

1 Création de projet

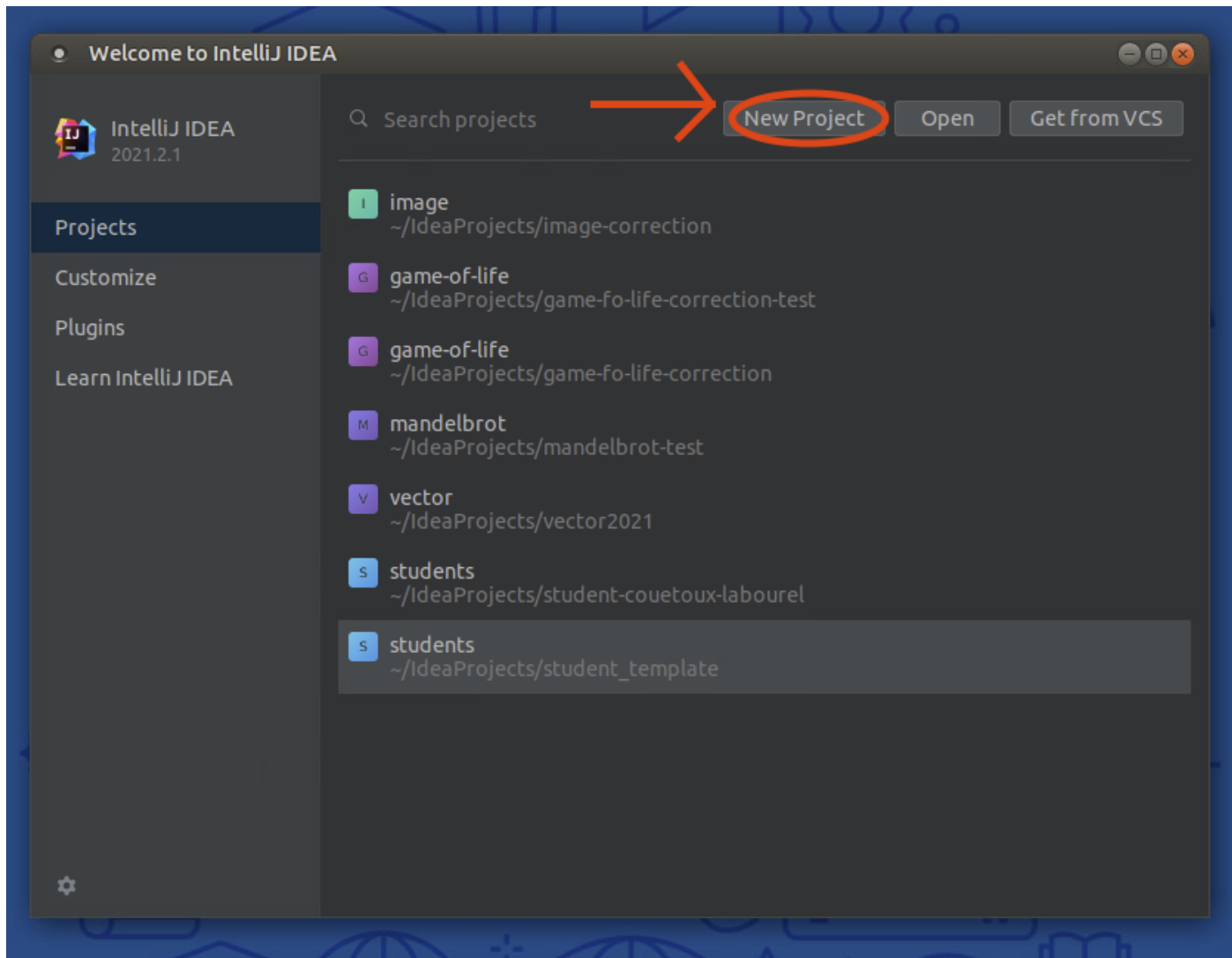
Le but de ce TP est d'apprendre à configurer un projet en utilisant IntelliJ IDEA ou bien entièrement en ligne de commande.

Si vous effectuez ce TP depuis votre machine personnelle, il vous faut installer les outils suivants :

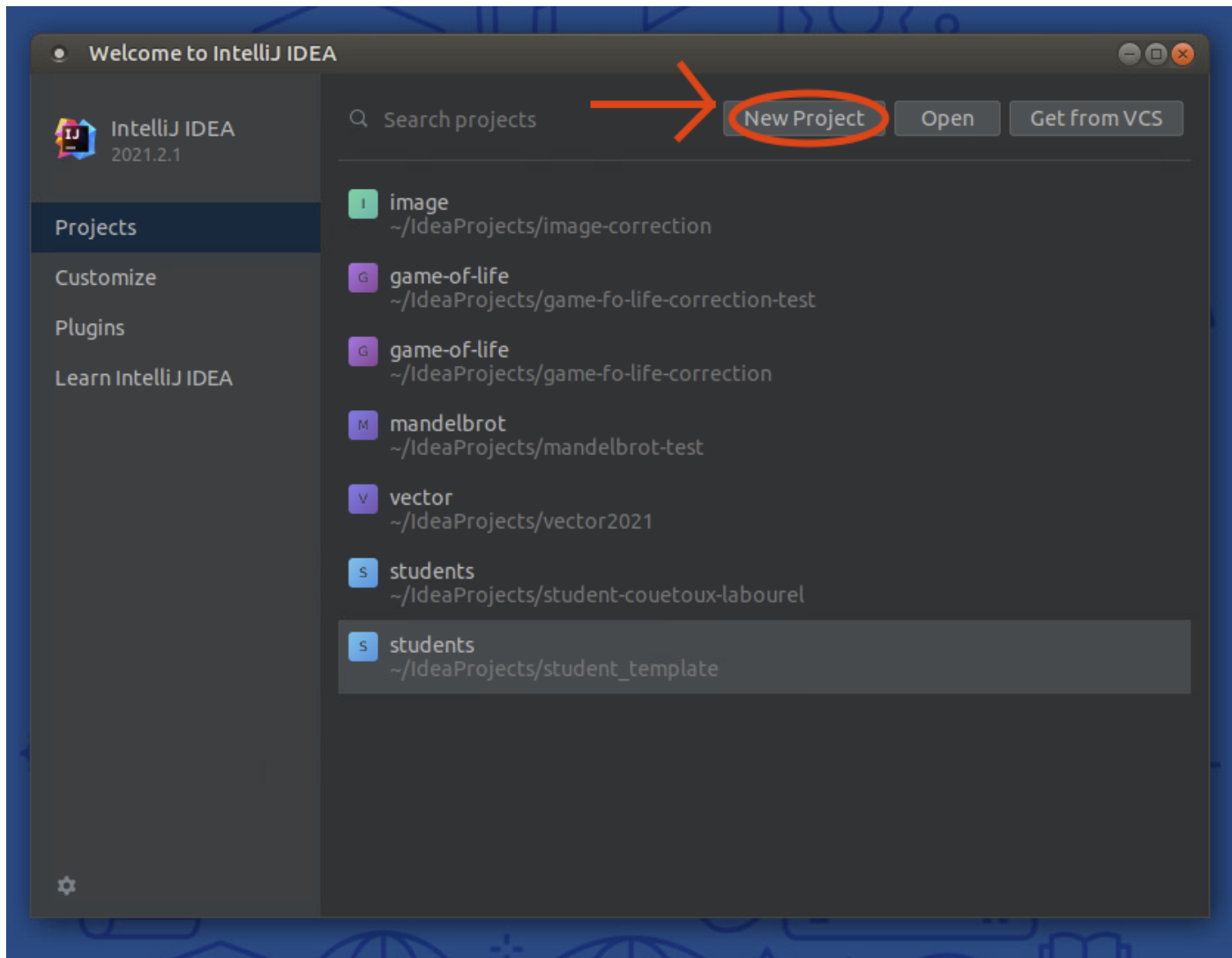
- **Terminal shell** si vous êtes sous windows, nous vous conseillons d'installer ubuntu pour avoir accès à un terminal shell. Sous MacOS et linux, un terminal shell est disponible de base.
- **JDK version 16 ou plus** la méthode d'installation dépend de votre système d'exploitation :
 - **Windows** : Télécharger et exécuter l'Installeur windows
 - **MacOS** : Télécharger et exécuter l'Installeur MacOS
 - **Linux** : exécuter la commande suivante dans le terminal `sudo apt install openjdk-17-jdk`. Un mot de passe administrateur vous sera demandé.
- **Gradle** : Suivez les instructions au lien suivant : [install gradle](#)
- **git** : Télécharger et installer git au lien suivant : [download git](#)
- **IntelliJ IDEA** : vous pouvez télécharger IntelliJ IDEA ou bien JetBrains Toolbox qui est un outils pour gérer les IDE proposé par l'entreprise JetBrains. En tant qu'étudiant, vous avez d'ailleurs accès à des licences gratuites pour leurs produits et notamment la version ultime d'IntelliJ IDEA. Vous pouvez aussi utiliser Eclipse ou Visual studio code qui sont des outils similaires.

Création de projet versionné à l'aide d'IntelliJ

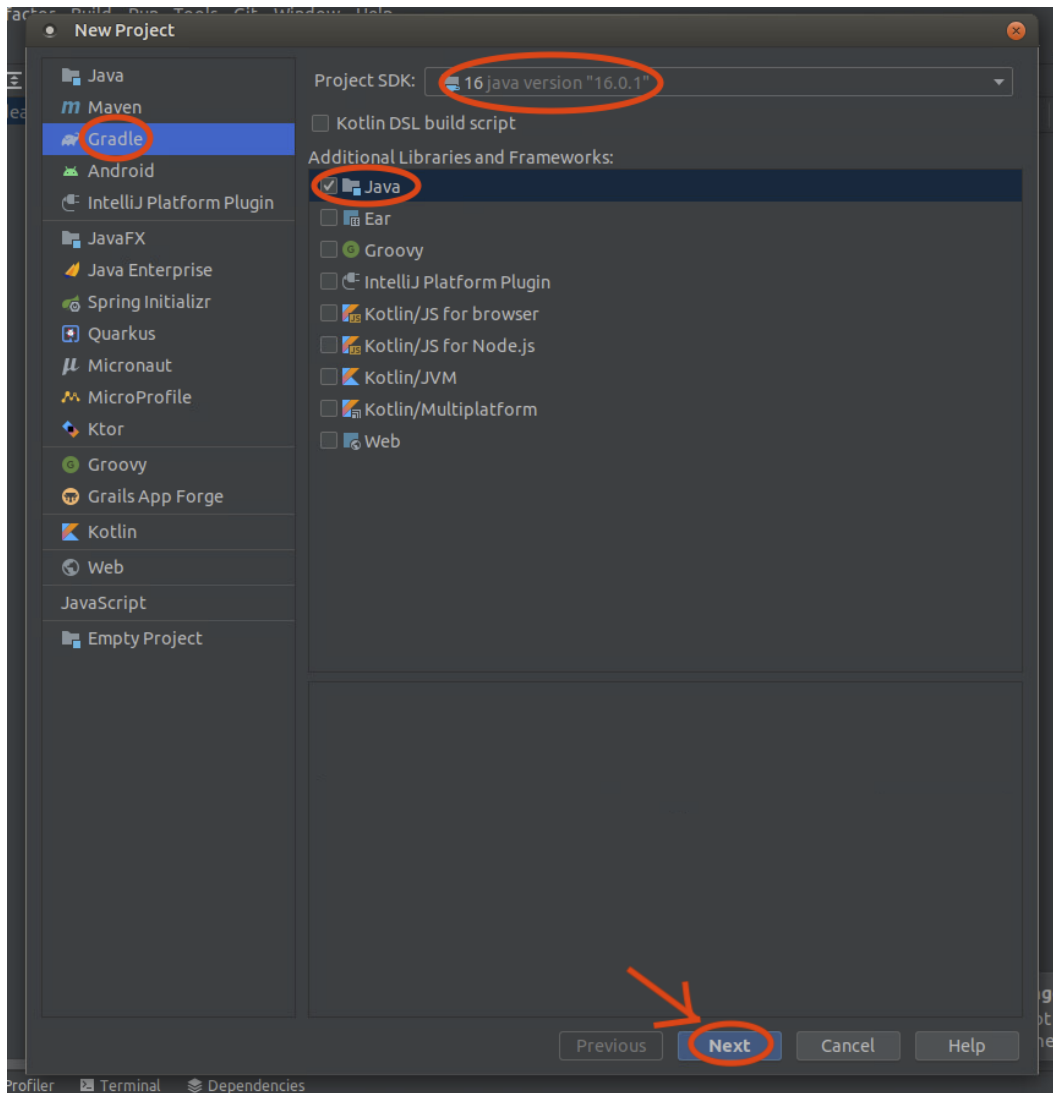
- Lancez IntelliJ IDEA
- Commencez la création d'un nouveau projet soit en fermant tout vos projets en cours via le menu **file** -> **Close project** et en cliquant sur **New Project**.



