



A Tutorial

About Git

Version Control System (VCS) for tracking changes in computer files

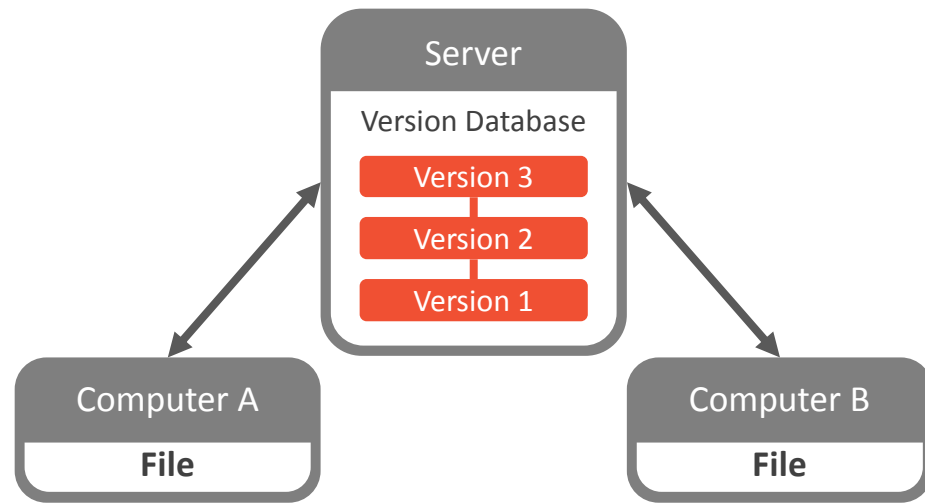
Distributed version control

Local and remote **Repositories**

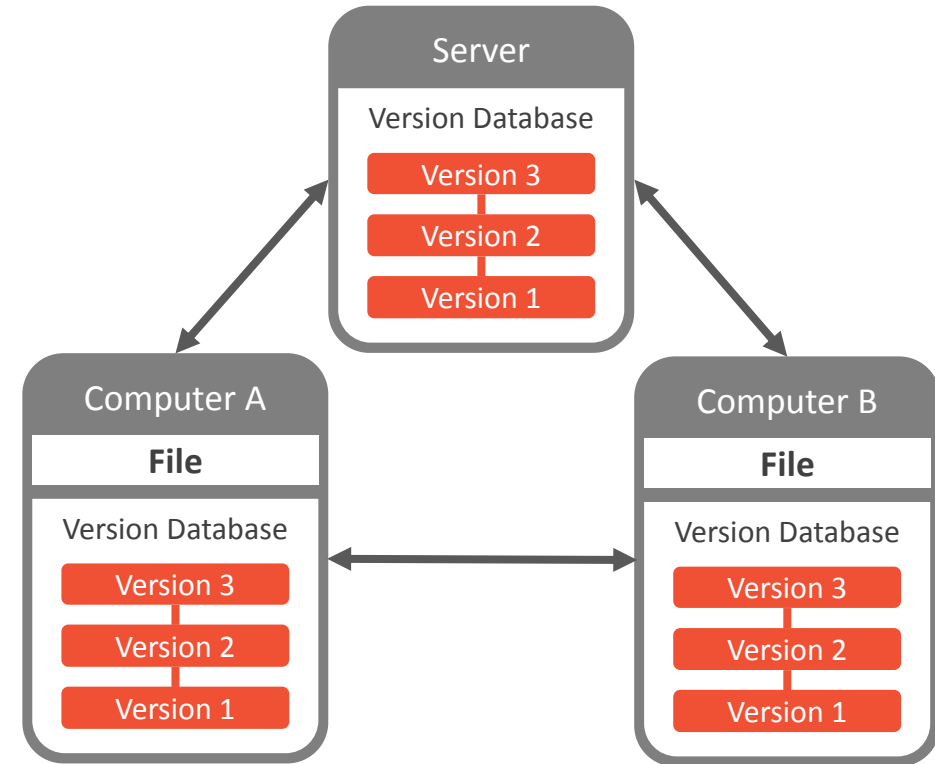


Created in 2005 by
Linus Torvalds

Version Control Systems



Centralized SVN, CVS, Perforce

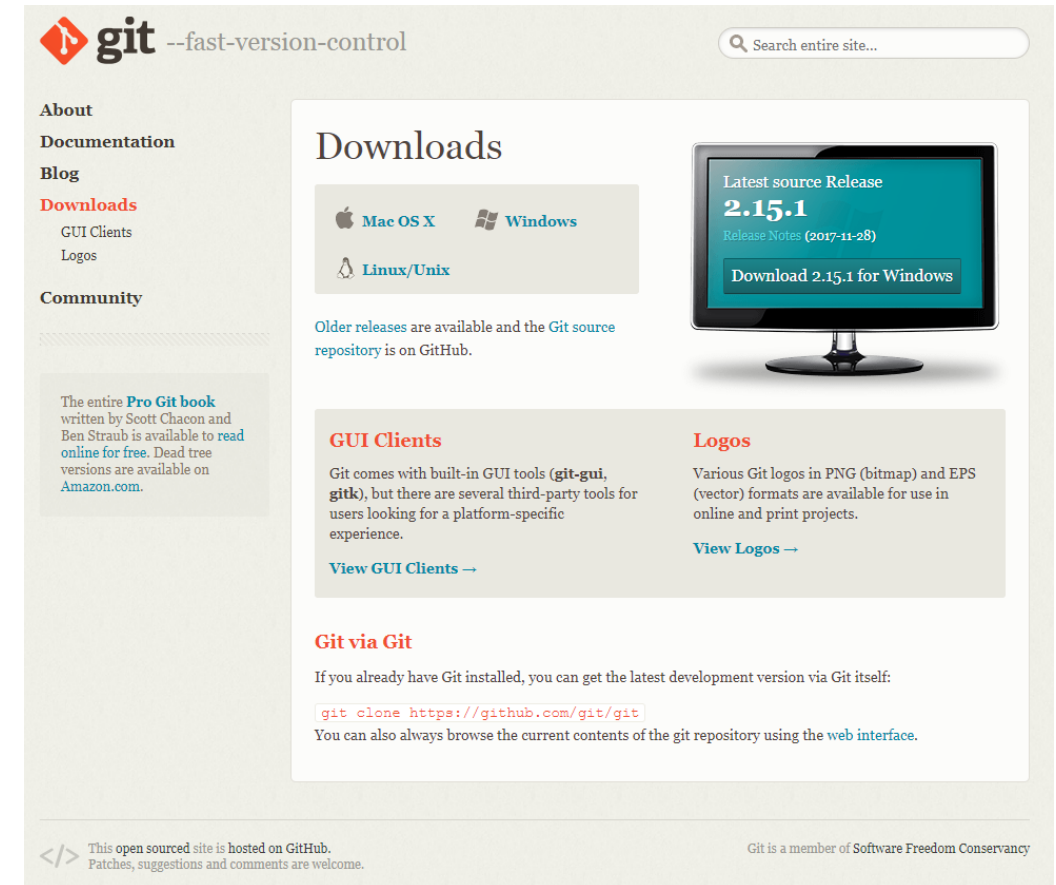


Distributed Git, Mercurial

Installation

<https://git-scm.com/downloads> offers
Install files for each operating system

Graphical User Interface (GUI) not
recommended



Configure Git

Verify installed git **Version**

```
git --version
```

Configure git

```
git config -global user.name "your name"  
git config -global user.email "your@email.nl"  
git config --list
```

