

Workshop Wednesdays!

Episode 1: Git



Why Git?

- Keep your code safe
- Collaborate
- Track changes
- Make troubleshooting easier
- Allow easy deployment

What's Git?

- Version Control System
- Authored by Linus Torvalds to manage Linux kernel development
- Version 1.0 released on 2005
- Distributed in nature
- Awesome merge algorithms!

Let's Git Started!

git config

- `git config --global user.name "Benjie Jiao"`
- `git config --global user.email "hi@benjie.me"`

git init

- Initializes an unversioned project to a Git Repository
- Creates the .git directory

(File) .gitignore

- List of files that should NOT be tracked

(File) README.md

- Contains main documentation for the project

git clone

- Fetches a local copy of an existing Git repo

Save your stuff.

git add .

- Stage changed files into next commit

git commit

- Save a snapshot of staged files into repo
- `git commit` will open a text editor for you to enter description of commit
- `git commit -m "<description>"` to enter description right away

git commit

- Save a snapshot of staged files into repo
- `git commit` will open a text editor for you to enter description of commit
- `git commit -m "<description>"` to enter description right away

git commit

- Save a snapshot of staged files into repo
- `git commit` will open a text editor for you to enter description of commit
- `git commit -m "<description>"` to enter description right away

git push

- Syncs local changes into remote repository
- `git push origin master` syncs current local repository to “origin” remote repository’s master branch

git pull

- Merges remote repository changes into local copy
- `git pull origin master` pulls changes from “origin” remote repo, “master” branch

