Git-Flow

Un modèle de versionnement efficace avec Git

Philippe Bollard

philippe.bollard@univ-grenoble-alpes.fr

29 juin 2017

CNRS/IPAG/OSUG

Plan

1. Présentation
2. Modèle de branches
3. Cas pratique
4. Bilan

Présentation

Git-Flow, c'est quoi ?

Une extension de Git

- disponible sur GNU/Linux, BSD, MacOS, Windows
- rajoute un ensemble de sous-commandes à Git
- automatise une série de commandes Git pour une action précise

Un ensemble de bonnes pratiques

- standardise l'organisation d'un dépôt Git
- clarifie le nommage des branches
- impose un workflow pour le développement d'un projet
- convient au développement agile (type SCRUM)

Comment l'installer et s'en servir ?

Avec Debian, c'est très simple !

```
sudo apt-get install git-flow
```

Pour les autres... RTFM :)

github.com/petervanderdoes/gitflow-avh/wiki/Installation

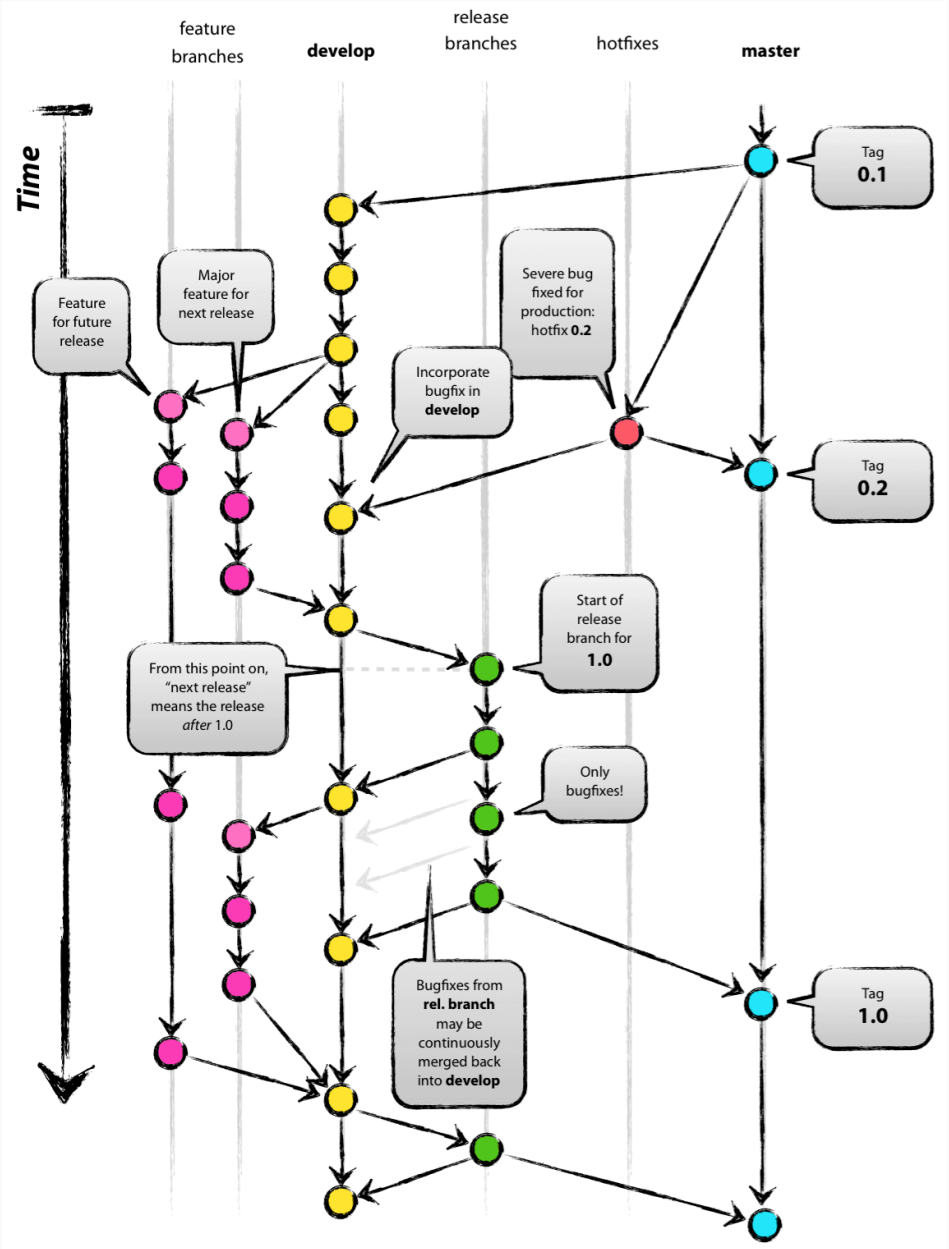
Quelques liens utiles

- danielkummer.github.io/git-flow-cheatsheet/
- github.com/petervanderdoes/gitflow-avh/wiki/
- nvie.com/posts/a-successful-git-branching-model/
- raphaelhertzog.fr/livre/memento-git/

Au fait, la bonne version, c'est Gitflow-AVH...

