
Introduction au versioning avec Git

Nicolas Bouvet - Quentin Cabanes • 02.2018

Résumé

Introduction au versioning

Présentation de Git

- Vocabulaire
- Commandes
- Les conflits
- Git flow

Travaux pratiques

- Les hébergeurs
 - Quelques consignes
-

Introduction au versioning

Le principe

- Version
- Historique
- Backup
- Stockage

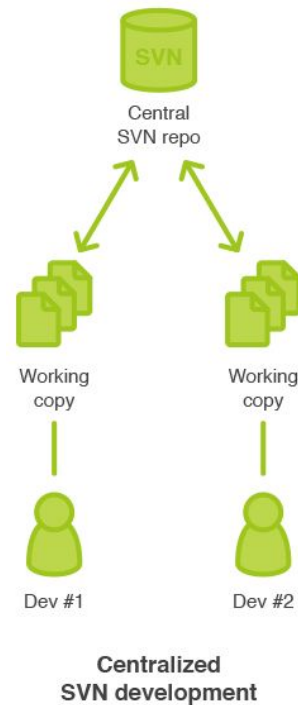
Le travail en équipe

- Collaboration
- Partage

Introduction au versioning

Gestion de versions centralisée

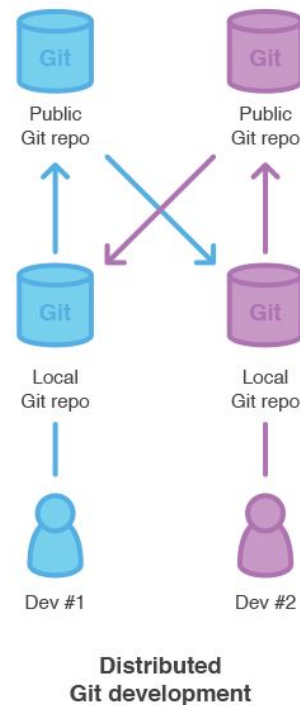
- Un seul espace de stockage
- Simple à mettre en place
- Difficile de travailler sans accès réseau



Introduction au versioning

Gestion de versions distribuée

- Permet de travailler sans accès réseau
- Permet de faire des tests en local
- Pas de mécanisme de verrouillage de fichier



Introduction au versioning

Subversion (SVN)

Avantage

- + Sécurisation de l'envoi de mise à jour

Inconvénient

- Encore des bugs
- Commandes insuffisantes
- Plus lent que Git
- Historique disponible en ligne uniquement

Introduction au versioning

Git

Avantage

- + Commandes très rapides
- + Historique des versions disponible hors-ligne
- + Meilleure résolution des conflits
- + Utilise moins de ressources

