# Git : dépôts distants

# Achref El Mouelhi

# Docteur de l'université d'Aix-Marseille Chercheur en programmation par contrainte (IA) Ingénieur en génie logiciel

elmouelhi.achref@gmail.com







- 2 Dépôt distant
- Cloner un dépôt
- ④ GitHub : Fork et Pull-request
- 5 Documentation en français



#### Dépôt nu

- dépôt sans espace de travail (en lecture seule)
- pas de Commit possible
- une sorte de serveur pour tous les utilisateurs

## Se placer dans le parent du dépôt courant

cd ..

© Achref EL MOUELHI ©

イロト イヨト イヨト イヨト

# Se placer dans le parent du dépôt courant

cd ..

Créer un répertoire

mkdir firstGitBare

# Se placer dans le parent du dépôt courant

cd ..

Créer un répertoire

mkdir firstGitBare

Se placer dans ce répertoire

cd firstGitBare

A D M A A A M M

→ ∃ →

# Se placer dans le parent du dépôt courant

cd ..

Créer un répertoire

mkdir firstGitBare

Se placer dans ce répertoire

cd firstGitBare

Créer un dépôt nu

git init --bare

イロト イヨト イヨト イ





イロト イ団ト イヨト イヨト

# Dépôt distant?

# dépôt nu, site hébergeur (GitHub, Bitbucket, GitLab...)...

© Achre

À partir de notre premier dépôt (firstGit), créer un premier dépôt distant

git remote add origin c:/cheminVers/firstGitBare

A (10) A (10) A (10)

À partir de notre premier dépôt (firstGit), créer un premier dépôt distant

git remote add origin c:/cheminVers/firstGitBare

Afficher la liste des dépôts distants OUELH

git remote

À partir de notre premier dépôt (firstGit), créer un premier dépôt distant

git remote add origin c:/cheminVers/firstGitBare

Afficher la liste des dépôts distants OUELH

git remote

Afficher les branches distantes

git branch -r // aucune

イロト イヨト イヨト イヨト

#### Envoyer (publier) la branche master sur le dépôt distant

git push origin master



▲帰▶ ▲ 国▶ ▲ 国≯

Envoyer (publier) la branche master sur le dépôt distant

git push origin master

Afficher les branches distantes

git branch -r

▲帰▶ ▲ 国▶ ▲ 国≯

Envoyer (publier) la branche master sur le dépôt distant

git push origin master



・ 同 ト ・ 三 ト ・

Envoyer (publier) la branche master sur le dépôt distant

git push origin master



#### Supprimer une branche distante

git push origin --delete nomBranche

ヘロン 人間 とくほど くほど



# Vérifier la réception de la branche à partir du firstGitBare

git branch // \*master

< ロ > < 同 > < 回 > < 回 >



# Vérifier la réception de la branche à partir du firstGitBare

git branch // \*master

# Afficher tous les Commit

git log --oneline

< ロ > < 同 > < 回 > < 回 >



#### Pour supprimer un remote



イロト イ団ト イヨト イヨト

git remote remove nomRemote



#### Pour supprimer un remote

git remote remove nomRemote

Pour renommer un remote git remote rename oldName newName

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >



# Se placer dans le parent du dépôt courant



cd ..



# Se placer dans le parent du dépôt courant

cd.

# Cloner le dépôt firstGit dans firstGitClone

#### git clone c:/cheminVers/firstGitBare firstGitClone



#### Se placer dans le répertoire cloné

#### cd firstGitClone

© Achref EL MOUELHIC

イロト イヨト イヨト イヨト

#### Se placer dans le répertoire cloné

cd firstGitClone

Vérifier le dépôt distant

git remote -v

#### Se placer dans le répertoire cloné

cd firstGitClone

Vérifier le dépôt distant

git remote -v

Vérifier les Commit

git log --oneline

< 回 > < 三 > < 三 >

#### Préparer un nouveau Commit

echo bonjour >> new.txt git add . git commit -m "adding new.txt"