Modèle de branches

Organisation des branches



Rôle des branches principales

master

- contient la dernière version stable du produit livrable et opérationnel
- ne reçoit aucun commit direct

develop

- devient la branche principale, véritable colonne vertébrale du dépôt
- branche d'intégration des développements
- ne reçoit, si possible, aucun commit direct

feature/mabranche

- branche de développement isolé d'une fonctionnalité
- créée à partir de develop
- sera fusionnée dans develop puis détruite

Rôle des branches secondaires

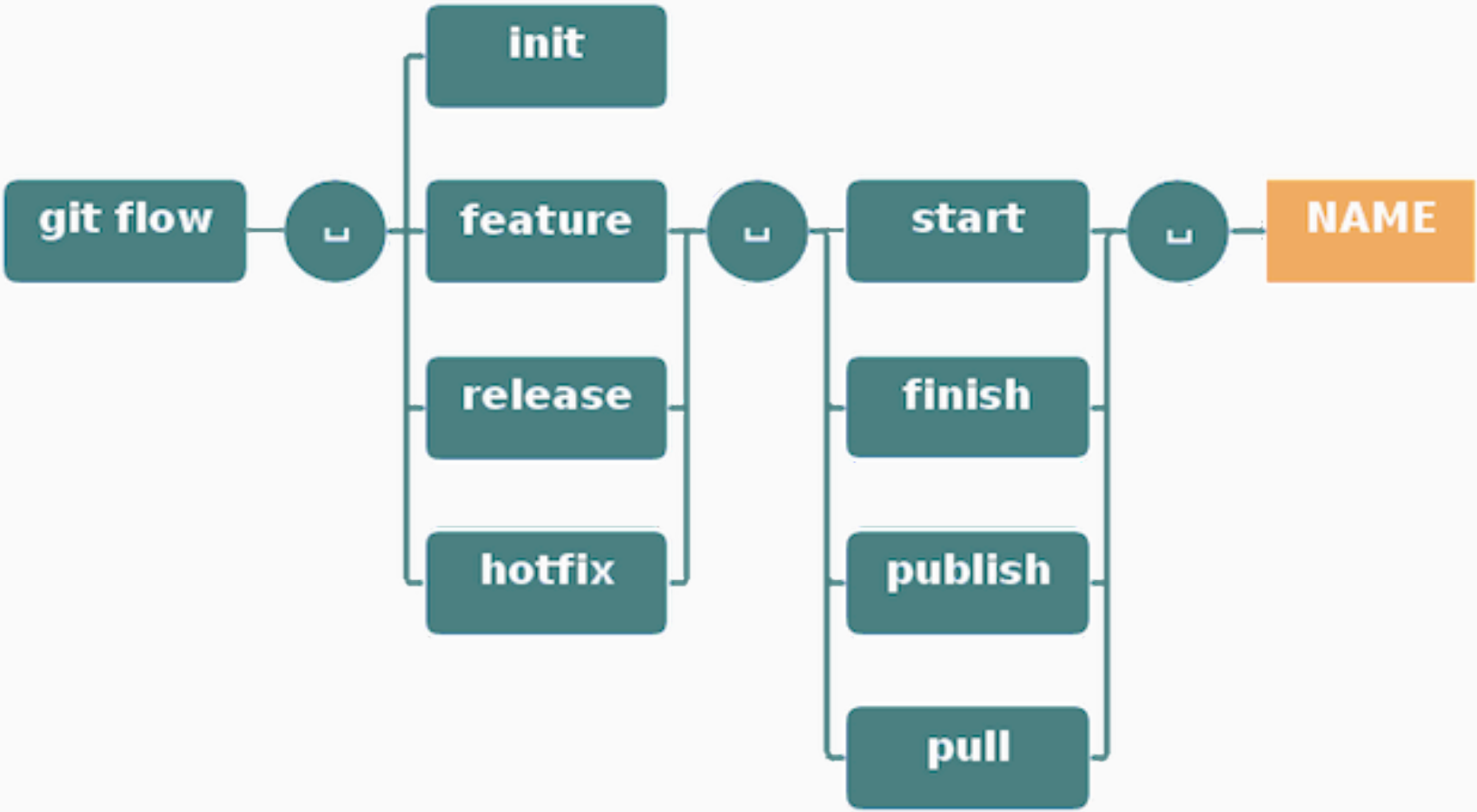
release/mabranche

- branche de préparation à la livraison d'une nouvelle version
- permet de stabiliser une livraison sans bloquer le développement
- créée à partir de develop
- sera fusionnée dans master et dans develop puis détruite
- génère la création d'un tag identifiant la version

hotfix/mabranche

- branche de réalisation d'un correctif
- permet de corriger un bug en prod sans bloquer le développement
- créée à partir de master
- sera fusionnée dans master et dans develop puis détruite
- génère la création d'un tag identifiant la version

