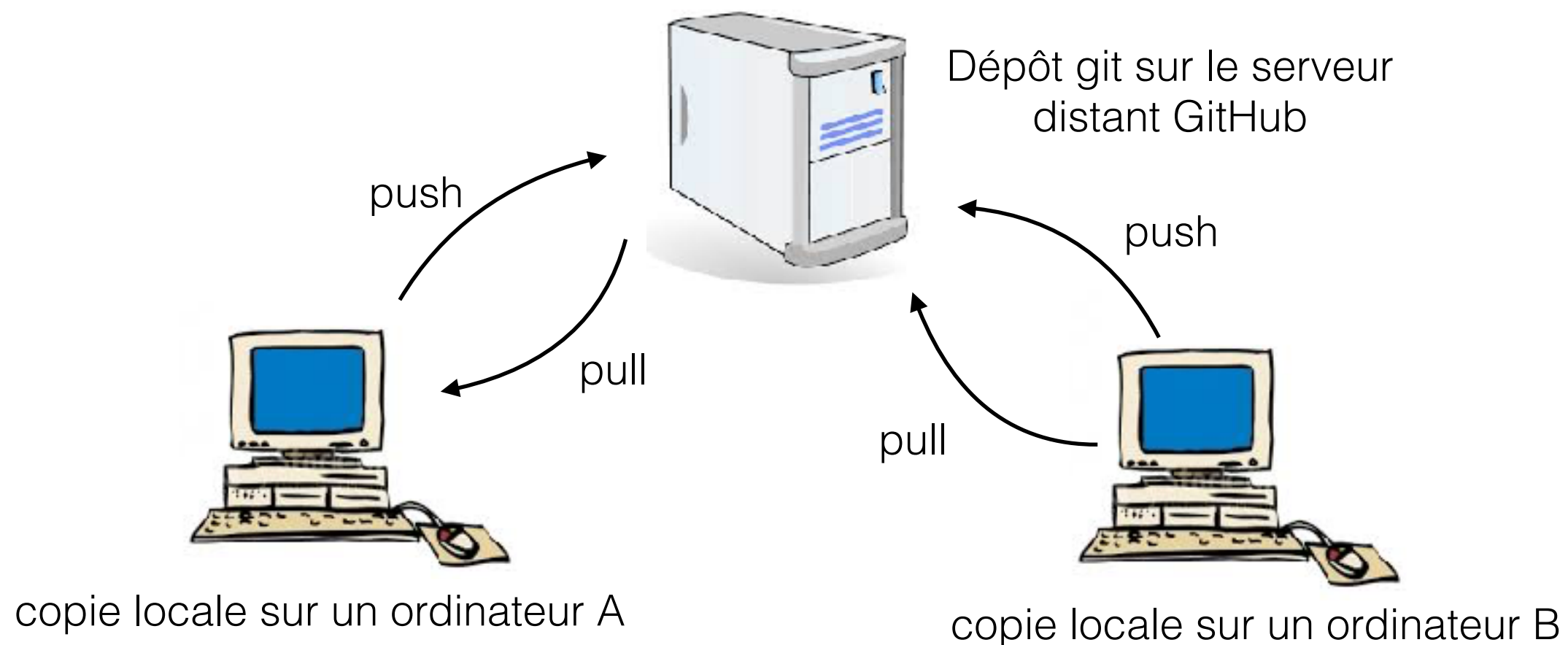




Quelques bases

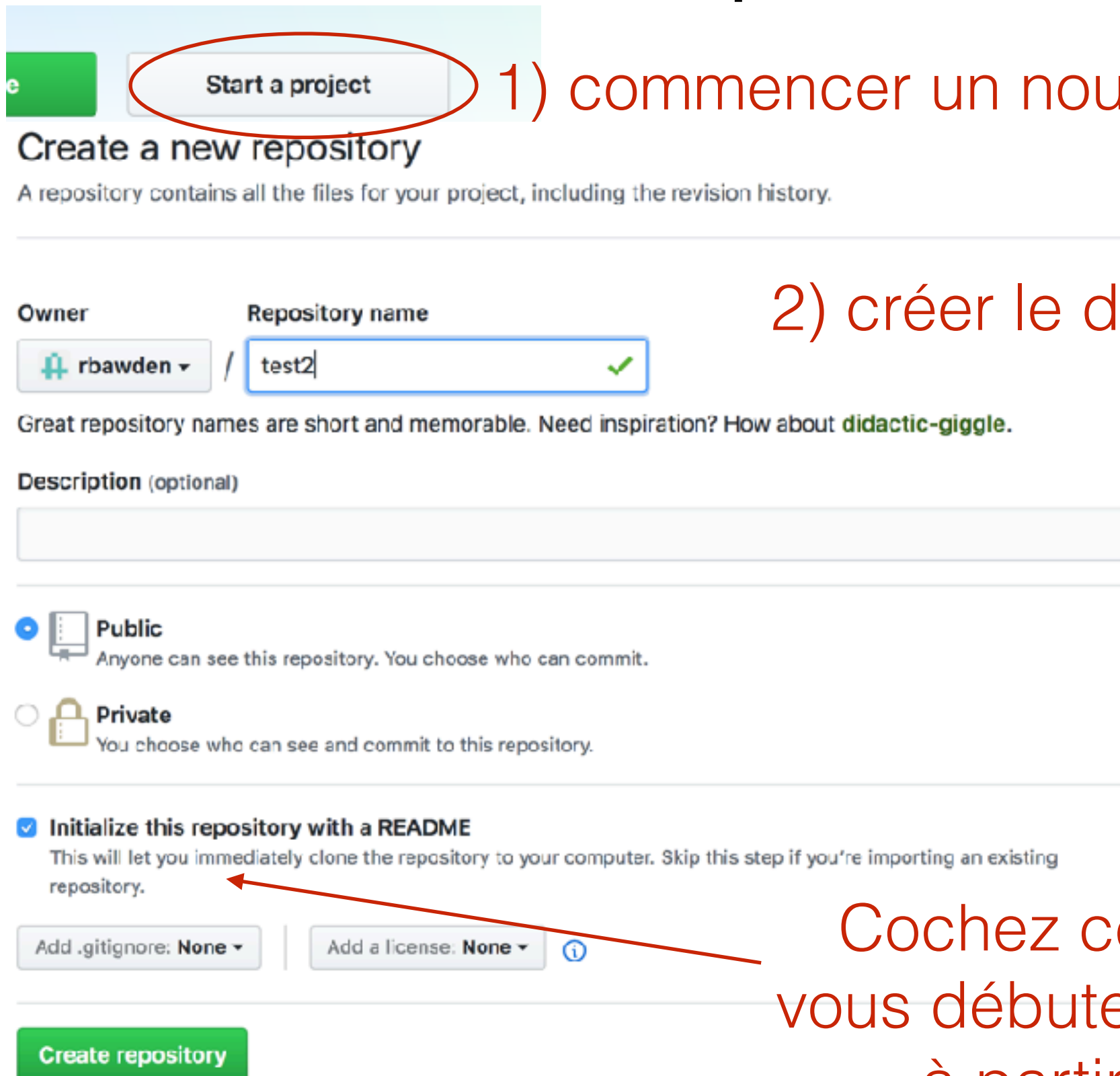
Comment fonctionne Git ?