Inconvénient

- Plus complexe à apprendre



Présentation de Git



Repository

Espace de stockage



Clone

Action de télécharger un repository



Commit

Mise à jour des fichiers du repository local



Push

Mise à jour du repository selon le repository local



Pull

Téléchargement des changements du repository et fusion des fichiers



Branch

Espace de stockage parallèle à l'espace de stockage principal

Présentation de Git



Fetch

Téléchargement des changements du repository



Tag

Permet de créer un label sur un commit pour mieux identifier l'évolution



Merge

Fusion de branches ou de fichiers



Checkout

Permet de naviguer entre les versions ou branches du repository



Remote

Permet de gérer les connections à d'autres repository

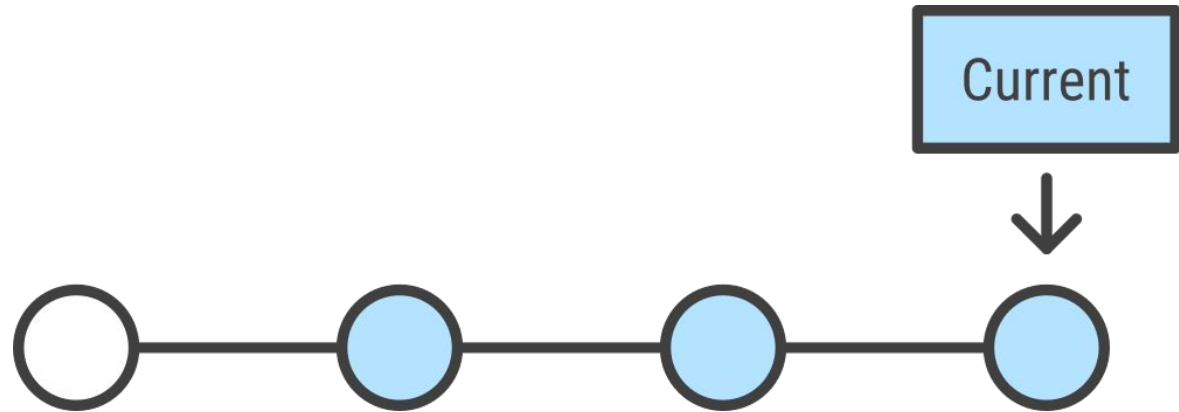


Rebase

Change la branche principale et efface l'ancien historique de commit

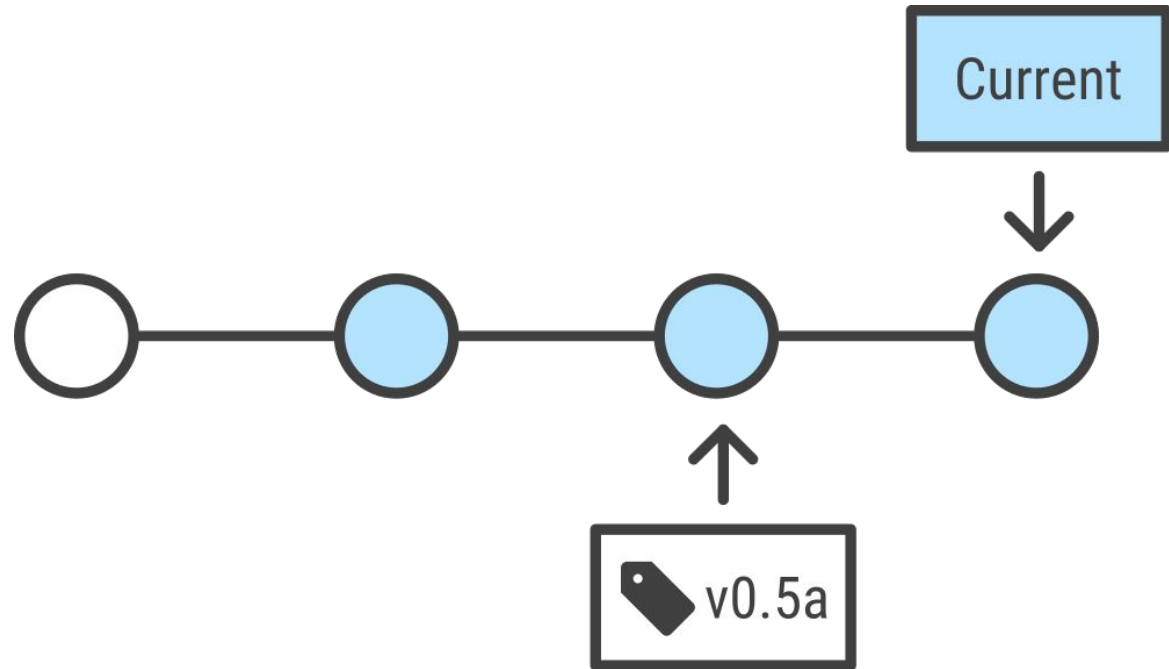
Présentation de Git

Représentation graphique d'un repository



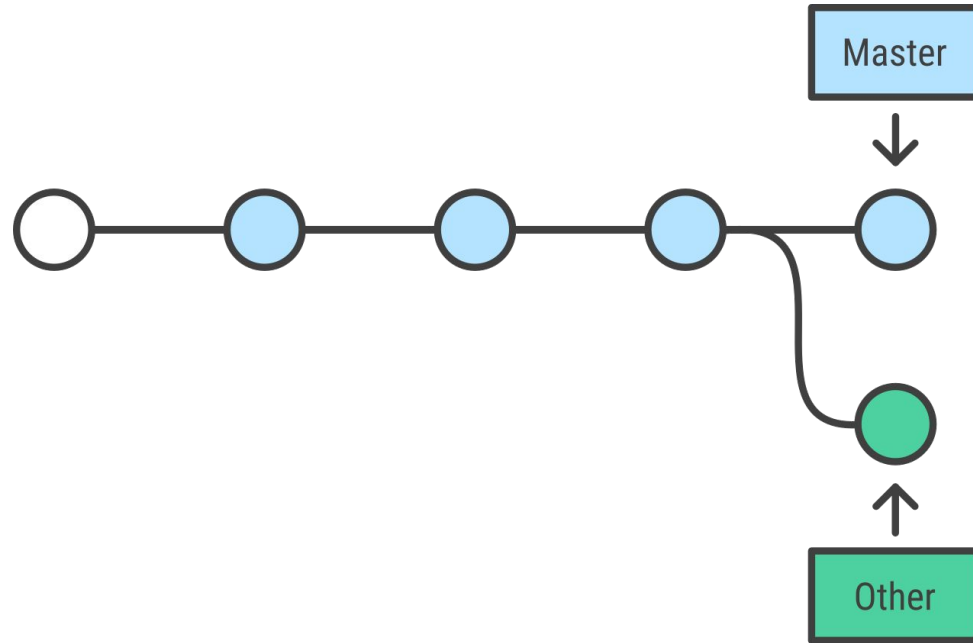
Présentation de Git

Représentation graphique d'un repository