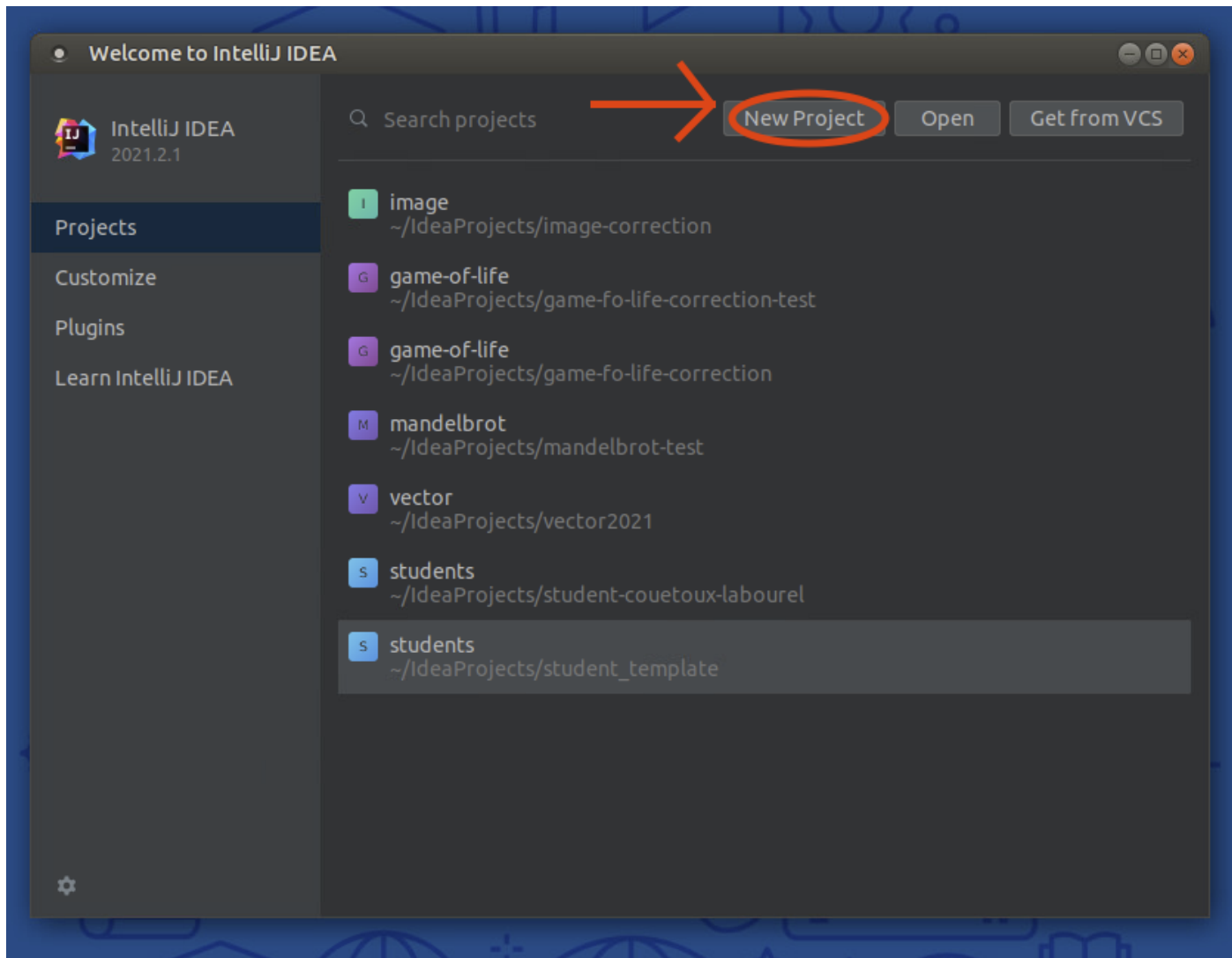
Vous pouvez aussi créer un nouveau projet en allant dans le menu `file -> New -> Project`.



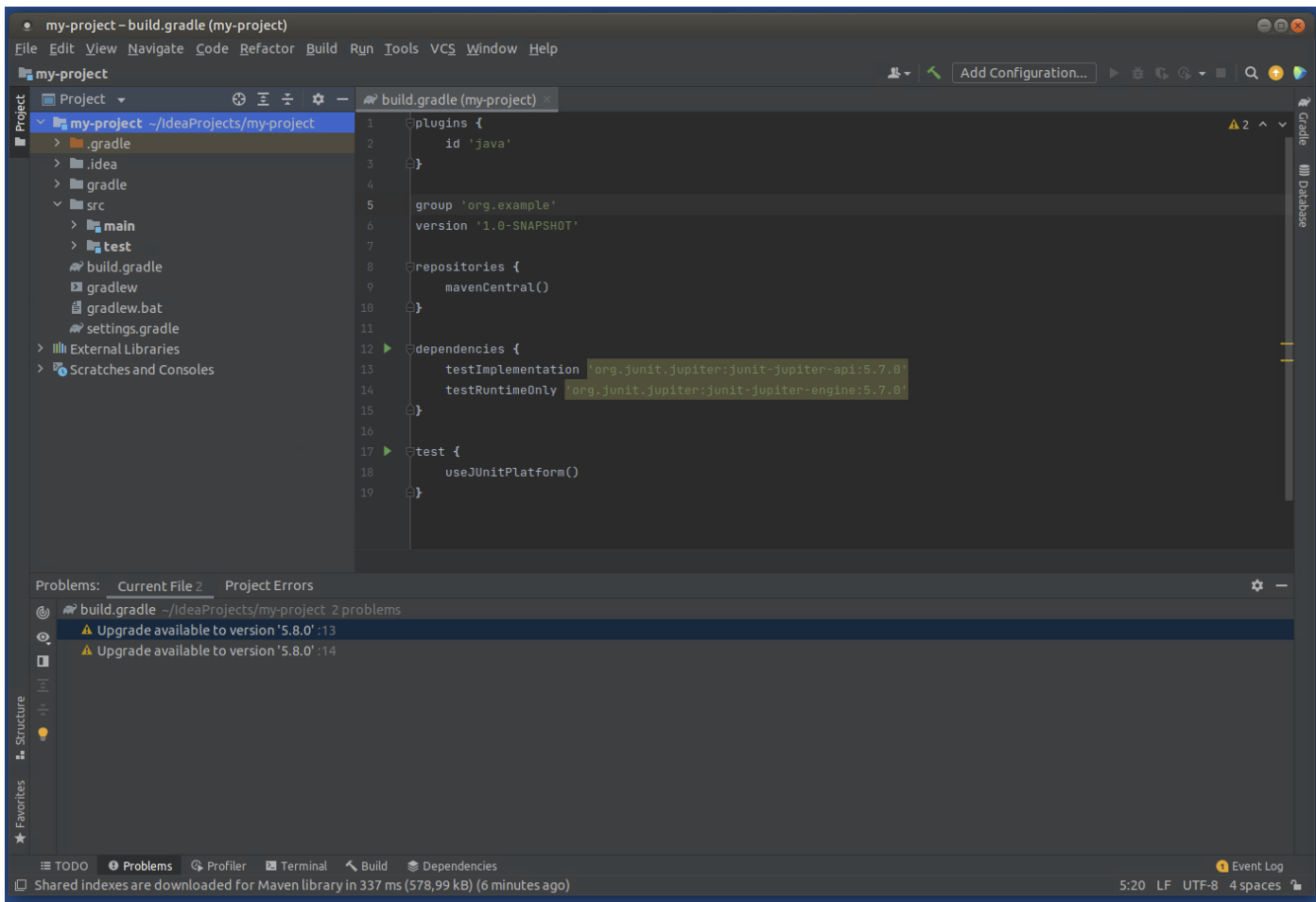
- Créer un projet gradle en choisissant **gradle** dans la liste à gauche. Gradle est ce que l'on appelle un moteur de production. C'est un outil pour compiler et exécuter les projets. Il est particulièrement utile pour gérer les dépendance de bibliothèques. En fait, cet outil télécharge automatiquement les bibliothèques configurées dans le projet dans le fichier `build.gradle`. Cochez `java` (si ce n'est pas déjà fait) en tant qu'**Additionnal Libraries and Frameworks** et vérifier que le **Project SDK** a un numéro de version supérieure ou égal à 16. Cliquer sur le bouton **Next**.



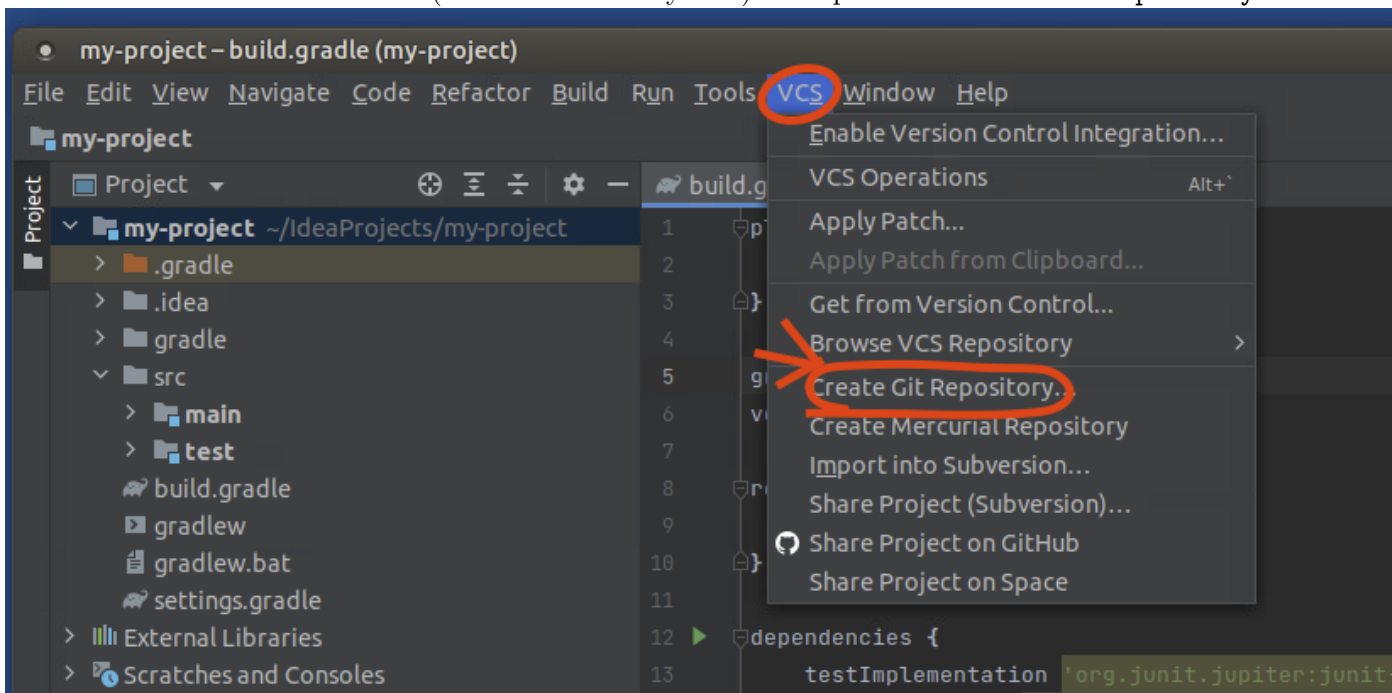
— Entrer le nom de votre projet et validez sa création en cliquant sur sur le bouton `finish`.