Commandes


git flow
␣
init
feature
release
hotfix
␣
start
finish
publish
pull
␣
NAME

Cas pratique

Initialisation d'un dépôt Git

Création d'un dépôt local

```
~/$ mkdir demo
~/$ cd ~/demo
~/demo$ git init
~/demo$ touch README.md
~/demo$ git add README.md
~/demo$ git commit -m "Initial commit"

~/demo$ git remote add origin git@maforge.fr/demo.git
~/demo$ git push -u origin master
```

En clonant un dépôt vide existant

```
~/$ git clone git@maforge.fr/demo.git
~/$ cd demo
~/demo$ touch README.md
~/demo$ git add README.md
~/demo$ git commit -m "add README"
~/demo$ git push -u origin master
```

Initialisation de Git-Flow pour ce dépôt

Initialiser git-flow

```
~/demo$ git flow init
```

Répondre aux questions

```
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] tag/
Hooks and filters directory? [~/demo/.git/hooks]
```

Création d'une fonctionnalité 1

Se placer dans develop (automatique après init)

```
~/demo$ git checkout develop
Basculement sur la branche 'develop'
```

Créer la branche de fonctionnalité 1

```
~/demo$ git flow feature start mafonction1
Basculement sur la nouvelle branche 'feature/mafonction1'

Summary of actions:
- A new branch 'feature/mafonction1' was created, based on 'develop'
- You are now on branch 'feature/mafonction1'

Now, start committing on your feature. When done, use:

     git flow feature finish mafonction1
```

Développement d'une fonctionnalité 1

Pour faire simple, on ajoute un fichier...

```
~/demo$ touch fonction1.txt
~/demo$ git add fonction1.txt
~/demo$ git commit -m "Fonction 1"
[feature/mafonction1 6a44a23] Fonction 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fonction1.txt
```

Création d'une fonctionnalité 2

Se placer dans develop

```
~/demo$ git checkout develop
Basculement sur la branche 'develop'
```

Créer la branche de fonctionnalité 2

```
~/demo$ git flow feature start mafonction2
Basculement sur la nouvelle branche 'feature/mafonction2'

Summary of actions:
- A new branch 'feature/mafonction2' was created, based on 'develop'
- You are now on branch 'feature/mafonction1'

Now, start committing on your feature. When done, use:

     git flow feature finish mafonction2
```

Développement d'une fonctionnalité 2

Pour faire simple, on ajoute un fichier...

```
~/demo$ touch fonction2.txt
~/demo$ git add fonction2.txt
~/demo$ git commit -m "Fonction 2"
[feature/mafonction2 2bdb068] Fonction 2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fonction2.txt
```

Finalisation de la fonctionnalité 1

Se placer dans la branche adéquate

```
~/demo$ git checkout feature/mafonction1
Basculement sur la branche 'feature/mafonction1'
```

Fermer la branche et l'intégrer dans develop

```
~/demo$ git flow feature finish mafonction1
Basculement sur la branche 'develop'
Mise a jour fe78ff5..6a44a23
Fast-forward
 fonction1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fonction1.txt
 Branche feature/mafonction1 supprimee (precedemment f2c2d2b).

 Summary of actions:
 - The feature branch 'feature/mafonction1' was merged into 'develop'
 - Feature branch 'feature/mafonction1' has been locally deleted
 - You are now on branch 'develop'
```

Reprise de la fonctionnalité 2

Se placer dans la branche adéquate

```
~/demo$ git checkout feature/fonction2
Basculement sur la branche 'feature/fonction2'
```

Intégrer les autres développements (facultatif)

```
~/demo$ git merge develop
Merge made by the 'recursive' strategy.
 fonction1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fonction1.txt
```

Partager la fonctionnalité 2 (non-finalisée) avec l'équipe

Se placer dans la branche adéquate

```
~/demo$ git checkout feature/fonction2
Basculement sur la branche 'feature/fonction2'
```

