An introduction to Git: what it is, and how to use it

Noushin Nabavi

2019-03-17

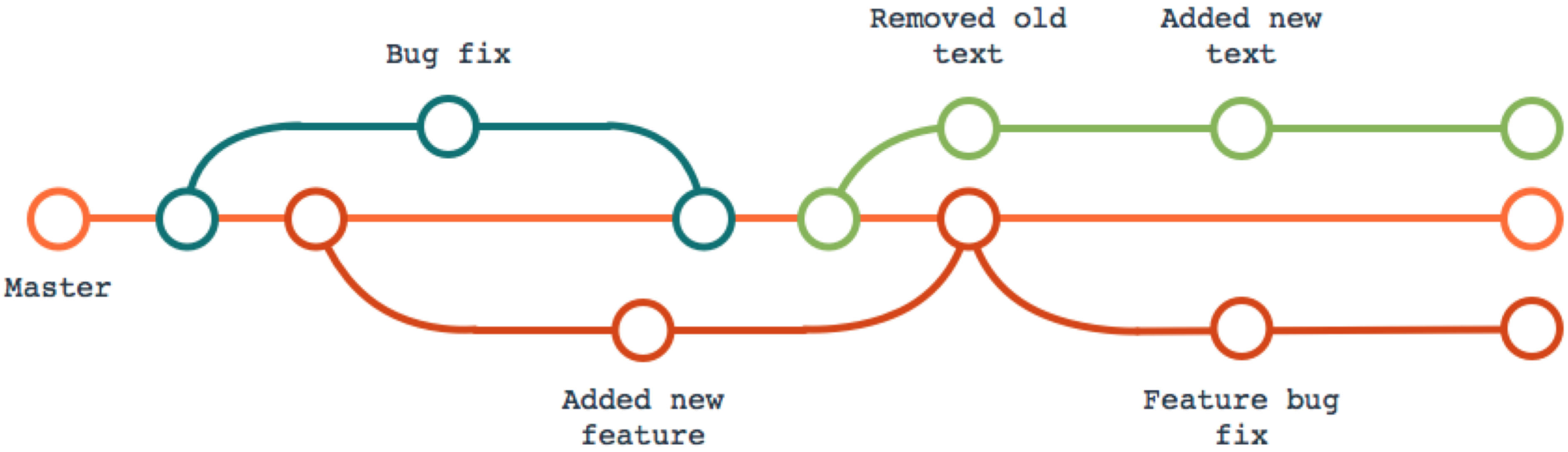
What is ***git***?

git is free and open source

is a distributed version-control system for tracking changes in source code during software development (est. 2005).

It is designed for coordinating work among programmers, but it can be used to track changes in any set of files (i.e. content *tracker*).

Its goals include speed, data integrity, and support for distributed, non-linear workflows



A peek at how Git branching might look over time

1. Getting started:

- Download ***git***: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

- Check if download worked by typing
  `git –version` in command line

2. Create your local Git repository:

- Let’s call the project folder git-demo:

  `cd git-demo`
  `git init`

- (git init command adds a local Git repository to the project)

3. Let’s Add some Small Code now

- Create a file called demo.txt in the project folder and add the following text into it:

```
Initial Content
```

4. Staging the code:

- Use the following command for staging the file:

```
`git add demo.txt`
```

- In case you want to add multiple files you can use:

```
`git add file1 file2 file3`
```

- If you want to add all the files inside your project folder to the staging area, use the following command:

```
`git add .`
```

5. Staging the code:

- Committing is the process in which the code is added to the **local repository**. Before committing the code, it has to be in the **staging area**. The staging area is there to keep track of all the files which are to be committed.

- Any file which is not added to the staging area will not be committed. This gives the developer control over which files need to be committed.

- Use the following command to commit the file:

```
`git commit –m "Initial Commit”`
```

- “Initial Commit” is the commit message here (Enter a relevant commit message to indicate what code changes were done in that particular commit)

6. Git Status:

- Use git status to find out information regarding what files are modified and what files are there in the staging area — it shows other information as well, which we can ignore for now.

```
`git status`
```

7. Git log:

Use git log to print out all the commits which have been done up until now.
The command used for this is:

`git log`

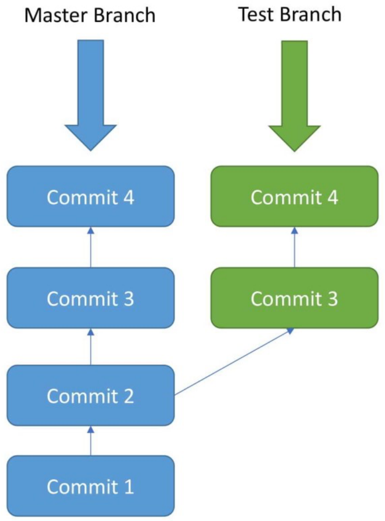
The log shows the author of each commit, the date of the commit, and the commit message.