Présentation de Git

Représentation graphique d'un repository

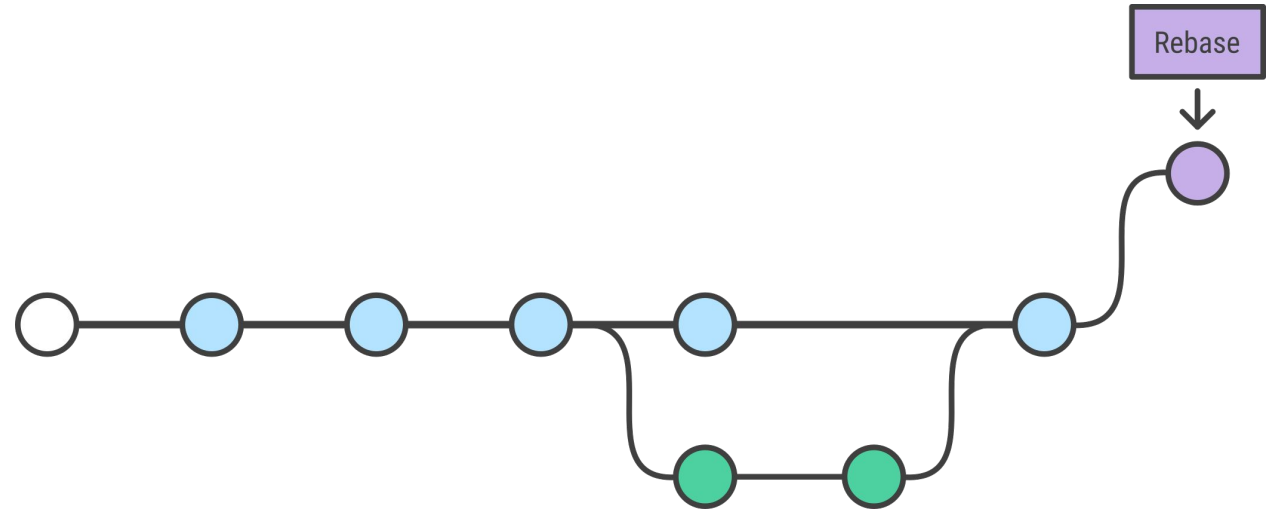


Représentation graphique d'un repository



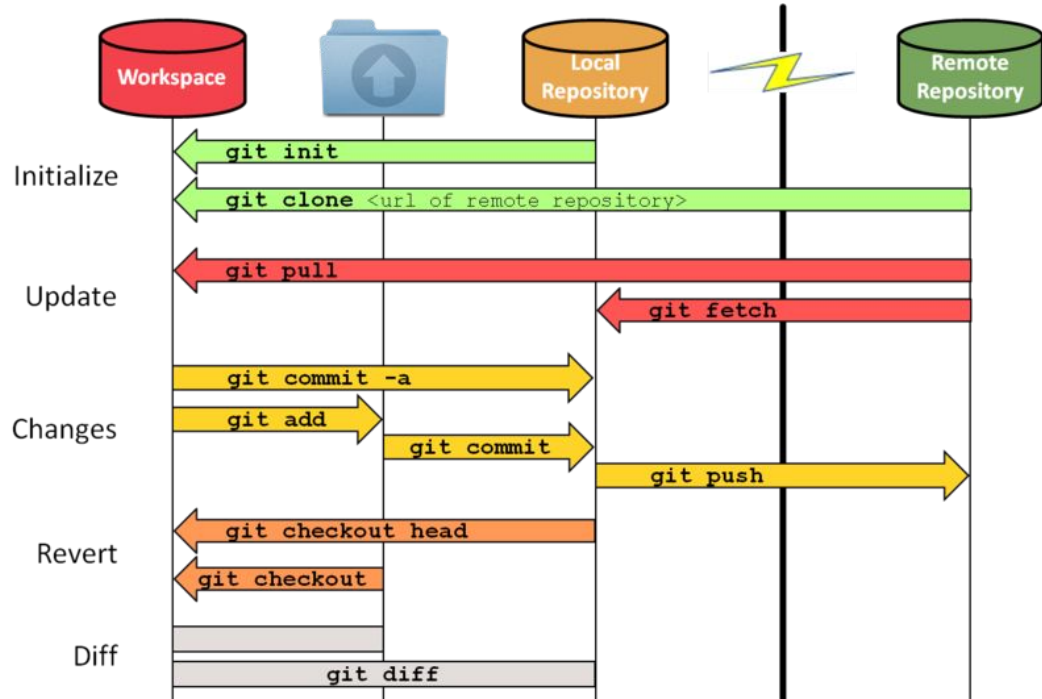
Présentation de Git

Représentation graphique d'un repository



Pause (10 min)

Présentation de Git - Commandes



Présentation de Git - Commandes

Initialisation

```
# Créer un repository git local dans le dossier courant  
$> git init
```

```
# Créer un dossier avec un repository git local dedans  
$> git init <nom_du_dossier>
```

```
# Créer un dossier à partir d'un repository online  
$> git clone <adresse_du_repository> <nom_du_dossier>  
$> git clone http://github.com/ece/projet toto
```

Présentation de Git - Commandes

Mise à jour

```
# Télécharge les changements effectués sur le repository  
$> git pull
```

```
# Equivalent à `git pull`  
$> git fetch  
$> git merge
```

Présentation de Git - Commandes

Effectuer des changements et les envoyer

```
# Déclarer les fichiers au repository local  
$> git add mon_fichier.txt
```

```
# Effectuer le changement dans le repository local  
$> git commit -m "Le message de commit"
```

```
# Envoyer les changements au serveur du repository  
$> git push
```

Présentation de Git - Commandes

Changer de version

```
# Revenir à une ancienne version  
$> git checkout <version>
```

```
# Changer de branche  
$> git checkout <nom_de_la_branche>
```

```
# Ajouter une branche  
$> git checkout -b <nom_de_la_branche>
```

Présentation de Git - Commandes

Observer les changements

```
# Montrer les changements entre les commits  
$> git diff
```

Présentation de Git - Commandes

Exemple

```
# On récupère un repository
$> git clone http://github.com/toto/projet-info-ing2

# On crée un fichier `Cartes.c` et on ajoute le dossier
# `image`. On ajoute au repo et on commit.
$> git add image
$> git commit -m "Ajout images des cartes"
$> git add Cartes.c
$> git commit -m "Création des fonctions de cartes"

# On met à jour le repository distant
$> git push
```

Présentation de Git - Les conflits



Savoir résoudre un conflit Git permet d'éviter les problèmes lorsque plusieurs personnes travaillent sur un même fichier !