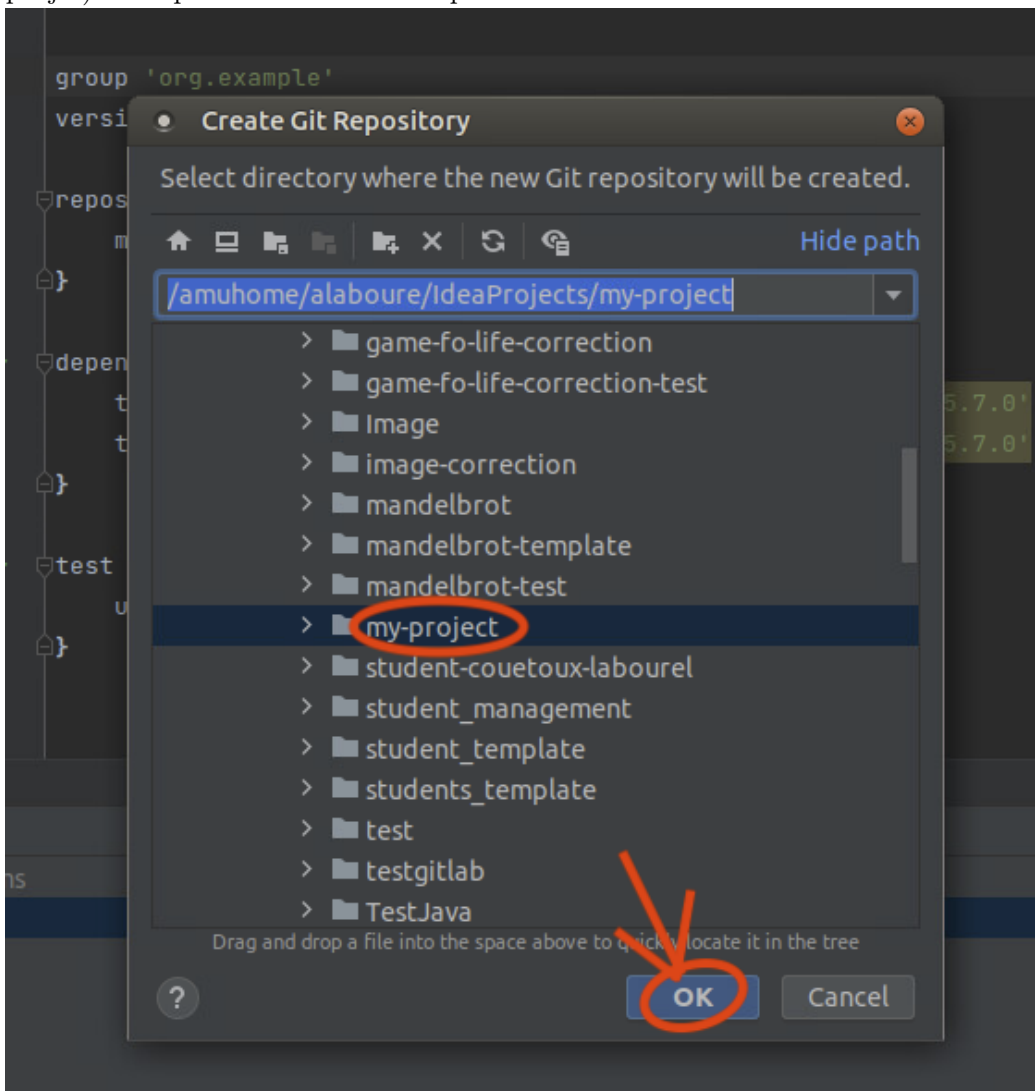
— Votre projet devrait se configurer automatiquement et vous devriez obtenir l’affichage suivant.



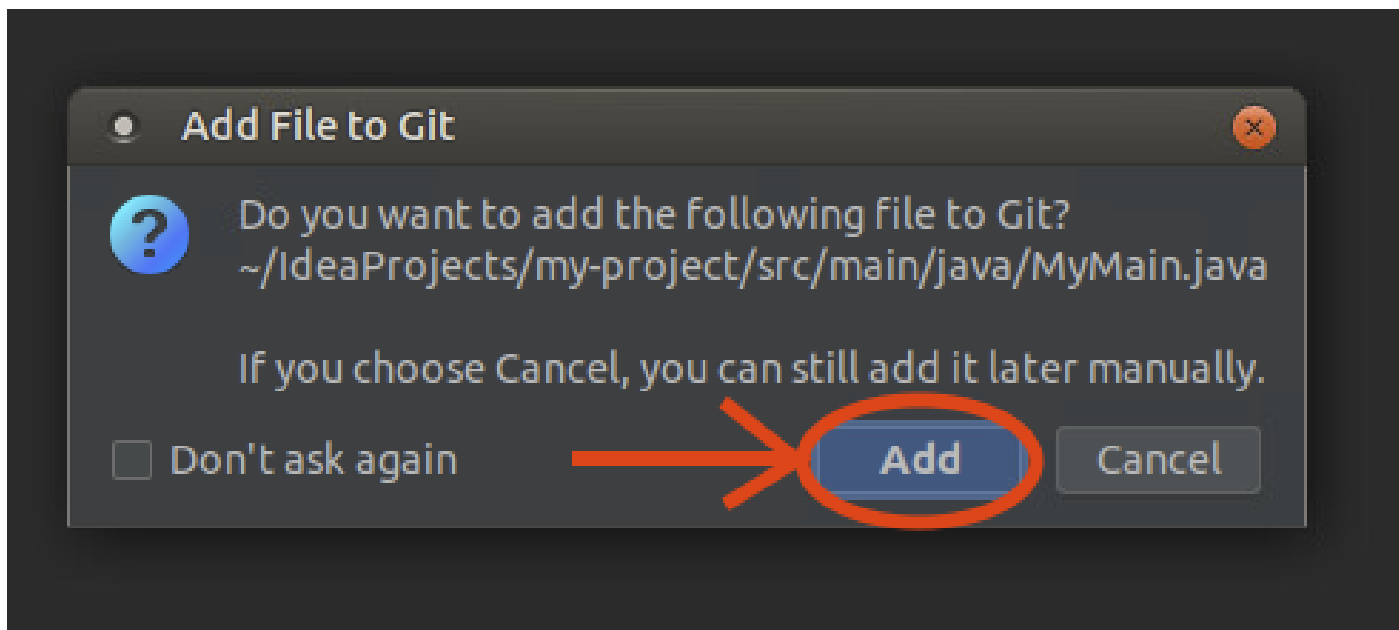
- vous allez maintenant créer un dépôt local git qui va permettre de faire de la gestion de version en local. Pour cela allez dans le menu VCS (Version Control System) et cliquez sur Create Git Repository.



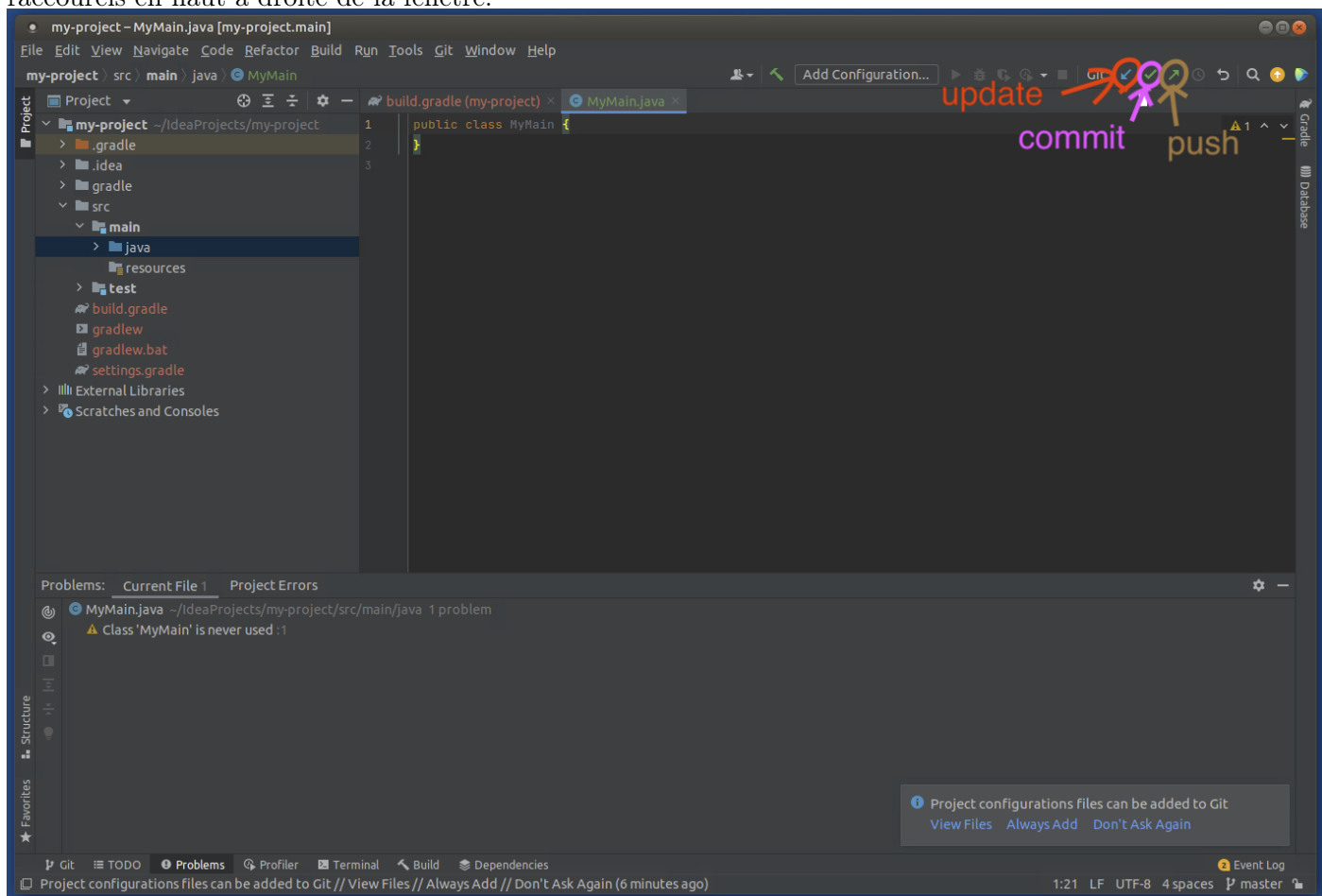
- Choisissez le répertoire pour votre dépôt (normalement le choix de base est le répertoire contenant votre projet) et cliquez sur le bouton `ok` pour valider.



