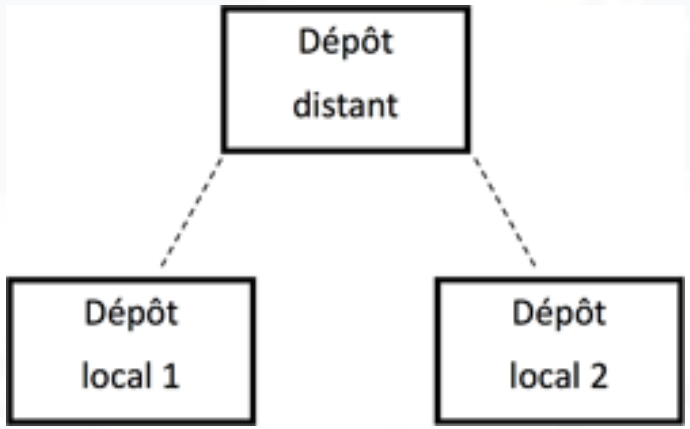

git
Semifir
Adrien Vossough
https://www.semifir.com/

Dépôt distant

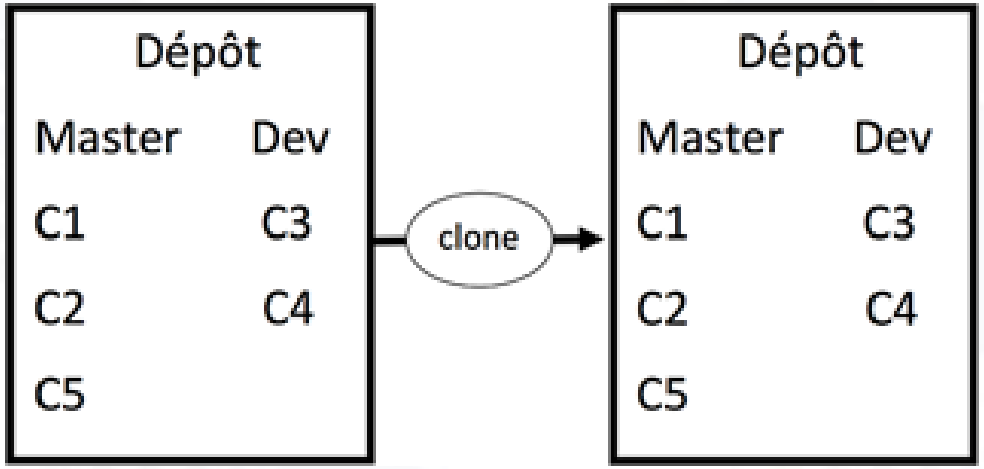
Dépôt distant

Un dépôt distant va centraliser les dépôts locales des développeurs d'un même projet.
Il s’appelle "bare repository" ou "remote".



Cloner un dépôt

Cloner, copier le dépôt distant vers un dépôt local pour y travailler dessus.



Il existe quatre protocoles pour échanger entre deux dépôts :

1. Local : à l'aide du système de fichiers.
2. SSH : à l'aide d'un accès SSH sur la machine du dépôt distant.
3. HTTP : via un serveur HTTP.
4. Git : protocole propre à Git

Cloner un dépôt

Protocoles	Avantages	Inconvénients
Local	Mise en place facile Mêmes restrictions d'accès que celles du système de fichiers.	Mise en place distante, compliqué
SSH	Protocole populaire et sécurisé par une authentification, des restrictions et un chiffrement.	Compétences d'administration système pour la mise en place et la gestion des droits.
HTTP	L'accès est simple via ce protocole universel. Mise en place simple via un serveur HTTP.	Protocole lent. Pas d'authentification ni de restriction. Trafic réseau non chiffré (excepté en HTTPS).
Git	Protocole très rapide	Pas d'authentification ni de restriction. compétences d'administration système. Trafic réseau non chiffré.

Cloner un dépôt

Commandes :

cloner un dépôt via le système de fichiers:

```
git clone ~/Depots/Serveur/CMS.git
```

clonage via SSH :

```
git clone ssh://git.entreprise@serveur_ssh:CMS.git
```

clonage via HTTP :

```
git clone http://site.com/depots/CMS.git
```

clonage via Git :

```
git clone git://site.com/depots/CMS.git
```

Branches distantes

Branches distantes

Liste des dépôts liés :

Un dépôt local peut être lié à plusieurs dépôts distants :

```
# liste des dépôts distants
git remote
```

Avec git clone, un dépôt distant "origin" est placé dans le fichier .git/ :

```
[remote "origin"]
        url = https://github.com/angular/angular.git
        fetch = +refs/heads/*:refs/remotes/origin/*
```

1. La première ligne est le nom du dépôt distant tel qu'il sera utilisé par Git.
2. URL du dépôt distant.
3. endroit où sont stockées les branches locales et les branches distantes suivies.

Branches distantes :

Git utilise des branches cachées liées au serveur distant appelées "**remote tracking branches**"/"branches distantes suivies". Elles servent d'intermédiaire entre les branches locales et distantes.

Lors de l'envoie des commits de master vers le dépôt distant, Git effectue :

1. Les commits de la branche locale sont appliquées à la branche distante, nommée origin/master.
2. Git envoie les nouveaux commits de la branche distante à sa branche locale master.

Envoyer les modifications

Pour partager avec ses collaborateurs sur le dépôt distant "remote nommé origin" les modifications de la branche master :

```
git push origin master
```

Si un autre développeur a envoyé des modifications sur le serveur, il faudra les récupérer avant de pouvoir partager nos données.

Envoyer les modifications

Il existe plusieurs méthodes par défaut pour l'envoi des modifications :

1. **matching** : activée par défaut sur Git 1.x, envoie toutes les branches locales qui correspondent aux branches distantes suivies
2. **simple** : activée par défaut sur Git 2.X. Elle envoie uniquement la branche courante si son nom correspond à celui de la branche distante.
3. **nothing** : n'envoie rien.
4. **current** : envoie la branche courante vers la branche distante suivie de même nom.
5. **upstream** : envoie la branche courante vers sa branche distante suivie, quel que soit le nom des branches.

```
git config --global push.default nom_méthode
```

Lorsque le développeur envoie une branche inconnue du dépôt distant, il faut l'envoyer avec la syntaxe :

```
git push --set-upstream origin nom_branche
```

Recevoir les modifications

mettre à jour la branche courante avec les commits du dépôt distant :

```
git pull
```

git pull est un raccourci de deux commandes :

1. git fetch : télécharge les commits du dépôt distant pour la branche concernée. Ces commits seront intégrés dans la branche distante suivie.

2. git merge FETCH_HEAD : merge les modifications de la branche distante vers la branche locale.

Recevoir les modifications

```
git branch --all

#Cette commande affiche les deux branches suivantes :

* master                    # branche locale
  remotes/origin/master     # branche distante suivie servant d'intermédiaire au dépôt distant
```

git show affiche des informations sur plusieurs types d'objets comme les informations d'un commit : tag, branches, HEAD, etc.

```
git show id_commit
master : ab12
remotes/origin/master : ab12
FETCH_HEAD : ab12
```

Les références remotes/origin/master et FETCH_HEAD sont identiques car elles pointent sur le commit le plus récent de la branche distante suivie.

Branches distantes

Git fetch