- Un outil de contrôle de versions, qui permet de travailler facilement à plusieurs sur un même projet
- Une copie à distance du projet est stockée sur le serveur GitHub (le “dépôt”)
- Une copie locale est stockée sur l'ordinateur de chaque collaborateur



- Les modifications faites localement sont transférées vers le serveur distant avec la commande **“git push”**
- Votre copie locale peut être mise à jour pour intégrer les dernières modifications envoyées sur le serveur par les autres collaborateurs avec la commande **“git pull”**

Créer un dépôt Git



The screenshot shows the GitHub 'Create a new repository' form. A red circle highlights the 'Start a project' button. A red arrow points from the text 'Cochez cette case si vous débutez votre projet à partir de zéro' to the 'Initialize this repository with a README' checkbox, which is checked.

Start a project

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: rbawden / Repository name: test2 ✓

Great repository names are short and memorable. Need inspiration? How about **didactic-giggle**.

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

Create repository

1) commencer un nouveau projet

2) créer le dépôt

Cochez cette case si vous débutez votre projet à partir de zéro

rbawden / test

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description, website, or topics provided. [Add topics](#) [Edit](#)

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

rbawden Initial commit

README.md Initial commit

README.md

test

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

<https://github.com/rbawden/test.git>

Open in Desktop Download ZIP

Clonez le dépôt vers votre ordinateur (pour faire une copie locale):
`git clone https://github.com/rbawden/test.git`
dans un terminal à l'endroit où vous voulez cloner le répertoire

Ajouter des fichiers et faire des modifications

- Ajouter un fichier
git add monfichier
- Ajouter plusieurs fichiers
git add monfichier1 monfichier2
- Vérifier le statut du répertoire
git status
- Supprimer un fichier (localement **et** dans le répertoire git ! Attention !)
git rm monfichier
- Supprimer un fichier du répertoire git mais conserver la copie locale
git rm --cached monfichier

Soumettre des modifications et les envoyer sur le serveur

- Soumettre des modifications (les fichiers qui ont été modifiés et ajoutés)
git commit -m "votre message"
- Envoyer les modifications au serveur
git push

Mettre à jour la copie locale

- Mettre à jour votre copie locale avec les modifications envoyées par d'autres collaborateurs sur le serveur :
git pull
- A faire très souvent pour éviter des conflits de versions !
- A faire avant de faire une séquence commit / push

Conflits entre versions

- Des modifications conflictuelles peuvent apparaître, dès lors que deux collaborateurs modifient les mêmes parties du code en parallèle : une fois que l'un des collaborateurs a envoyé ses modifications sur le serveur (commit / push), git ne peut pas fusionner ces modifications avec celle de l'autre collaborateur s'il effectue un pull
- Ceci sera indiqué au moment de faire un pull. Suivez les instructions pour résoudre les conflits. La plupart du temps il est plus facile de résoudre les conflits manuellement...

Résolution des conflits

- Les conflits sont marqués comme suit:

```
<<<<<< HEAD:mergetest
Personne A a modifié cette ligne
=====
Personne B a aussi modifié cette ligne
>>>>>> 4e2b407f501b68f8588aa645acafffa0224b9b78:mergetest
```

- Choisissez la partie à garder, en ne gardant que la bonne version ou en fusionnant les deux manuellement
- Éviter des conflits:
 - * en séparant votre code/rapport en plusieurs fichiers et en partageant bien le travail
 - * en parlant entre vous
 - * en faisant très souvent des pull, des commit et des push

Plus de documentation :

<https://doc.ubuntu-fr.org/git>

<https://openclassrooms.com/courses/gerez-vos-codes-source-avec-git>