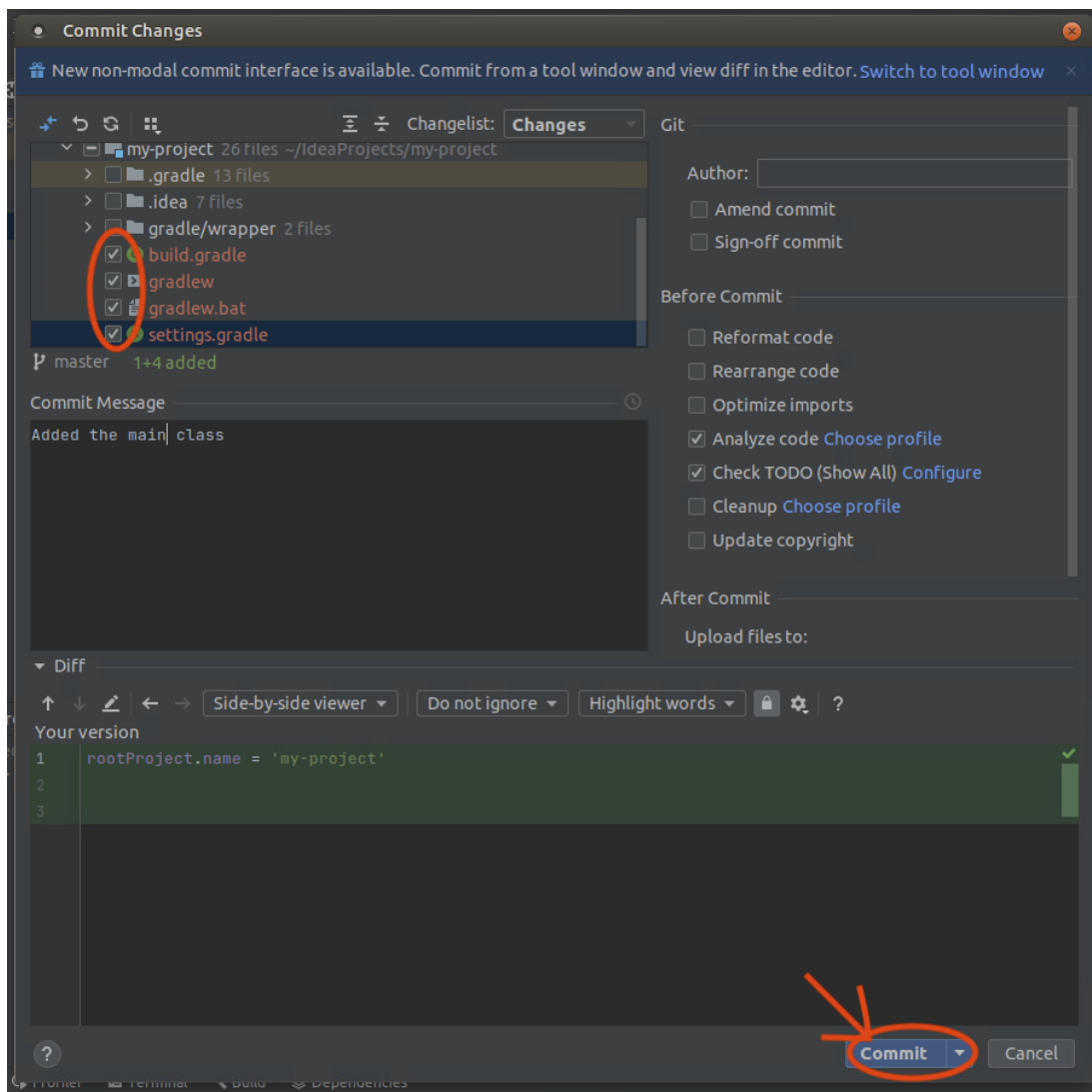
- Créez une classe dans le répertoire `src -> main -> java`. Une fenêtre va s'ouvrir pour demander si vous voulez rajouter le fichier contenant votre classe dans le dépôt `git`. Cliquez sur le bouton `add` pour validez cet ajout.



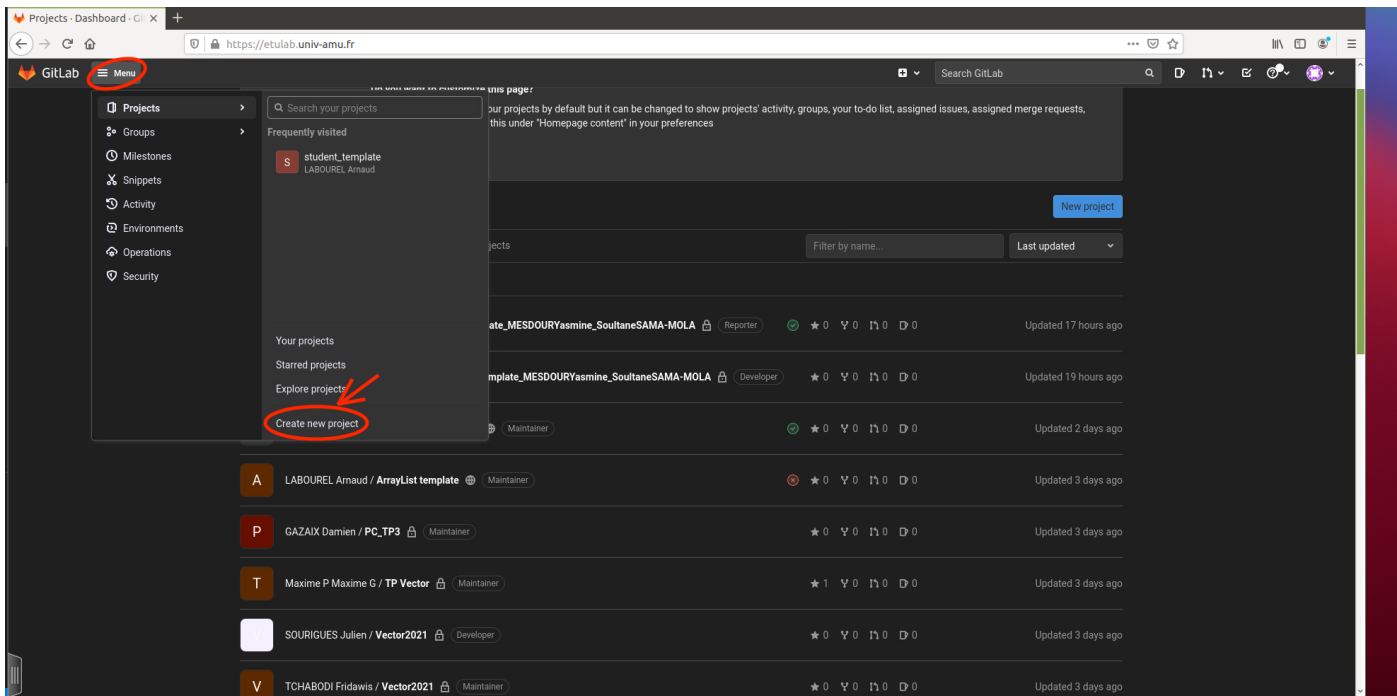
- Vous pouvez effectuer les opérations de base de git (update du projet, commit et push) grace aux raccourcis en haut à droite de la fenêtre.



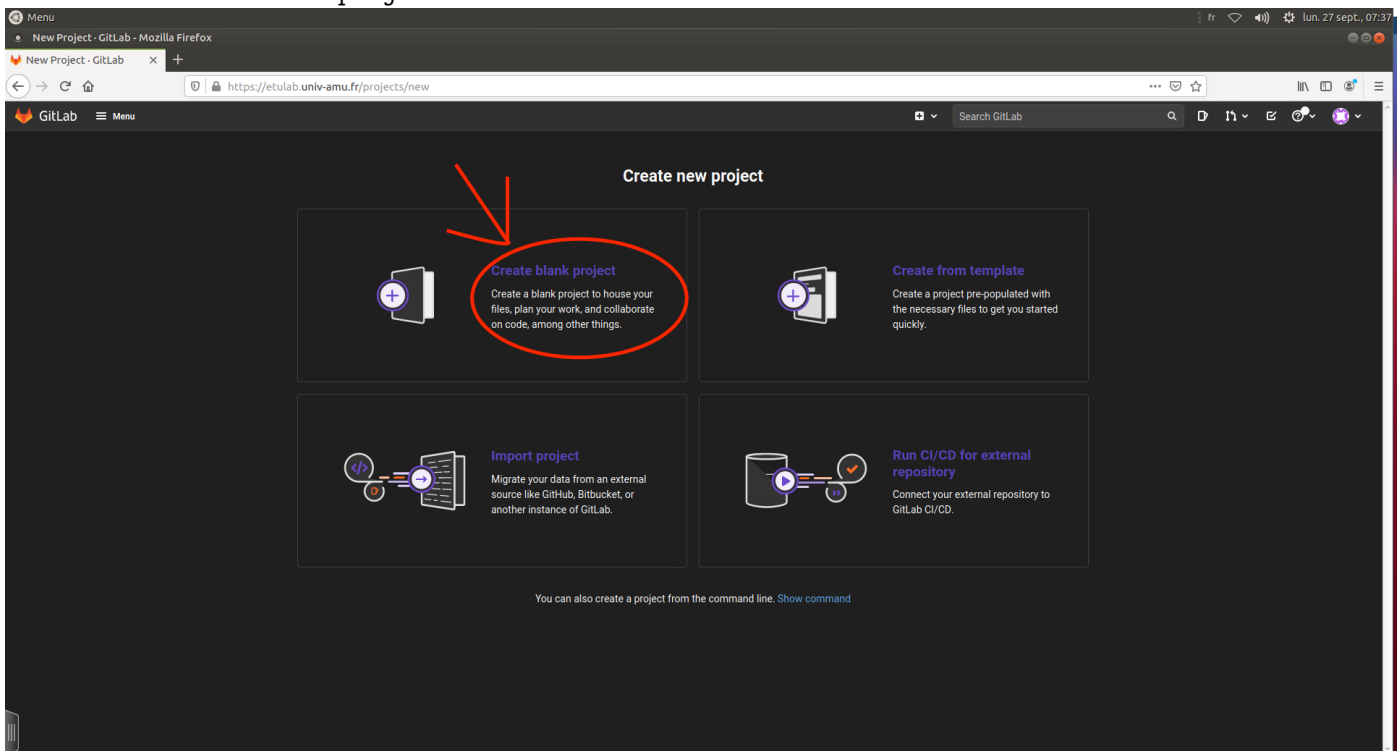
- Faites le premier commit de votre projet en ajoutant en plus de vos fichiers .java les fichiers de configuration du git : build.gradle, gradlew, gradlew.bat et settings.gradle.