Open **documentation** of verb

```
git help <verb>  
git <verb> --help
```

Initialize Repository

Initialize **Repository** at current location

```
git init
```



Check **Status** of repository

```
git status
```

File Status Lifecycle

Adding untracked files

```
git add <file>
```

Committing staged files

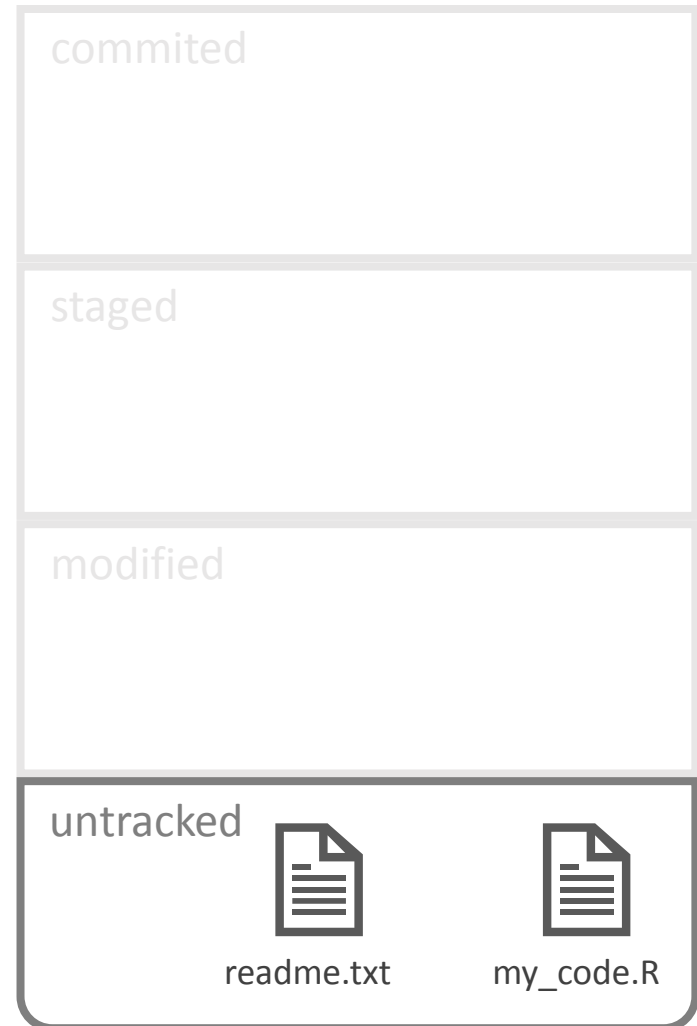
```
git commit -m "message"
```

Staging modified files

```
git add <file>
```

Removing staged files

```
git reset <file>
```



Log

Show **Logged** commits

```
git log
```

b3bd9b b06ecf



```
commit 401eeb8ec8b5c4727f867586668b50ec6ee71b82
```

```
Author: Your Name <your@email.nl>
```

```
Date:   Sun Jan 14 17:15:58 2018 +0100
```

```
Initial commit.
```

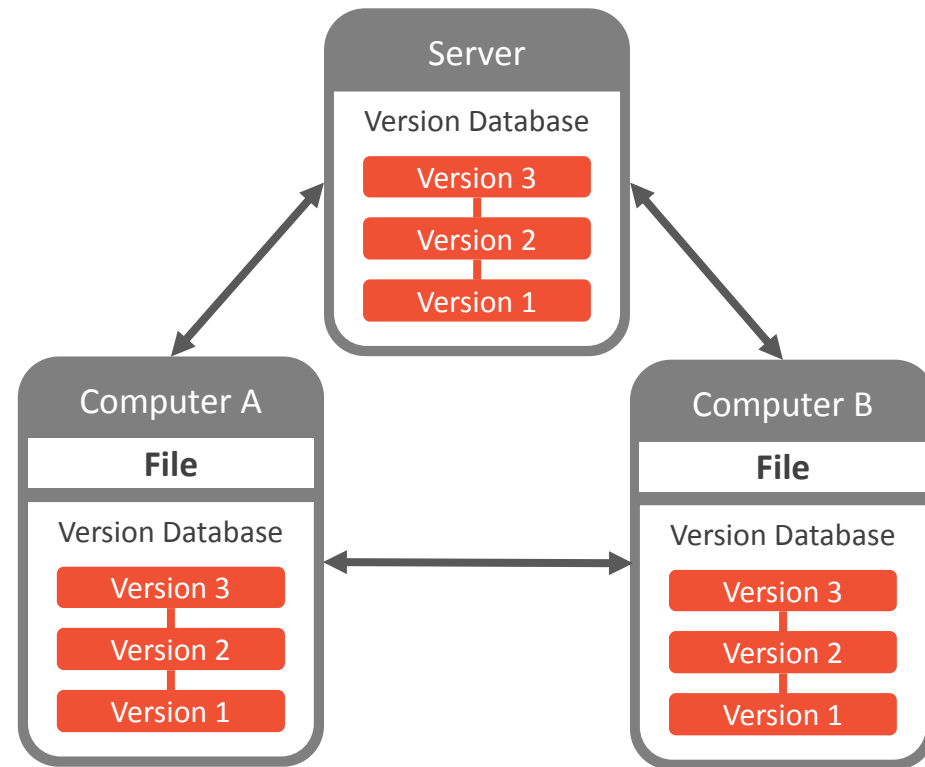
```
commit 338d88a99bc9d997e63343b3c3b3d3ee33f23d33
```

```
Author: Your Name <your@email.nl>
```

```
Date:   Mon Jan 15 13:53:20 2018 +0100
```

```
Modified readme file.
```


