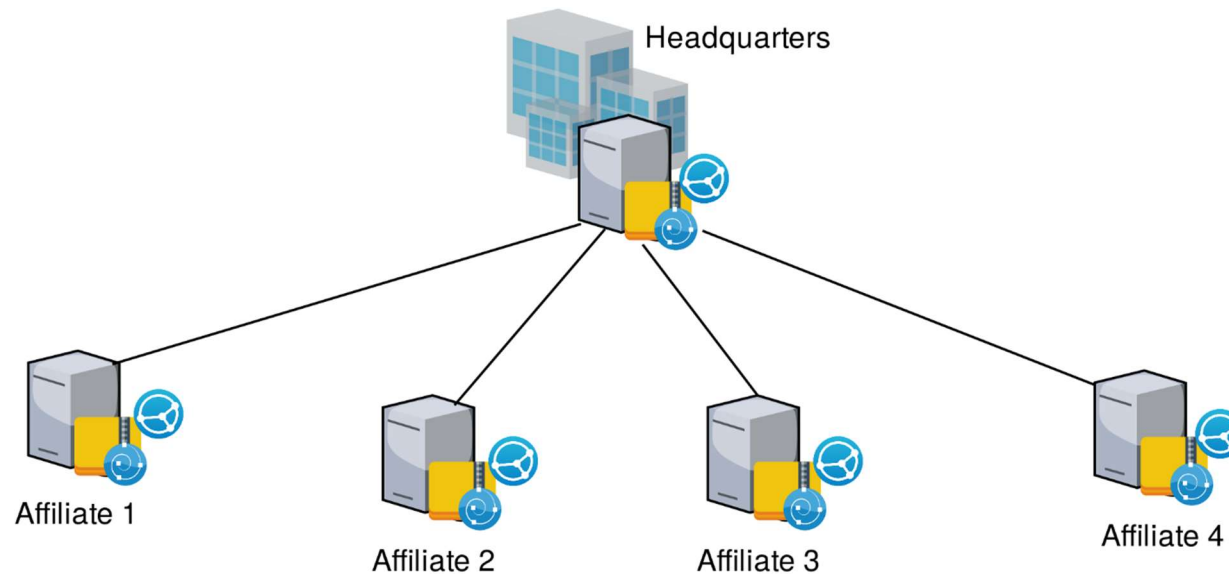


Chapitre 4 – Utiliser un dépôt distant

• A- Afficher les dépôts distants

Pour pouvoir collaborer sur un projet Git, il est nécessaire de savoir comment gérer les dépôts distants. Les dépôts distants sont des versions de votre projet qui sont hébergées sur Internet ou le réseau d'entreprise. Comme c'est le cas avec notre projet de test.



- **git remote**

Pour visualiser les serveurs distants que vous avez enregistrés, vous pouvez lancer la commande `git remote`.

Si vous avez cloné un dépôt, vous devriez au moins voir l'origine `origin` — c'est-à-dire le nom par défaut que Git donne au serveur à partir duquel vous avez cloné.

```
$ git remote  
origin
```

Vous pouvez aussi spécifier `-v`, qui vous montre l'URL que Git a stockée pour chaque nom court :

```
$ git remote -v  
origin https://gitlab.com/micheligestpro/coursgit.git (fetch)  
origin https://gitlab.com/micheligestpro/coursgit.git (push)
```

Notez que ces dépôts distants sont accessibles au moyen de différents protocoles (`ssh`, `https`)

- **B- Ajouter des dépôts distants**

- **git remote add**

Pour ajouter un nouveau dépôt distant Git comme nom court auquel il est facile de faire référence, lancez `git remote add toto [url]` :

```
$ git remote
origin
$ git remote add pb https://github.com/paulboone/ticgit
$ git remote -v
origin      https://github.com/schacon/ticgit (fetch)
origin      https://github.com/schacon/ticgit (push)
pb          https://github.com/paulboone/ticgit (fetch)
pb          https://github.com/paulboone/ticgit (push)
```

Maintenant, vous pouvez utiliser le mot-clé `toto` sur la ligne de commande au lieu de l'URL complète. Par exemple, si vous voulez récupérer toute l'information que Paul a mais que vous ne souhaitez pas l'avoir encore dans votre branche, vous pouvez lancer `git fetch toto` :

```
$ git fetch toto
From https://github.com/paulboone/ticgit
* [new branch]      master      -> toto/master
* [new branch]      ticgit       -> toto/ticgit
```

Changez de branche :

```
$ git checkout toto/master  
Note: checking out 'toto/master'.
```

Ouvrez votre navigateur et vous pourrez constater que ce projet, qui n'a rien à voir avec le notre, est quand même sur votre disque dur, en local.