Basic Workflow

Basic Workflow

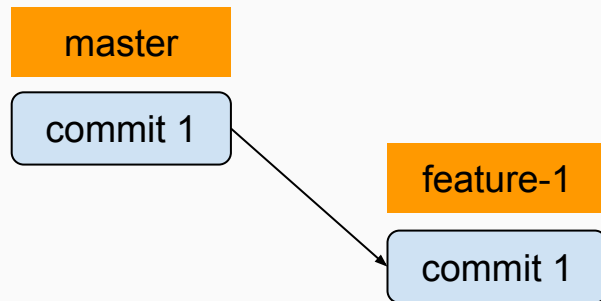
master

commit 1

Basic Workflow

Step 1. Create a new branch

```
git branch -b feature-1
```



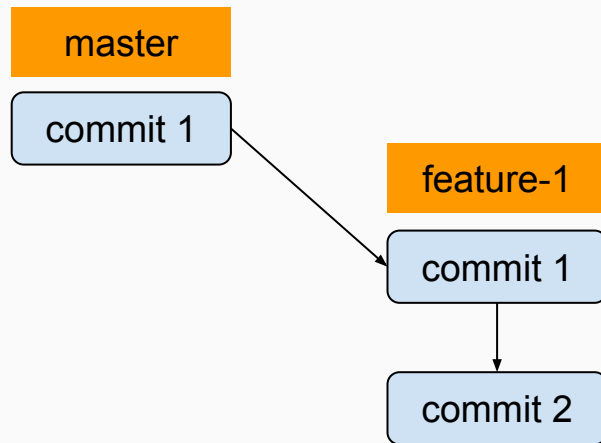
Basic Workflow

Step 1. Create a new branch

```
git branch -b feature-1
```

Step 2. Push all commits to that branch

```
git push -u origin feature-1
```



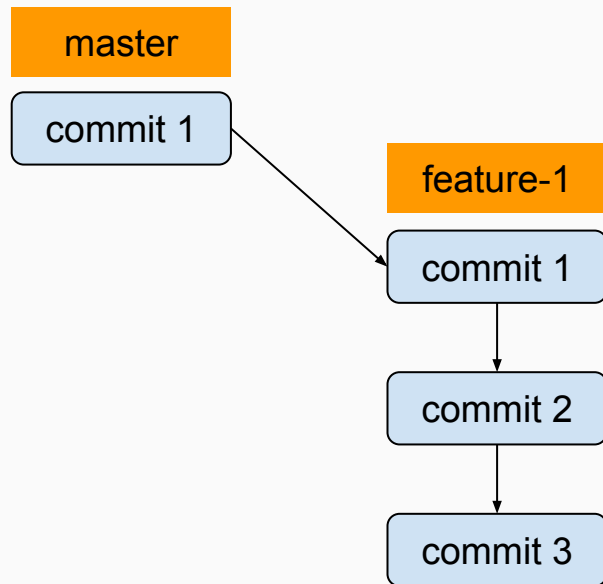
Basic Workflow

Step 1. Create a new branch

```
git branch -b feature-1
```

Step 2. Push all commits to that branch

```
git push -u origin feature-1
```



Basic Workflow

Step 1. Create a new branch

```
git branch -b feature-1
```

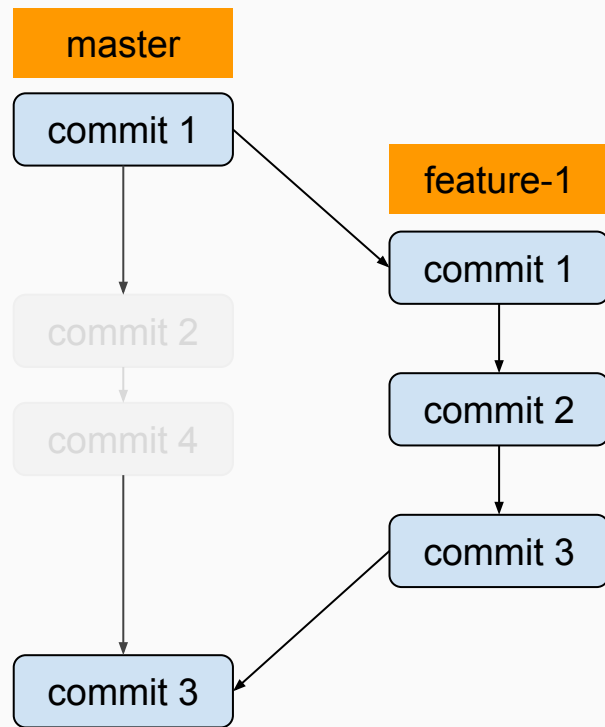
Step 2. Push all commits to that branch

```
git push -u origin feature-1
```

Step 3. Once done, merge changes to master

```
git checkout master
```

```
git merge feature-1
```



Basic Workflow

Step 1. Create a new branch

```
git branch -b feature-1
```

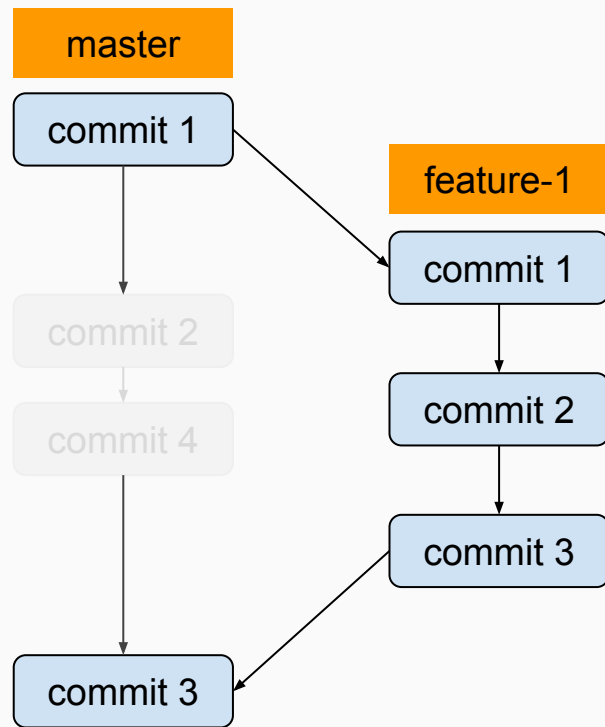
Step 2. Push all commits to that branch

```
git push -u origin feature-1
```

Step 3. Once done, merge changes to master

```
git checkout master
```

```
git merge feature-1
```



Exercise #1

1. Clone [git@github.com:microsat-dpad/git-demo.git](https://github.com/microsat-dpad/git-demo.git)
2. Create a hello world script (eg. kim.py, rk.py, jerine.py)
3. Push your hello world script to Github

Issue: What if master has changed?