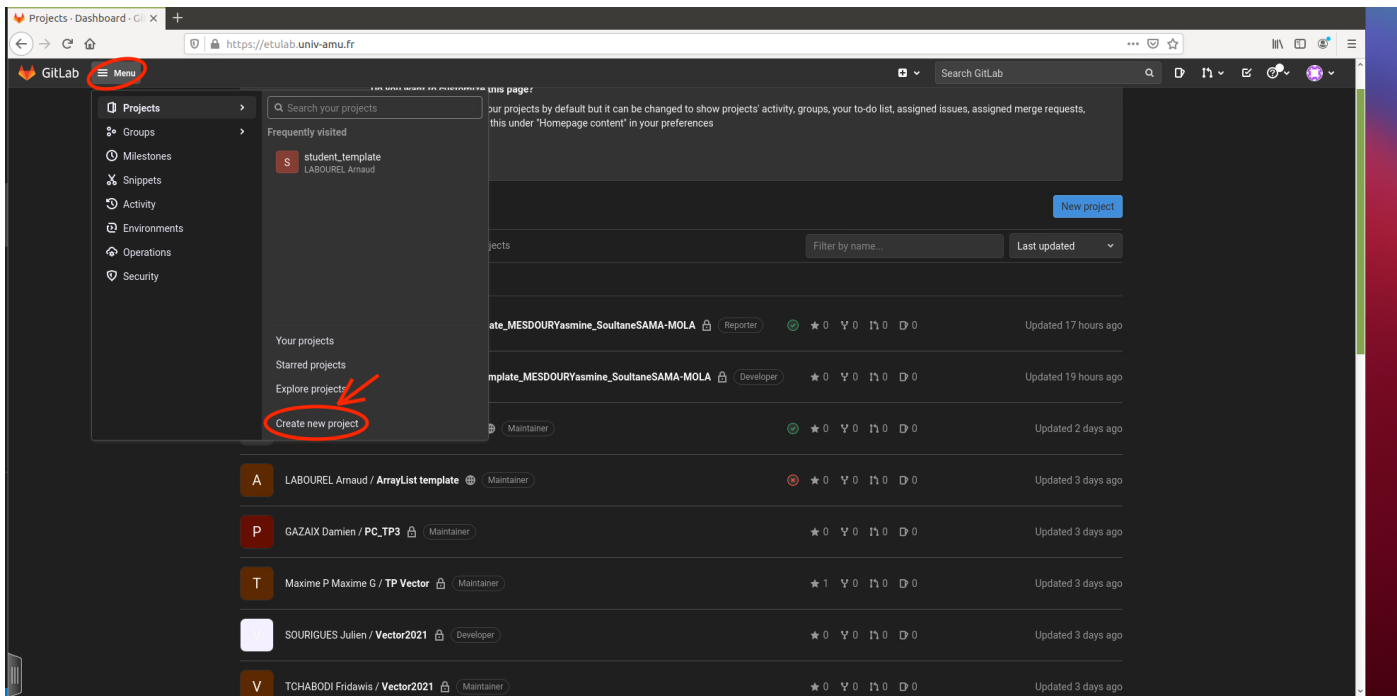
— Connectez vous via à un navigateur à votre compte etulab et créez un nouveau projet en cliquant sur menu puis Create new project.



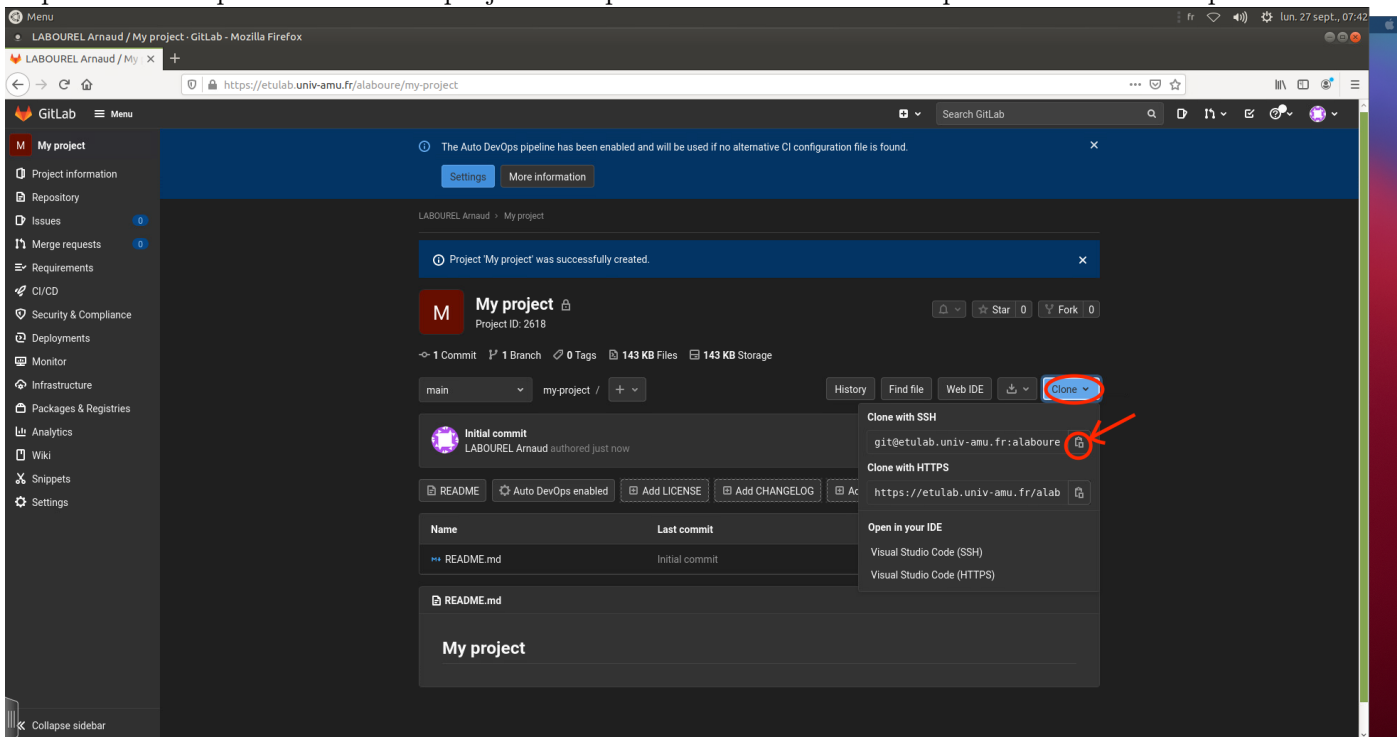
— Choisissez Create blank project.



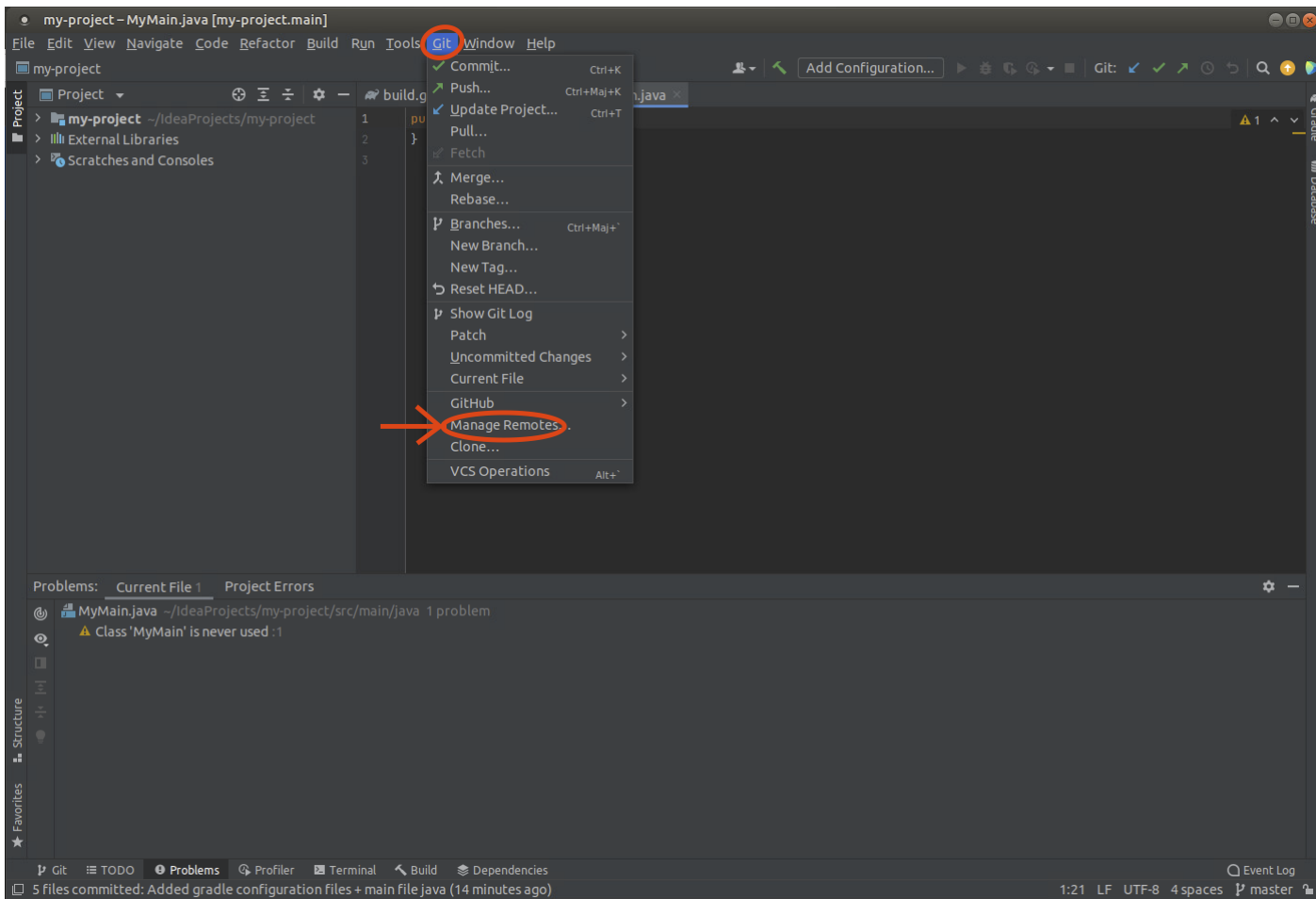
— Choisissez un nom pour votre projet et validez la création en cliquant sur le bouton Create project.



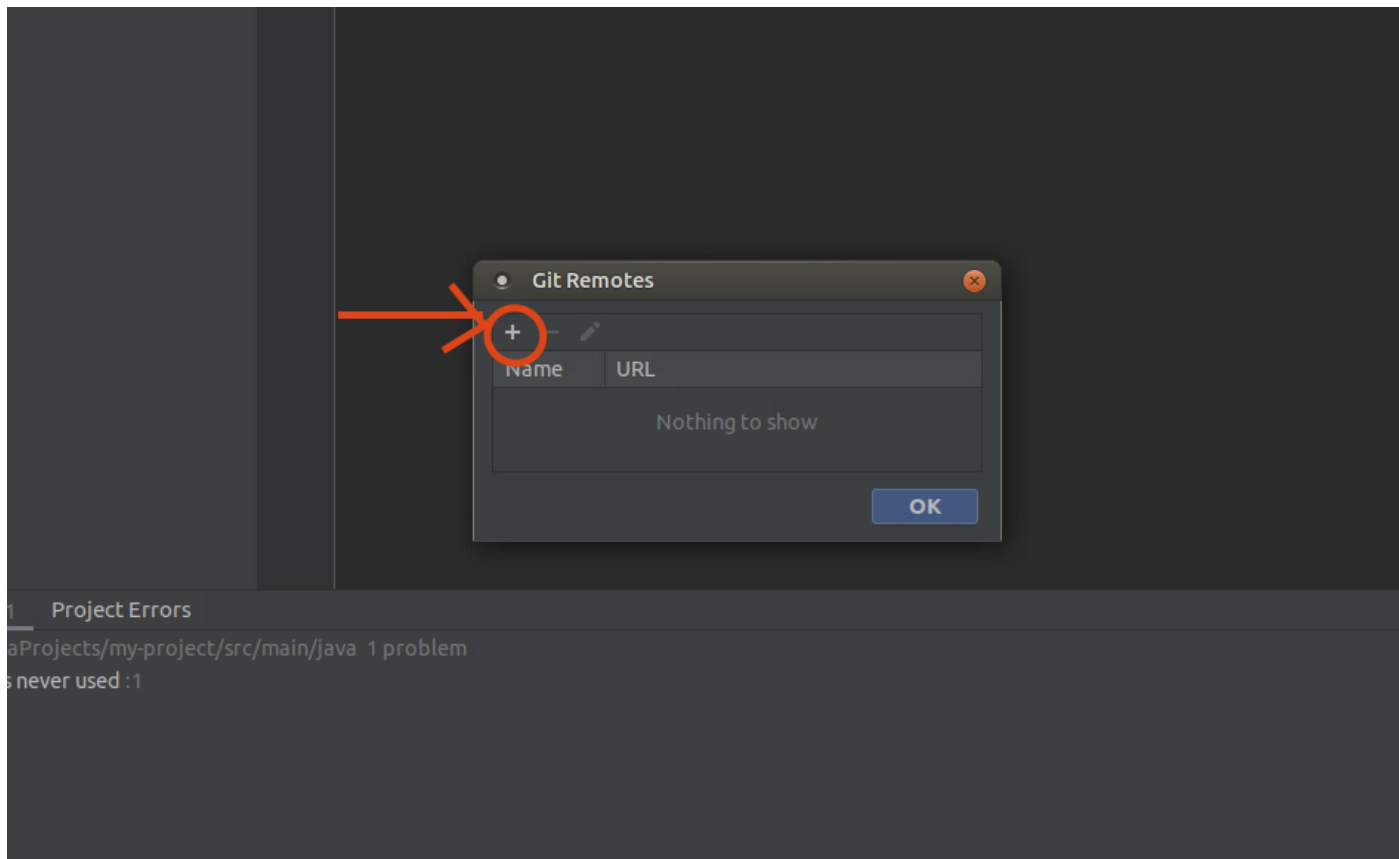
— Copiez l'adresse pour cloner votre projet en cliquant sur le bouton clone puis sur l'icône de copie ssh.