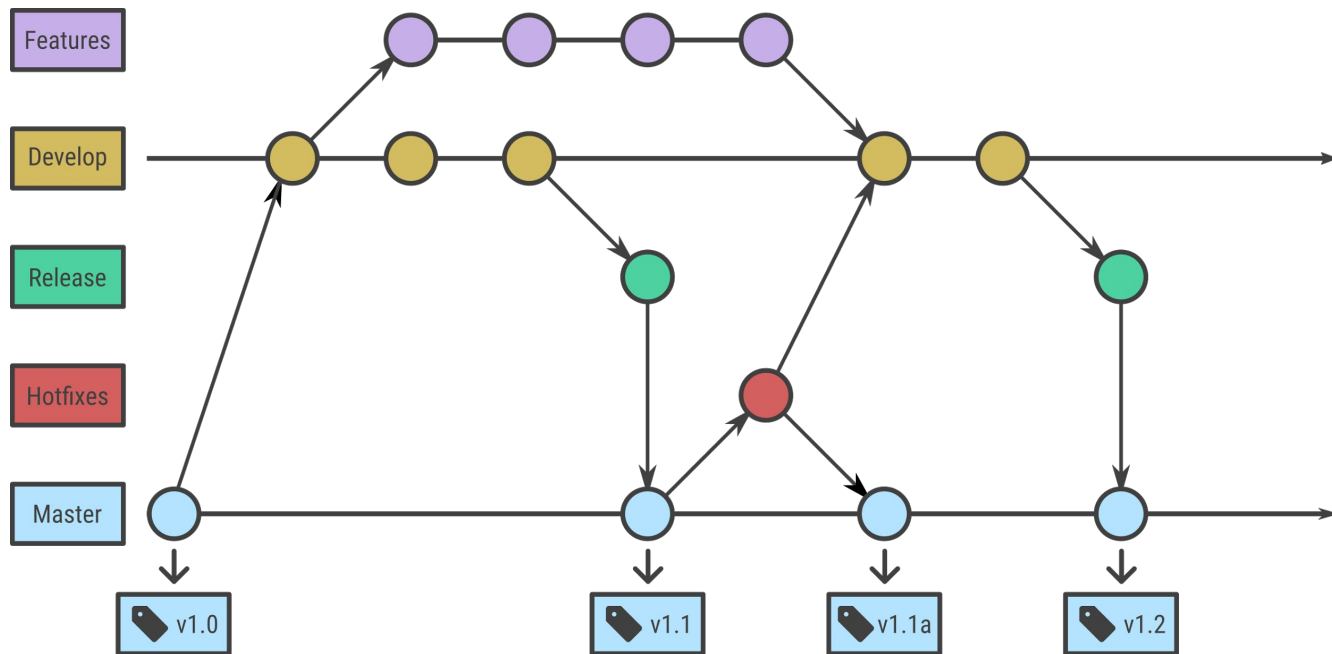
```
<<<<<< HEAD
Here is the original change.
=====
Here is the modified change.
>>>>>> 57018fe1230ad2123c198a12d314cc56
```

Présentation de Git - Les conflits

Résoudre un conflit

1. Résoudre le conflit
2. Commit la résolution
3. Merge le tout
4. Push

Présentation de Git - Git Flow

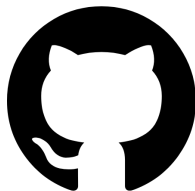


Travaux pratiques

Travaux pratiques - Hebergeurs

GitHub

- Repos publics illimités
- Repos privés payants
(Sauf pour les étudiants)



Bitbucket

- Repos publics illimités
- Repos privés gratuits jusqu'à 5 utilisateurs



Des questions ?

Travaux pratique

1. Télécharger le TP sur Campus
2. Lire le sujet
3. Travailler en équipe de 2 ou 3