э

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

#### Préparer un nouveau Commit

```
echo bonjour >> new.txt
git add .
git commit -m "adding new.txt"
```

Publier le commit dans firstGitBare

git push origin master

# Préparer un nouveau Commit echo bonjour >> new.txt git add . git commit -m "adding new.txt"

#### Publier le commit dans firstGitBare

git push origin master

Vérifier cela depuis firstGitBare

git log --oneline



# Pour récupérer le Commit depuis firstGit





# Pour récupérer le Commit depuis firstGit

```
git pull origin master
```

```
EL MOUL
Vérifier cela depuis firstGitBare
```

```
git log --oneline
```

▲ 同 ▶ → 三 ▶

Préparer un nouveau Commit à partir de firstGitClone

```
echo bonjour >> a.txt
git add .
git commit -m "adding a.txt"
```

イロト イヨト イヨト イヨト

Préparer un nouveau Commit à partir de firstGitClone

```
echo bonjour >> a.txt
git add .
git commit -m "adding a.txt"
```

Publier le commit dans firstGitBare

git push origin master

. . . . . . .

Préparer un nouveau Commit à partir de firstGitClone

```
echo bonjour >> a.txt
git add .
git commit -m "adding a.txt"
```

Publier le commit dans firstGitBare

git push origin master

Vérifier cela depuis firstGitBare

git log --oneline

Refaire la même chose depuis firstGit

```
echo bonjour >> b.txt
git add .
git commit -m "adding b.txt"
```

イロト イヨト イヨト イヨト

Refaire la même chose depuis firstGit

```
echo bonjour >> b.txt
git add .
git commit -m "adding b.txt"
```

Publier le commit dans firstGitBare

git push origin master

. . . . . . .

Refaire la même chose depuis firstGit

```
echo bonjour >> b.txt
git add .
git commit -m "adding b.txt"
```

Publier le commit dans firstGitBare

git push origin master

Impossible car les travaux publiés par firstGitClone n'ont pas été récupérés

A (10) A (10) A (10)



#### Solution

git pull origin master

э

・ロト ・ 四ト ・ ヨト ・ ヨト



#### Solution

git pull origin master

Personnaliser le commit de merge



#### Solution

git pull origin master

MOUELT Personnaliser le commit de merge

Tout publier dans firstGitBare

git push origin master



# Et si on ne veut pas fusionner, on préfère plutôt le rebase

# git pull --rebase origin master



#### Remarques

- On peut également faire un push pour un Commit, un tag...
- On peut forcer le push -f même en cas de contenu divergeant : notre historique remplacera celui du dépôt distant

- ∢ ∃ ▶

#### Exercice

- Créer un nouveau repository GitHub (R1)
- 2 Ajouter un fichier et le commiter (C1)
- Cloner le repository (R2) sur votre ordinateur
- 4 Lister toutes les branches locales et distantes
- Sur R1, modifier le fichier et commiter (C2)
- Sur R2, récupérer C2
- 🕖 Vérifier avec git log
- Sur R2, créer une branche B1
- Iller sur B1, modifier le fichier et commiter (C3)

#### Publier B1 sur R1

#### Suite

- Créer une branche B2 sur R1
- Récupérer B2 sur R2
- 1 Dans R2, aller sur B2
- 1 Tagger B2 sur R2 (T1)
- 1 Publier T1 sur R1
- Vérifier que T1 est sur R1
- Sur R1 B1, modifier la première ligne du fichier et commiter (C4)
- Bur B1 de R2, modifier la première ligne du fichier et commiter (C5)
- Publier C5 sur R1 (conflit)

#### 20 Résoudre le conflit

21 Vérifier la présence d'un commit de merge sur B1 de R1

A B A B A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 B
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A



#### Fork et Pull-request

- Fork : copier un dépôt Git. Il permet de travailler librement sur une copie du projet sans affecter le projet d'origine.
- **Pull-request** : demande au propriétaire d'un projet de relire un code avant le merge.

# Étapes

- Trouvez un projet GitHub auquel vous souhaitez contribuer
- 2 "Forkez"-le
- Olonez-le sur votre ordinateur
- Oréez une nouvelle branche
- Apportez vos modifications
- Renvoyez-le à votre dépôt
- Cliquez sur le bouton Compare & pull request
- Oliquez sur Create pull request pour ouvrir une nouvelle pull request

< ロ > < 同 > < 回 > < 回 >





イロト イ団ト イヨト イヨト

#### Documentation en français

https://git-scm.com/book/fr/v2

© Achre

æ