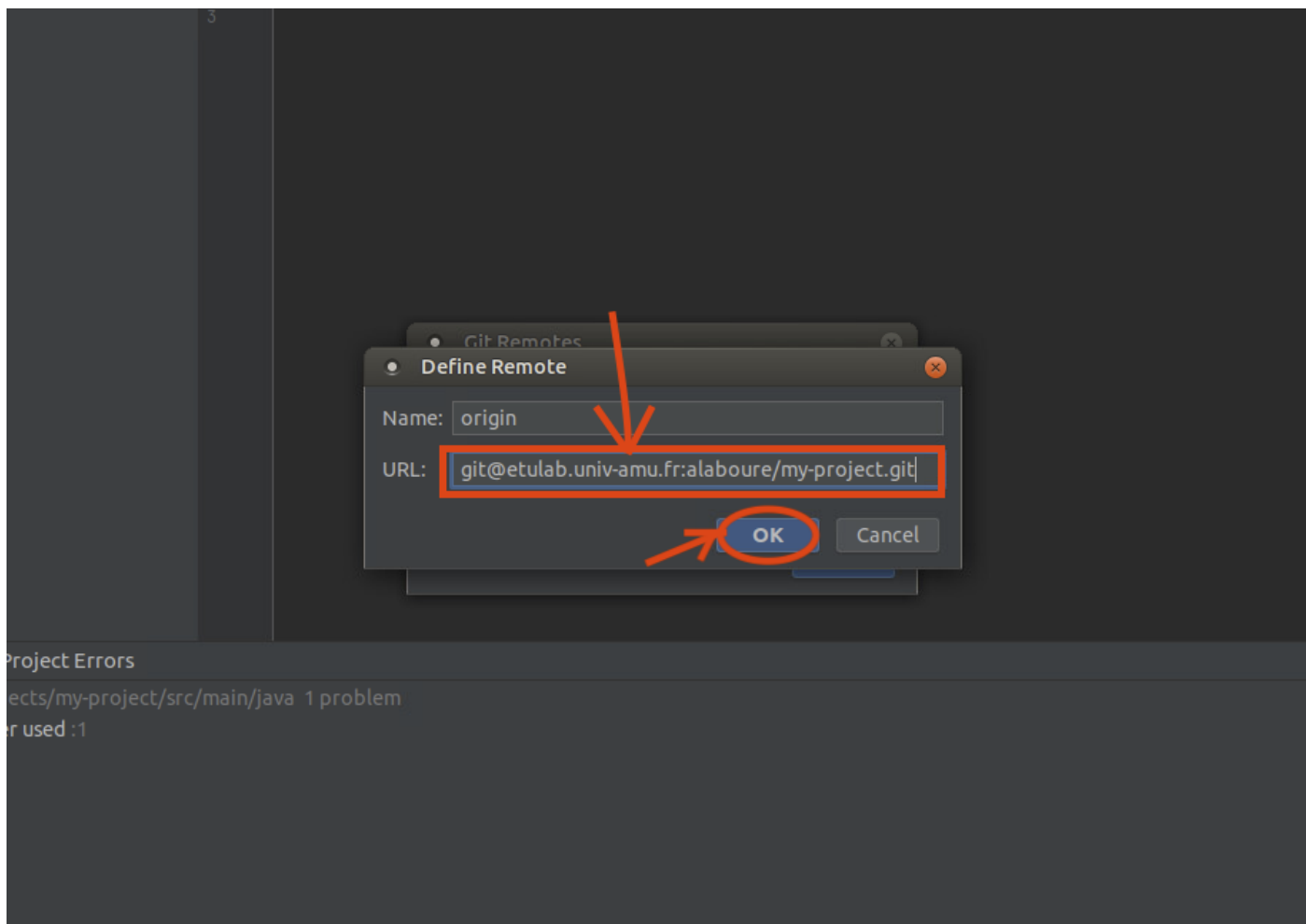
— Retournez sur votre projet sur IntelliJ et allez dans le menu : git puis Manage Remotes.



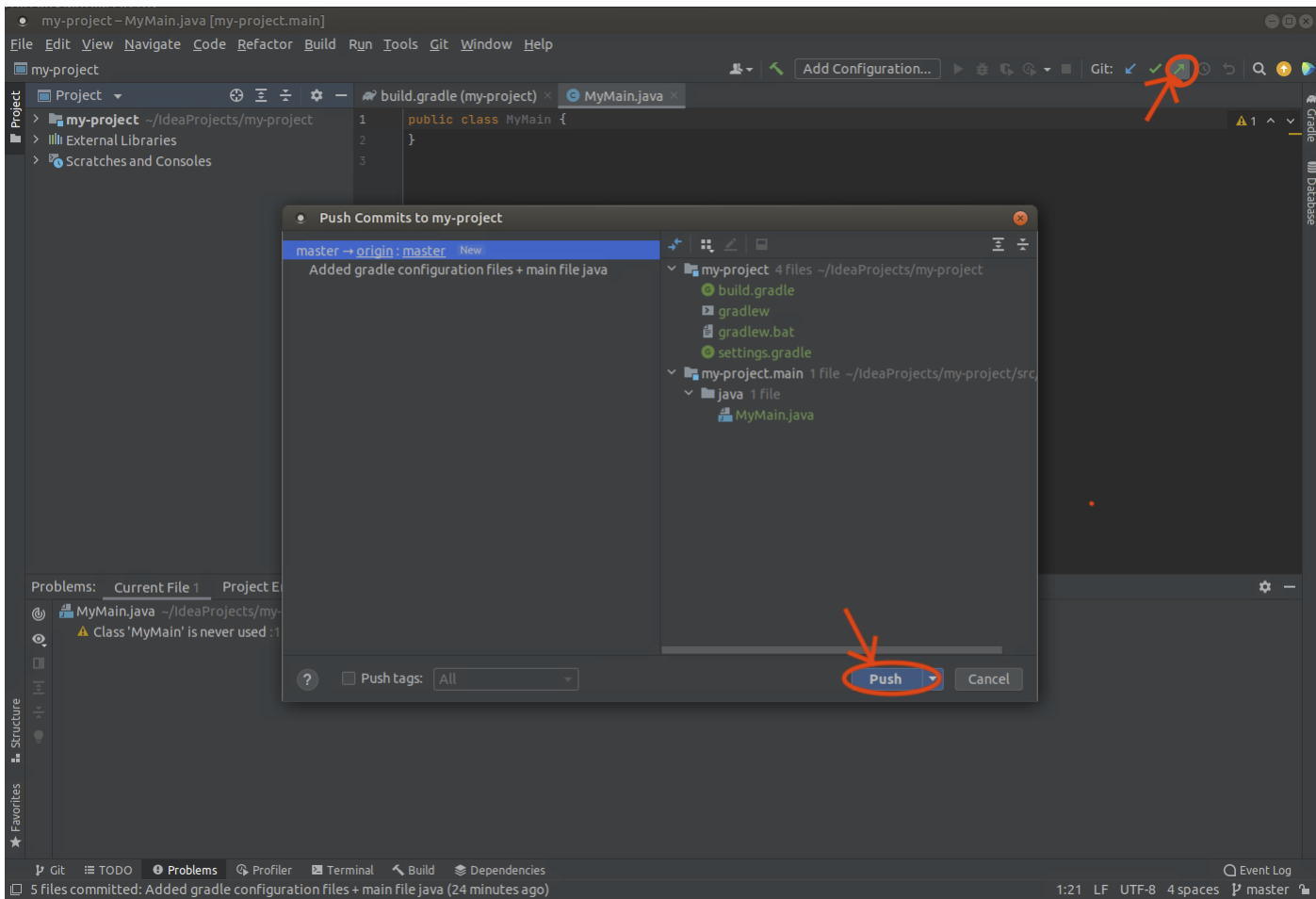
— Cliquez sur le bouton + pour ajouter un dépôt distant à votre dépôt local.



