.ssh/config), ajouter les éléments suivants

```
Host SITE_GITLAB
User IDENTIFIANT
IdentityFile [CLEF SSH]
```

(indiquer la clé ssh personnelle à la place de [CLEF SSH]).

3. aller dans son compte https://SITE_GITLAB/IDENTIFIANT

aller dans le projet DEPOT et vérifier qu'on voit bien l'adresse SSH : «git@SITE_GITLAB:IDENTIFIANT/DEPOT.git». exécuter ensuite

```
>git clone CHEMIN
```

où chemin est l'adresse SSH ci-dessus.

Normalement, ça donne donc :

```
>git clone git@SITE_GITLAB:IDENTIFIANT/DEPOT.git
```

4. On a donc un dépôt local Git initialisé avec le fichier README.

Pour configurer son dépôt local

```
>git config --global user.email "L'ADRESSE ELECTRONIQUE"
>git config --global user.name "NOM À AFFICHER POUR CHAQUE PARTICIPATION"
```

5. pour pouvoir travailler à plusieurs, il vaut mieux utiliser la méthode (workflow) merge request. Cela oblige à avoir un dépôt gitlab de projet, et un compte gitlab personnel par lequel passer pour faire des merge request.

Pour ça, vérifier que le dépôt git est configuré pour récupérer et pousser les modif sur le dépôt Gitlab d'origine

```
>git remote -v
origin    git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (fetch)
origin    git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (push)
```

Ajouter un lien nommé «upstream» vers le dépôt de référence DEPOT

```
>git remote add upstream git@SITE_GITLAB/PROJET/DEPOT.git
```

Vérifier le résultat

```
>git remote -v
origin    git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (fetch)
origin    git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (push)
upstream  git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (fetch)
upstream  git@SITE_GITLAB:IDENTIFIANT/DEPOT.git (push)
```

Désormais, pour récupérer les dernières modif du dépôt de référence (qui peut être plus à jour que le dépôt de son répertoire Gitlab personnel), on fait

```
>git fetch upstream
```

Pour intégrer des modifications dans le dépôt de référence, on va dans le répertoire Gitlab personnel et on crée un «merge request» :

cliquer sur l'onglet «Merge Requests» au milieu, puis sur le bouton vert à droite «New Merge Request» (cf. didacticiel dédié).

Quand on veut synchroniser le dépôt local avec le dépôt de référence, il faut :

1. importer les branches et les mises-à-jour (commit) de chacune. Les commit de master seront enregistrées dans la branche upstream/master

```
>git fetch upstream
remote: Counting objects: 75, done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 62 (delta 27), reused 44 (delta 9)
Unpacking objects: 100% (62/62), done.
From https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
 * [new branch]      master      -> upstream/master
```

2. mettre à jour la branche master locale

```
>git checkout master
Switched to branch 'master'
```

3. fusionner les changements de upstream/master vers le master local. Cela évite de perdre les changements de la branche locale

```
>git merge upstream/master
Updating a422352..5fdff0f
Fast-forward
 README                |    9 -----
 README.md             |    7 ++++++
 2 files changed, 7 insertions(+), 9 deletions(-)
 delete mode 100644 README
 create mode 100644 README.md
```

Si la branche locale n'a aucun commit unique, Git fera un fast-forward

```
>git merge upstream/master
Updating 34e91da..16c56ad
Fast-forward
 README.md             |    5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)
```