Branches

- Up until now we have not created any branch in Git. By default, Git commits go into the **master** branch.

What is a branch?

- A branch is nothing but a pointer to the latest commit in the Git repository. So currently our master branch is a pointer to the second commit “demo.txt file is modified”.



8. Create a New Branch in local folder

- Create a new branch called **test** using the following command:

```
`git branch test`
```

Do Some Commits in the New Branch

Modify demo.txt by adding the following snippet:

Initial Content
Adding more Content
Adding some Content from test Branch

Now stage and commit using the following commands:

```
`git add demo.txt `
`git commit -m "Test Branch Commit”`
```

You can verify the commit history in Test Branch using:

```
`git log`
```

9. Merging:

Currently, Test Branch is ahead of the Master by 1 commit. Let’s say that now we want all the code in the Test Branch to be brought back to the Master Branch. This is where git merge is very useful.

In order to merge the code from the test branch into the master branch, follow these steps:

- First go back to the master branch:

```
`git checkout master`
```

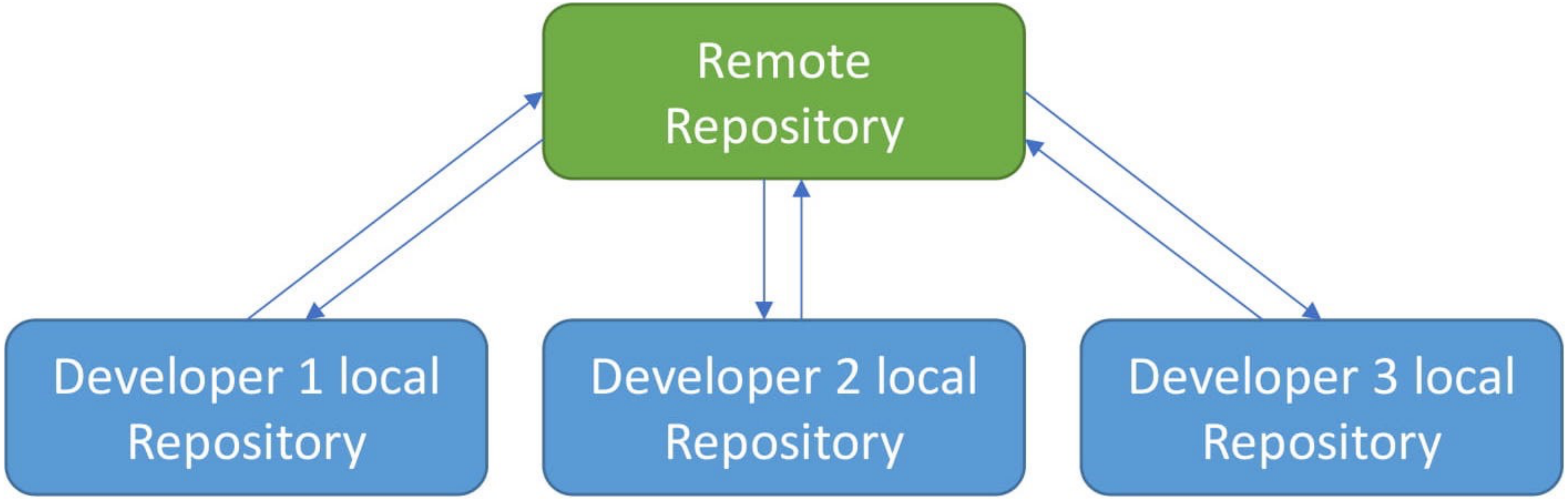
- Then run the merge command:

```
`git merge test`
```

Run `git log` now and you will notice that the master also has 3 commits.

The Remote Git Repository

- Until now, we have been working only in the local repository. Each developer will work in their local repository but eventually, they will push the code into a remote repository. Once the code is in the remote repository, other developers can see and modify that code.


Remote Repository
Developer 1 local Repository
Developer 2 local Repository
Developer 3 local Repository

GitHub can be used for the remote repository.

Go to https://github.com/ and create an account.

10. Github commands:

- In order to point your local repository to the remote repository, use the following command:

  `git remote add origin [repository url]`

- In order to push all the code from the local repository into the remote repository, use the following command:

  `git push -u origin master`

- git pull is used to pull the latest changes from the remote repository into the local repository. The remote repository code is updated continuously by various developers, hence git pull is necessary:

  `git pull origin master`

- git clone is used to clone an existing remote repository into your computer. The command for this is:

  `git clone [repository url]`