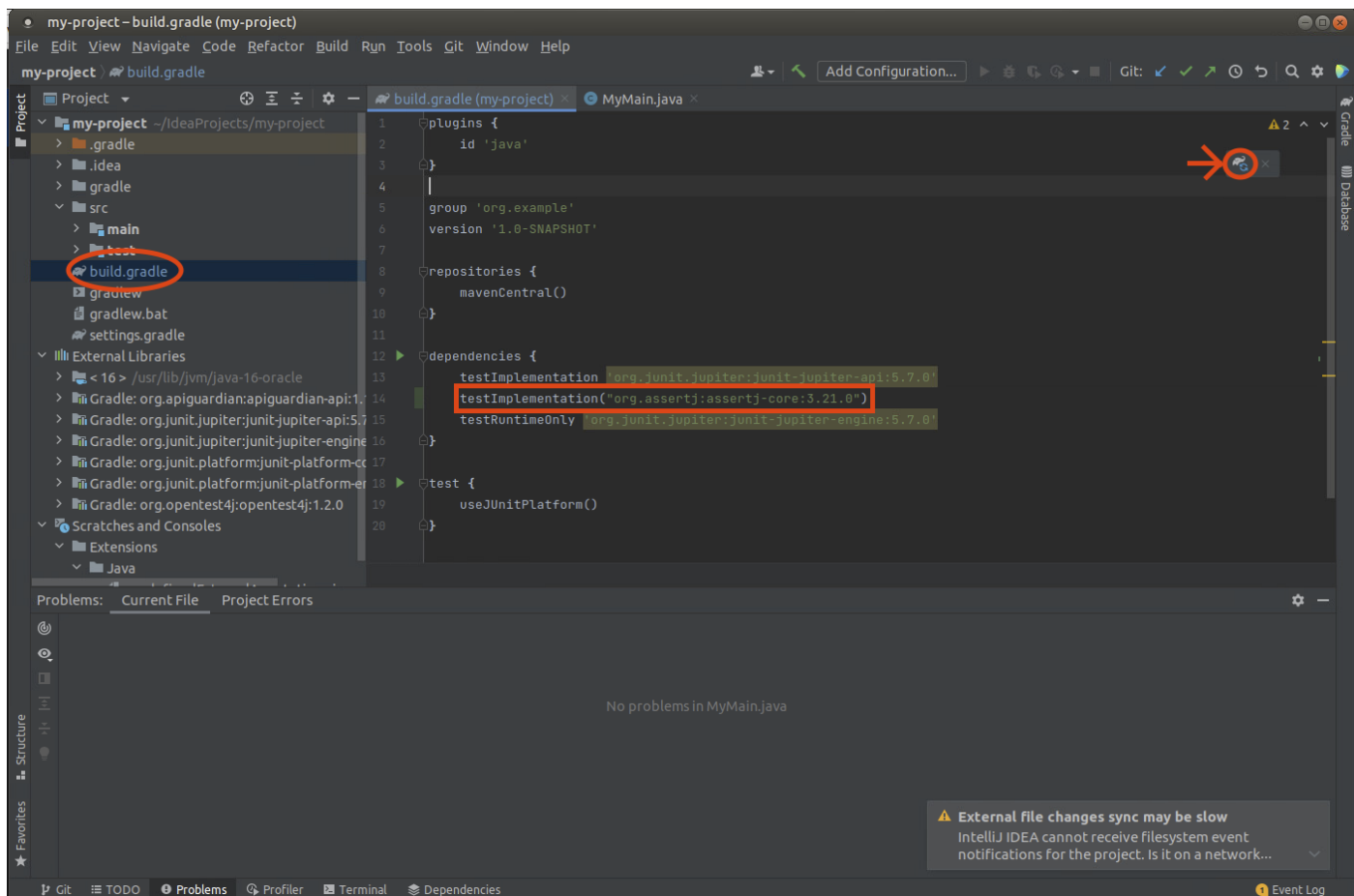
- Copiez l'adresse de votre dépôt etulab dans le champs `URL` et validez l'ajout en cliquant sur le bouton `OK` deux fois (un fois pour la fenêtre `Define remote` et une autre fois pour la fenêtre `Git remotes`).



- Vous pouvez maintenant faire le premier **push** de votre projet en cliquant par exemple dans le bouton dédié en haut à droite puis en validant en cliquant sur le bouton **push**.



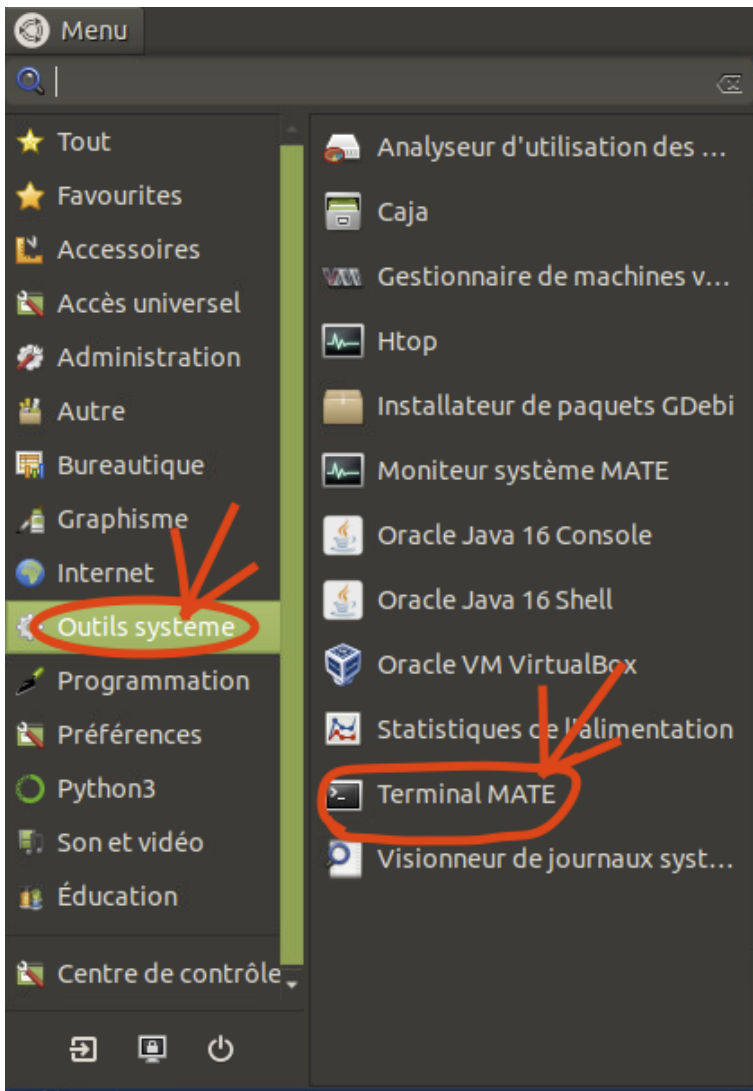
- Vous pouvez ajouter des dépendances (bibliothèques utilisées dans votre projet) en modifiant le fichier `build.gradle`. Vous pouvez par exemple ajouter la ligne `testImplementation("org.assertj:assertj-core:3.21.0")` dans les dépendances puis cliquez sur l'icône de mise à jour afin de rajouter la bibliothèque AssertJ qui permet la rédaction plus claire des tests unitaires.



1.1 Création de projet versionné en ligne de commande

1.1.1 Création de projet gradle en ligne de commande

- La première étape de la création du projet est de créer un projet gradle. Gradle est ce que l'on appelle un moteur de production. C'est un outil pour compiler et exécuter les projets. Il est particulièrement utile pour gérer les dépendance de bibliothèques. En fait, cet outil télécharge automatiquement les bibliothèques configurées dans le projet dans le fichier `build.gradle`. Malheureusement, la version installée de `gradle` n'est pas compatible avec la version de java installée. Vous allez donc devoir installer en local la dernière version de `gradle`. Pour cela télécharger la dernière version de gradle sur le page releases de gradle.
- Ouvrez un terminal en allant dans le menu : Outils système -> Terminal MATE.



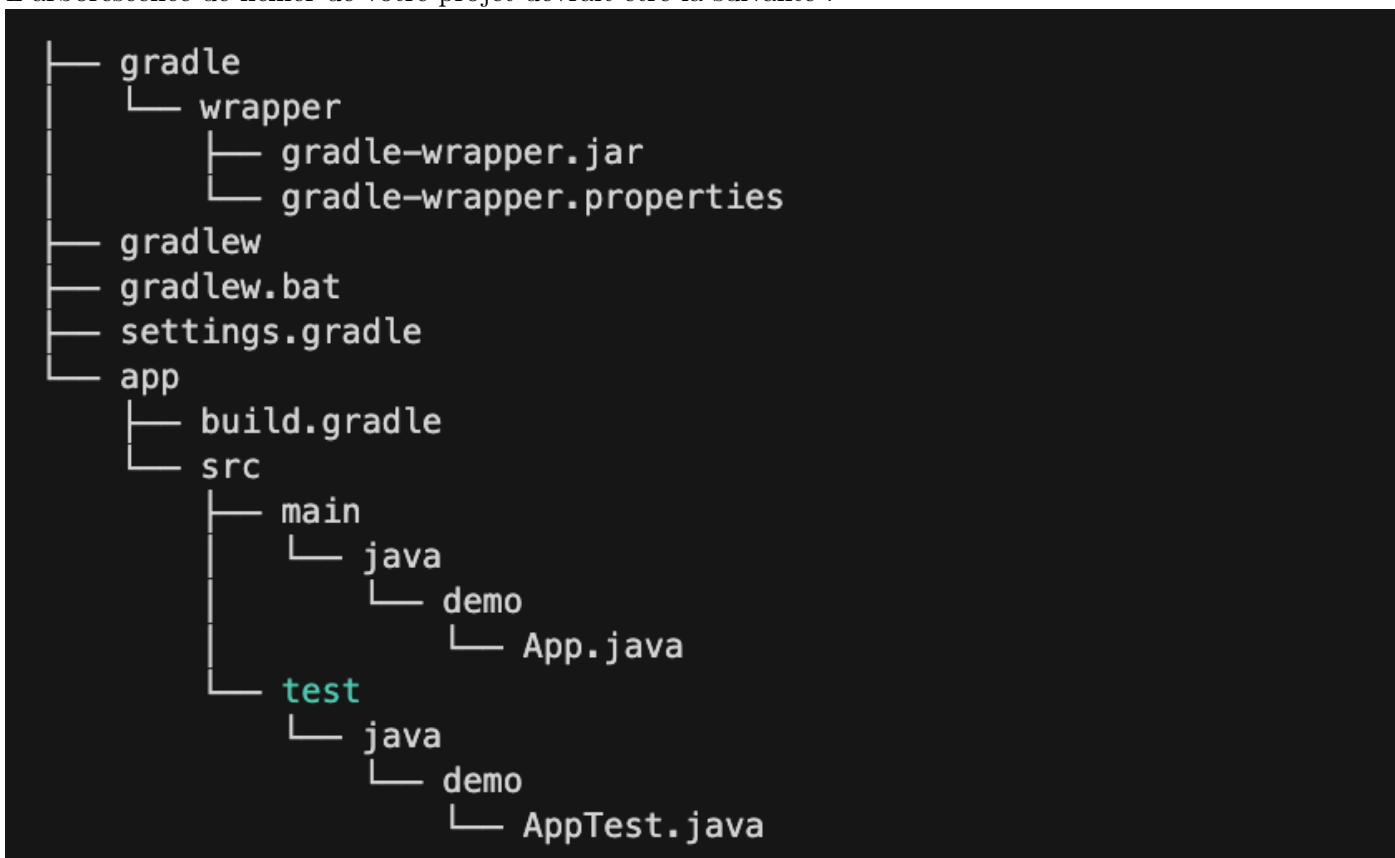
- Créer un répertoire `~/opt/gradle` qui va contenir la dernière version de `gradle` à l'aide des deux commandes suivantes :

```
mkdir ~/opt
mkdir ~/opt/gradle
```
- Dézippez les exécutables de gradle dans le répertoire que vous avez créé à l'aide de la commande suivante :

```
unzip -d ~/opt/gradle ~/Téléchargements/gradle-7.2-bin.zip
```
- Ajoutez gradle à vos chemins d'exécutables dans votre configuration `bash` à l'aide de la commande suivante :

```
echo "export PATH=/opt/gradle/gradle-7.2/bin:$PATH" >> ~/.bashrc
```
- Mettez à jour les chemins d'exécutables pour votre terminal grace à la commande `source ~/.bashrc`. Si tout est bien configuré tout nouveau terminal utilisera la nouvelle version de `gradle`.
- Vérifiez que vous avez la bonne version de gradle avec la commande `gradle --version`
- Créez le répertoire qui va contenir votre projet à l'aide de la commande `mkdir my-project` et placez-vous dedans à l'aide de la commande `cd my-project` (vous pouvez remplacer `my-project` par n'importe quel autre nom).
- Pour commencer la création de votre projet lancez la commande `gradle init`.