Solution: Better Workflow

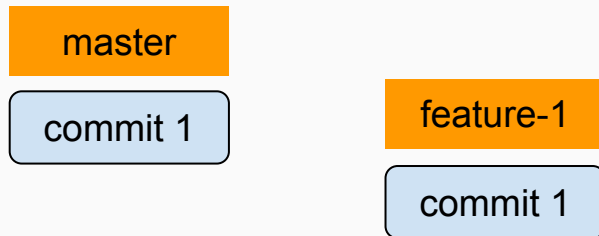
Better Workflow

master

commit 1

Better Workflow

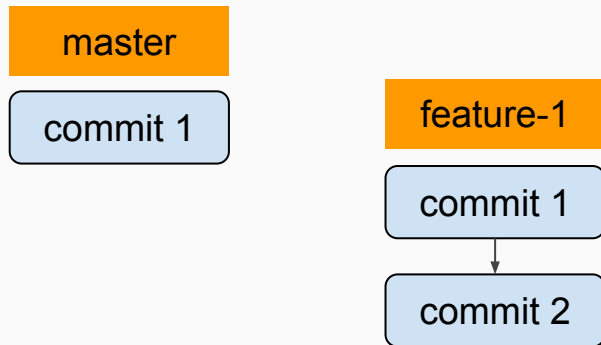
Step 1. Create a new branch



Better Workflow

Step 1. Create a new branch

Step 2. Push all commits to that branch

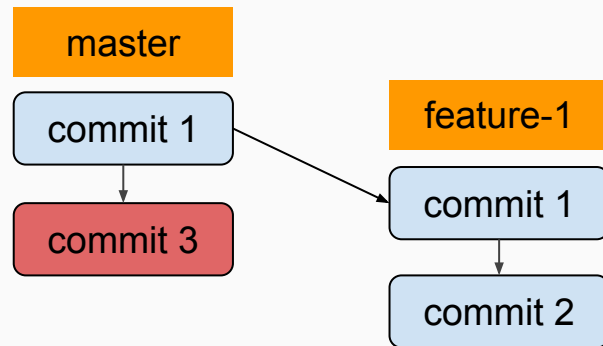


Better Workflow

Step 1. Create a new branch

Step 2. Push all commits to that branch

* May nag push sa master!

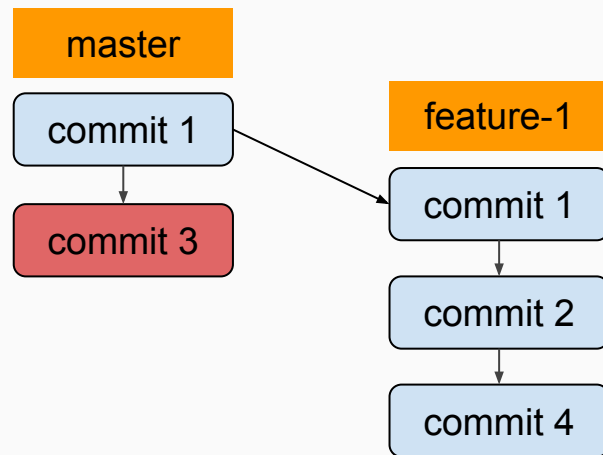


Better Workflow

Step 1. Create a new branch

Step 2. Push all commits to that branch

* May nag push sa master!



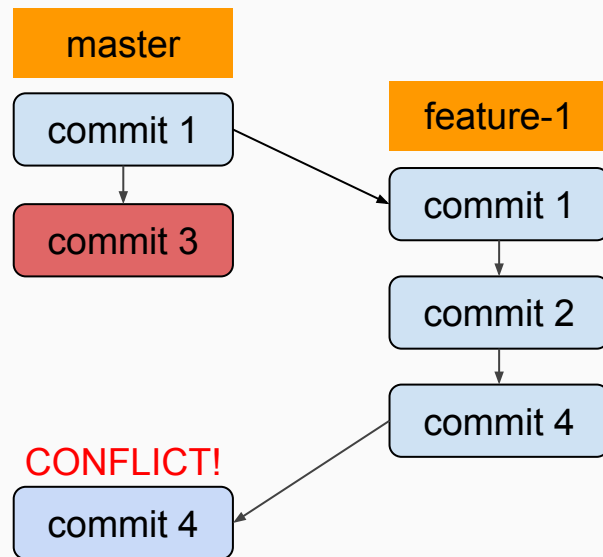
Better Workflow

Step 1. Create a new branch

Step 2. Push all commits to that branch

* May nag push sa master!