Publier la branche

```
~/demo$ git flow feature publish mafonction2
Decompte des objets: 9, fait.
Delta compression using up to 4 threads.
Compression des objets: 100% (5/5), fait.
Ecriture des objets: 100% (9/9), 792 bytes | 0 bytes/s, fait.
Total 9 (delta 1), reused 0 (delta 0)
To maforge.fr/demo.git
 * [new branch]      feature/mafonction2 -> feature/mafonction2
Deja sur 'feature/mafonction2'
Votre branche est a jour avec 'origin/feature/mafonction2'.

Summary of actions:
- A new remote branch 'feature/mafonction2' was created
- The local branch 'feature/mafonction2' was configured to track the r
- You are now on branch 'feature/mafonction2'
```

Création d'une livraison 1

Se placer dans develop

```
~/demo$ git checkout develop
Basculement sur la branche 'develop'
```

Créer la branche de release 1

```
~/demo$ git flow release start marelease1
Basculement sur la nouvelle branche 'release/marelease1'

Summary of actions:
- A new branch 'release/marelease1' was created, based on 'develop'
- You are now on branch 'release/marelease1'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

     git flow release finish 'marelease1'
```

Fignolage de la release (facultatif)

Ah, on a oublié le fichier de changelog...

```
~/demo$ touch changelog.txt
~/demo$ git add changelog.txt
~/demo$ git commit -m "changelog"
[release/marelease1 cc157fa] changelog
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 changelog.txt
```

Fermeture de la release

```
~/demo$ git flow release finish marelease1
Basculement sur la branche 'master'
Merge made by the 'recursive' strategy.
 changelog.txt | 0
 fonction1.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 changelog.txt
 create mode 100644 fonction1.txt
Basculement sur la branche 'develop'
Merge made by the 'recursive' strategy.
 changelog.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 changelog.txt
Branche release/marelease1 supprimee (precedemment cc157fa).

Summary of actions:
- Release branch 'release/marelease1' has been merged into 'master'
- The release was tagged 'tag/marelease1'
- Release tag 'tag/marelease1' has been back-merged into 'develop'
- Release branch 'release/marelease1' has been locally deleted
- You are now on branch 'develop'
```

Publication des tags (facultatif)

```
~/demo$ git push --tags
Decompte des objets: 4, fait.
Delta compression using up to 4 threads.
Compression des objets: 100% (4/4), fait.
Ecriture des objets: 100% (4/4), 514 bytes | 0 bytes/s, fait.
Total 4 (delta 1), reused 0 (delta 0)
To maforge.fr/demo.git
 * [new tag]         tag/marelease1 -> tag/marelease1
```

Création d'une correction sur release1

Se placer dans master

```
~/demo$ git checkout master
Basculement sur la branche 'master'
```

Créer la branche de hotfix 1.1

```
~/demo$ git flow hotfix start marelease1.1
Basculement sur la nouvelle branche 'hotfix/marelease1.1'

Summary of actions:
- A new branch 'hotfix/marelease1.1' was created, based on 'master'
- You are now on branch 'hotfix/marelease1.1'

Follow-up actions:
- Bump the version number now!
- Start committing your hot fixes
- When done, run:

     git flow hotfix finish 'marelease1.1'
```

Résolution du problème

Ah, on a oublié de remplir le fichier de changelog...

```
~/demo$ echo 'tagada' > changelog.txt
~/demo$ git add changelog.txt
~/demo$ git commit -m "changelog"
[hotfix/marelease1.1 c2fef92] changelog
 1 file changed, 1 insertion(+)
```

Fermeture du hotfix

```
~/demo$ git flow hotfix finish marelease1.1
Basculement sur la branche 'master'
Merge made by the 'recursive' strategy.
 changelog.txt | 1 +
 1 file changed, 1 insertion(+)
Basculement sur la branche 'develop'
Merge made by the 'recursive' strategy.
 changelog.txt | 1 +
 1 file changed, 1 insertion(+)
Branche hotfix/marelease1.1 supprimee (precedemment c2fef92).

Summary of actions:
- Hotfix branch 'hotfix/marelease1.1' has been merged into 'master'
- The hotfix was tagged 'tag/marelease1.1'
- Hotfix tag 'tag/marelease1.1' has been back-merged into 'develop'
- Hotfix branch 'hotfix/marelease1.1' has been locally deleted
- You are now on branch 'develop'
```

Bilan

Retour d'expérience

Positif

- facile à mettre en oeuvre (sur un projet vide)
- apporte une certaine rigueur dans la gestion du projet
- clarifie le nommage des branches et tags
- extension qui ne se substitue pas à la richesse de Git
- favorise l'usage des branches (dont il faut user et abuser avec Git !)

Négatif

- quelques pièges de config si master n'est pas préalablement initialisée
- reconfig nécessaire des dépôts existants (migrer master en develop)
- processus de release un peu lourd pour les cas simples sans fignolage
- n'empêche pas les dérives (commit sauvage dans master ou develop)

Questions ?