- On vous demande tout d'abord quel type de projet vous souhaitez créer. Tapez 2 puis **entrée** pour choisir application.
- On vous demande le langage de projet. Tapez 2 puis **entrée** pour choisir Java.
- On vous demande si votre projet va contenir des fonctionnalités partagées entre plusieurs sous-projets. Tapez 1 puis **entrée** pour choisir No.
- On vous demande si vous préférez utiliser Groovy ou Kotlin. Tapez 2 puis **entrée** pour choisir Groovy.
- On vous demande quel framework de tests vous souhaitez utiliser. Tapez 4 puis **entrée** pour choisir Junit Jupiter.
- On vous demande le nom de votre projet. Vous pouvez laisser le nom par défaut ou bien définir votre propre nom de projet.
- On vous demande le **package** par défaut de votre projet. Vous pouvez laisser le **package** par défaut ou bien définir votre propre **package**. Pour le reste de ce tutoriel on supposera que le nom du package est `demo`.
- L'arborescence de fichier de votre projet devrait être la suivante :

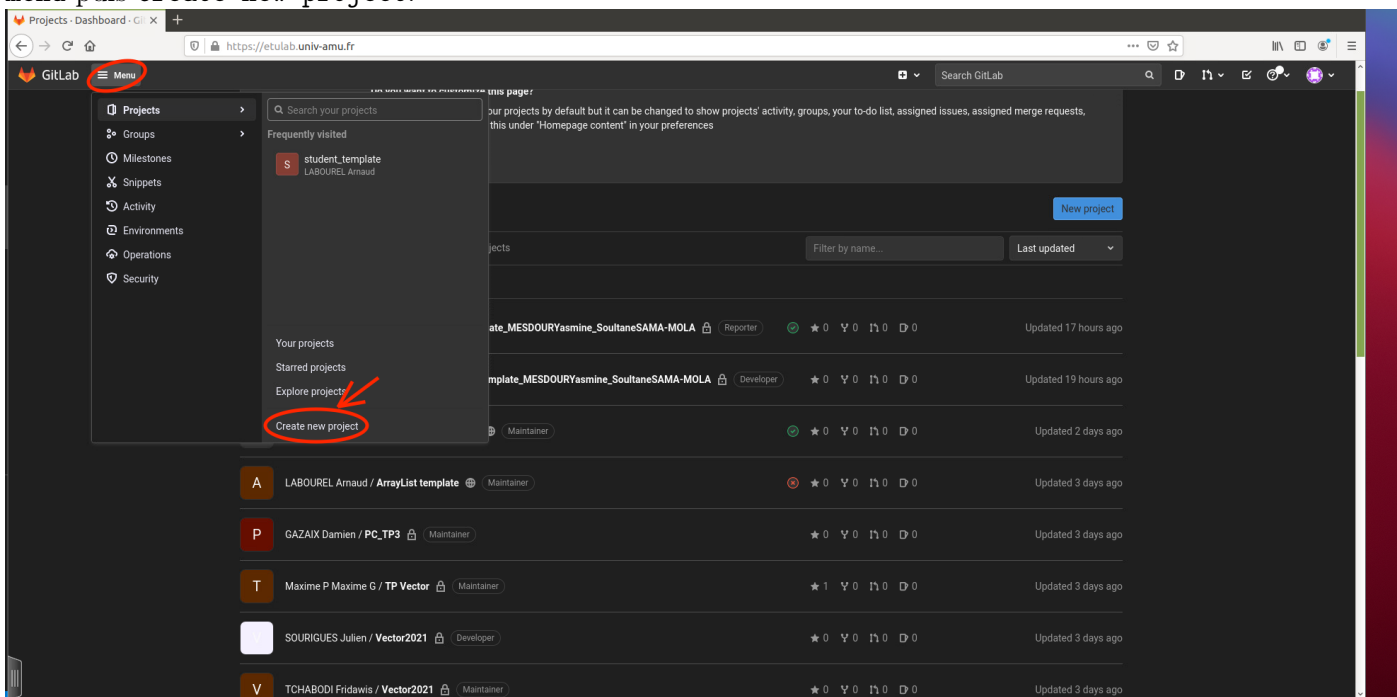


- Le fichier `App.java` contient la méthode `main` qui est exécuté lorsque vous taper la commande `gradle run`.
- Le fichier `AppTest.java` contient des tests qui sont exécutés lorsque vous taper la commande `gradle test`. Si les tests échouent la commande vous donne un lien à ouvrir avec un navigateur.

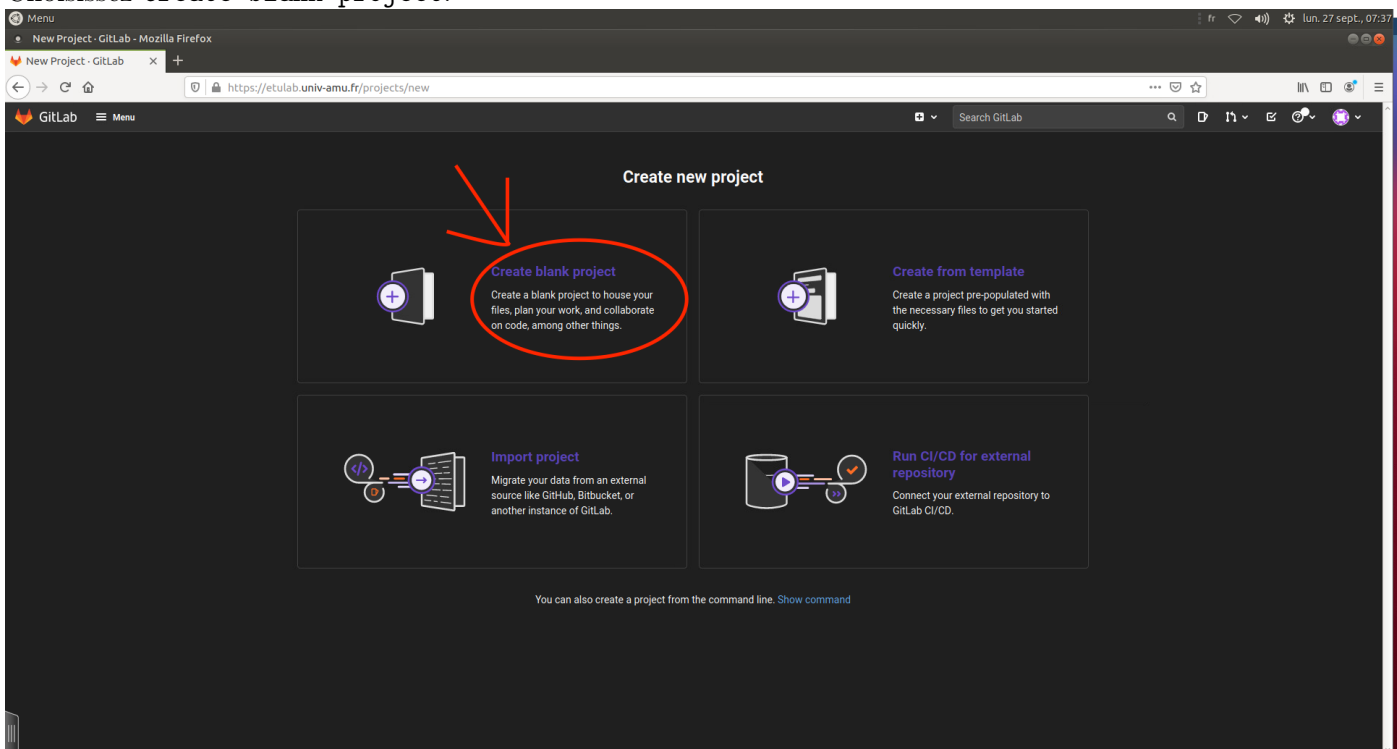
1.1.2 Création de dépôt git en ligne de commande

- Dans le répertoire contenant votre projet `gradle`, exécutez la commande `git init --initial-branch=main` pour créer un dépôt git local.
- Dans le répertoire contenant votre projet `gradle`, exécutez la commande `git add *` pour ajouter tous les fichiers de votre projet à votre prochain `commit`.

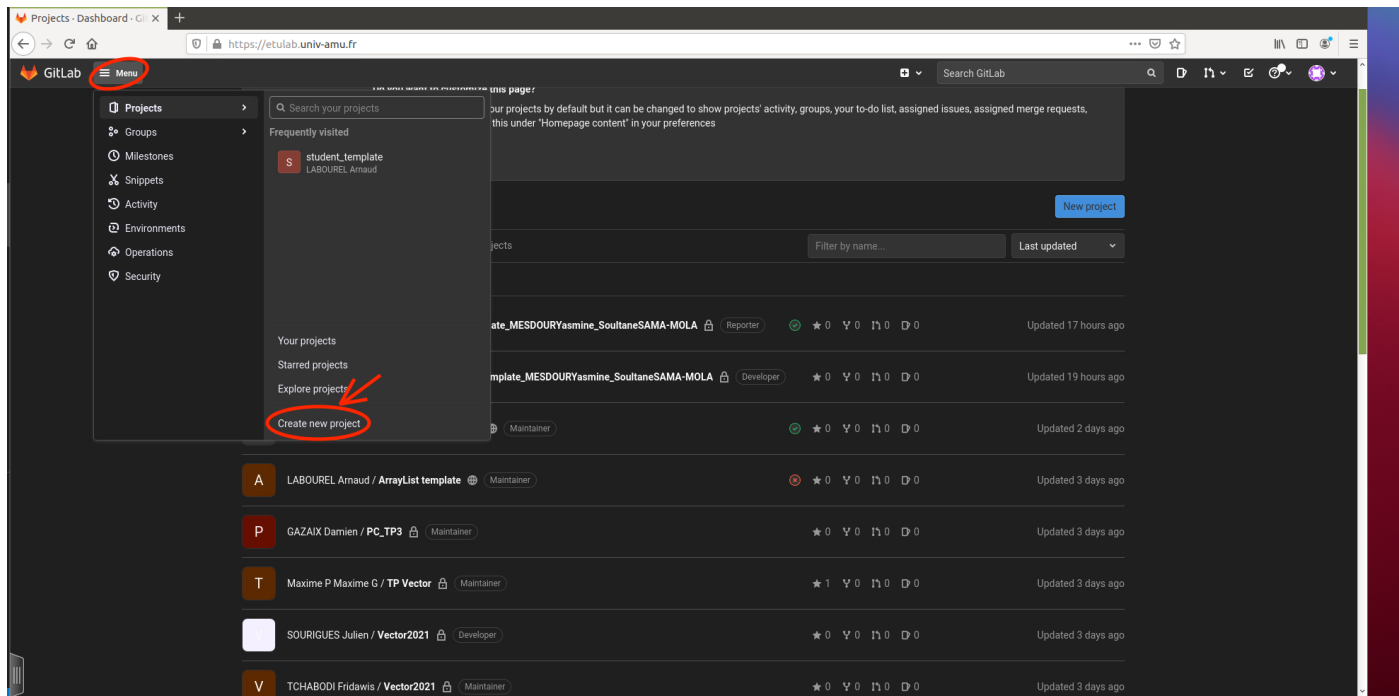
- Exécutez la commande `git commit -m"premier message"` pour faire un premier commit de votre projet avec votre propre message.
- Connectez vous via à un navigateur à votre compte etulab et créez un nouveau projet en cliquant sur **menu** puis **Create new project**.



- Choisissez **Create blank project**.



- Choisissez un nom pour votre projet, décochez la création du `README.md` et validez la création en cliquant sur le bouton **Create project**.



- Copiez l'adresse pour cloner votre projet en cliquant sur le bouton **clone** puis sur l'icône de copie **ssh**.
- Utiliser la commande suivante pour rajouter l'adresse du dépôt distant à votre dépôt local. Il vous faudra bien évidemment remplacer le dernier argument de la commande par l'adresse de votre projet que vous avez préalablement copiée.

```
git remote add origin git@git@etulab.univ-amu.fr:votre_login/le-nom-de-votre_projet.git
```

- Faites le premier **push** à l'aide de la commande `git push -u origin master`. Pour les **push** suivants, vous pourrez simplement utiliser la commande `git push` car vous avez configuré les branches.