Quel intérêt ?



- Afficher le dépôt distant <https://gitlab.com/micheligestpro/coursgit2.git>
- En le nommant coursGitDeToto
- Affichez le contenu du fichier readme

- **C- Récupérer et tirer depuis des dépôts distants**

- **git fetch**

Comme nous venons tout juste de le voir, pour obtenir les données des dépôts distants, vous pouvez lancer :

```
$ git fetch [remote-name]
```

Cette commande s'adresse au dépôt distant et récupère toutes les données de ce projet que vous ne possédez pas déjà.

Vous devez volontairement fusionner ses modifications distantes dans votre travail lorsque vous le souhaitez.

- **D- Pousser son travail sur un dépôt distant**

- **git push**

Lorsque votre dépôt vous semble prêt à être partagé, il faut le pousser en amont, sur le repository distant. La commande pour le faire est simple : `git push [nom-distant] [nom-de-branche]`.

Si vous souhaitez pousser votre branche `devPrenom` vers le serveur origin (pour rappel, cloner un dépôt définit automatiquement ces noms pour vous), alors vous pouvez lancer ceci pour pousser votre travail vers le serveur amont :

```
$ git push
fatal: The current branch devYan has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin devYan
```

Seulement, on peut voir que nous avons une erreur. Pourquoi ? Tout simplement parce que nous avons créé notre branche en LOCAL et qu'elle n'existe pas sur le serveur DISTANT.

Mais git nous indique comment procéder :

```
$ git push --set-upstream origin devYan
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 447 bytes | 447.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote:
remote: To create a merge request for devYan, visit:
remote:
https://gitlab.com/micheligestpro/coursgit/merge_requests/new?merge_request%5Bsource_branch%5D=devYan
remote:
To https://gitlab.com/micheligestpro/coursgit.git
 * [new branch]      devYan -> devYan
Branch 'devYan' set up to track remote branch 'devYan' from 'origin'.
```

Et voilà : ma branche locale et ma branche distante, qui vient d'être créée sont à jour et synchronisée !

- **E- Inspecter un dépôt distant**

- **git remote show origin**

Si vous souhaitez visualiser plus d'informations à propos d'un dépôt distant particulier, vous pouvez utiliser la commande `git remote show [nom-distant]`. Si vous lancez cette commande avec un nom court particulier, tel que `origin`, vous obtenez quelque chose comme :

```
* remote origin
Fetch URL: https://gitlab.com/micheligestpro/coursgit.git
Push URL: https://gitlab.com/micheligestpro/coursgit.git
HEAD branch: master
Remote branches:
  devYan tracked
  master tracked
  qualif tracked
Local branches configured for 'git pull':
  devYan merges with remote devYan
  master merges with remote master
  qualif merges with remote qualif
Local refs configured for 'git push':
  devYan pushes to devYan (up to date)
  master pushes to master (up to date)
  qualif pushes to qualif (up to date)
```

Cela donne la liste des URL pour le dépôt distant ainsi que la liste des branches distantes suivies.

- **F- Retirer et renommer des dépôts distants**

- **git remote rename**

Si vous souhaitez renommer une référence, vous pouvez lancer `git remote rename` pour modifier le nom court d'un dépôt distant.

- **Tp - git remote rename**

Par exemple, si vous souhaitez renommer `toto` en `devToto`, vous pouvez le faire avec `git remote rename` :

```
$ git remote rename toto depotToto
```

Ce qui donne :

```
$ git remote  
coursGitDeToto  
depotToto  
origin
```

Il faut mentionner que ceci modifie aussi les noms de branches distantes. Celle qui était référencée sous **toto/master** l'est maintenant sous **depotToto/master**.

- **git remote rm**

Si vous souhaitez retirer un dépôt distant pour certaines raisons — vous avez changé de serveur ou vous n'utilisez plus ce serveur particulier, ou peut-être un contributeur a cessé de contribuer — vous pouvez utiliser git remote rm.

- **TP - git remote rm**

```
$ git remote rm depotToto
```

```
$ git remote  
coursGitDeToto  
origin
```

• Quizz 4 – 60 mn

1-

2-

3-

4-

5-

6-