Step 3. Once done, merge changes to master



Better Workflow

Step 1. Create a new branch

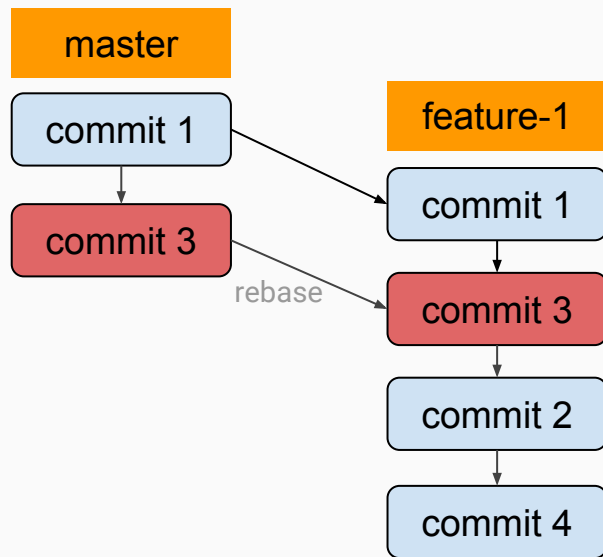
Step 2. Push all commits to that branch

* May nag push sa master!

Step 3. Once done, merge changes to master REBASE!

git fetch

git rebase master



Better Workflow

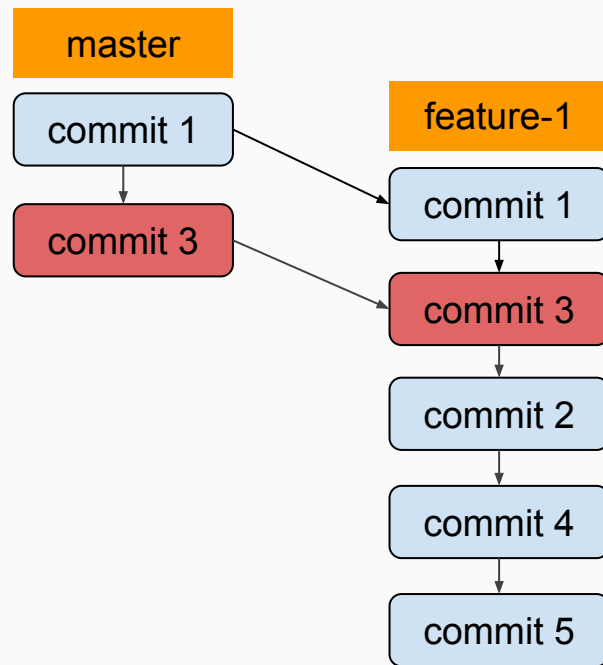
Step 1. Create a new branch

Step 2. Push all commits to that branch

* May nag push sa master!

Step 3. Once done, ~~merge changes to master~~ REBASE!

Step 4. Resolve possible conflicts



Better Workflow

Step 1. Create a new branch

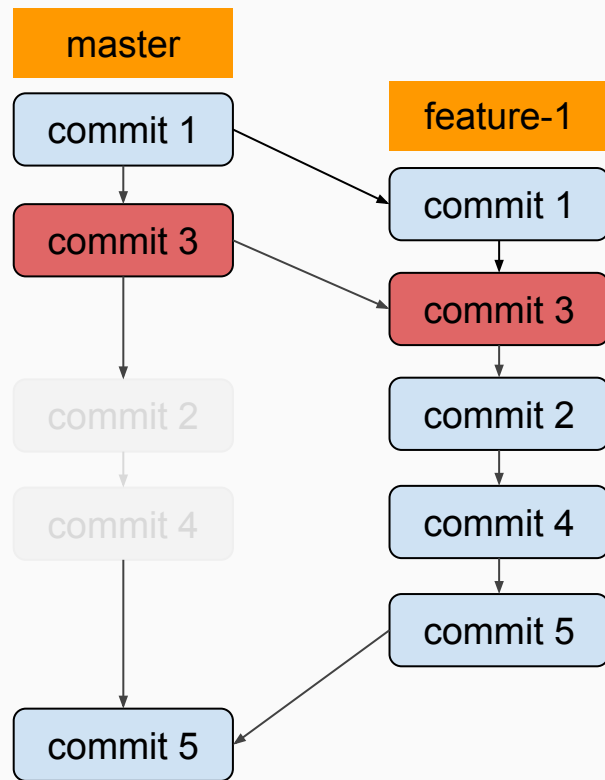
Step 2. Push all commits to that branch

* May nag push sa master!

Step 3. Once done, ~~merge changes to~~ master REBASE!

Step 4. Resolve possible conflicts

Step 5. Merge!



Exercise #2

1. Pull latest changes from repo
2. Follow new better workflow to push changes into README.md (Add instructions on how to run your hello world script)