Remote Repositories



Distributed Version Control System

Remote Repositories

Clone a remote repository

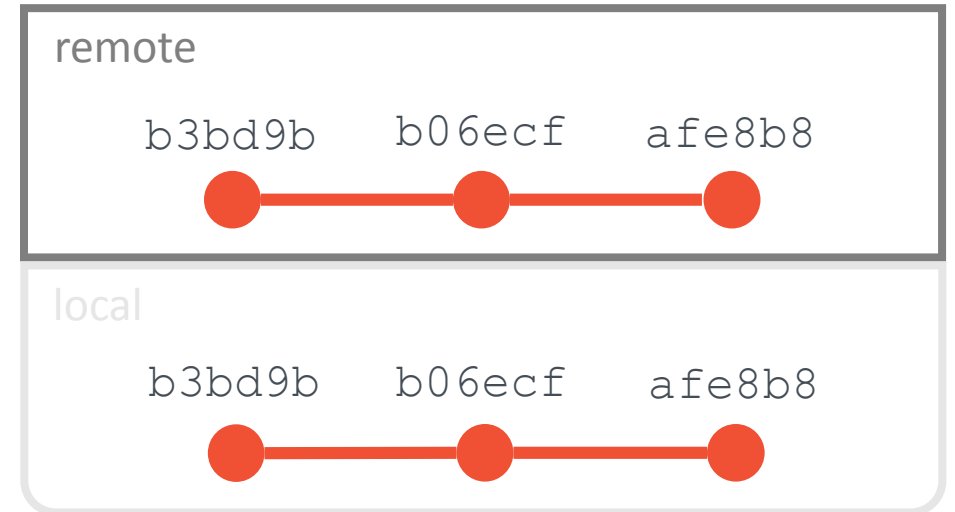
```
git clone <url> <where to clone>
```

Lists remote repository Information

```
git remote -v  
git branch -a
```

Push changes to remote repository

```
git pull origin master  
git push origin master
```



Branches

List local branches

```
git branch
```

Create a branch

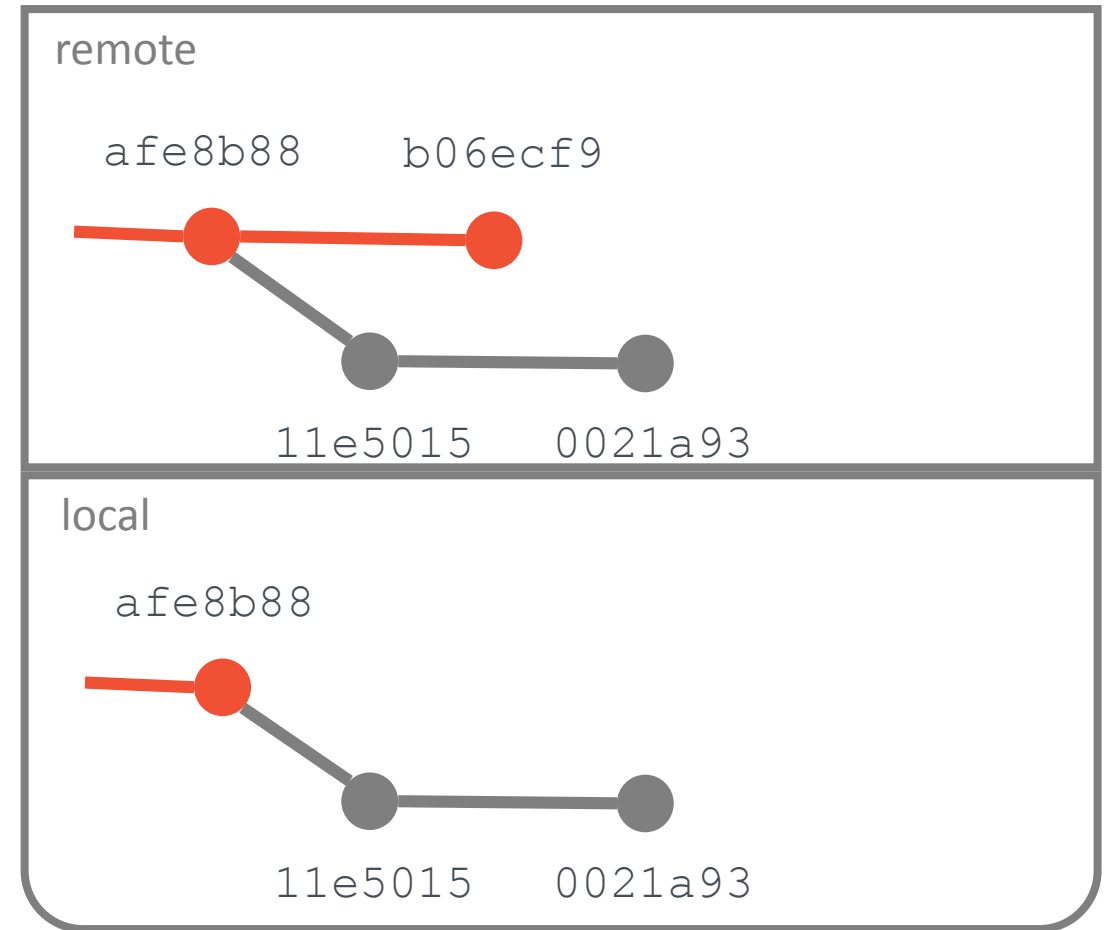
```
git branch <branch>
```

Enter a branch

```
git checkout <branch>
```

Push branch to repository

```
git push -u origin <branch>
```



Merging

Pull **Master** branch changes

```
git checkout master  
git pull origin master
```

Show **Merged** branches

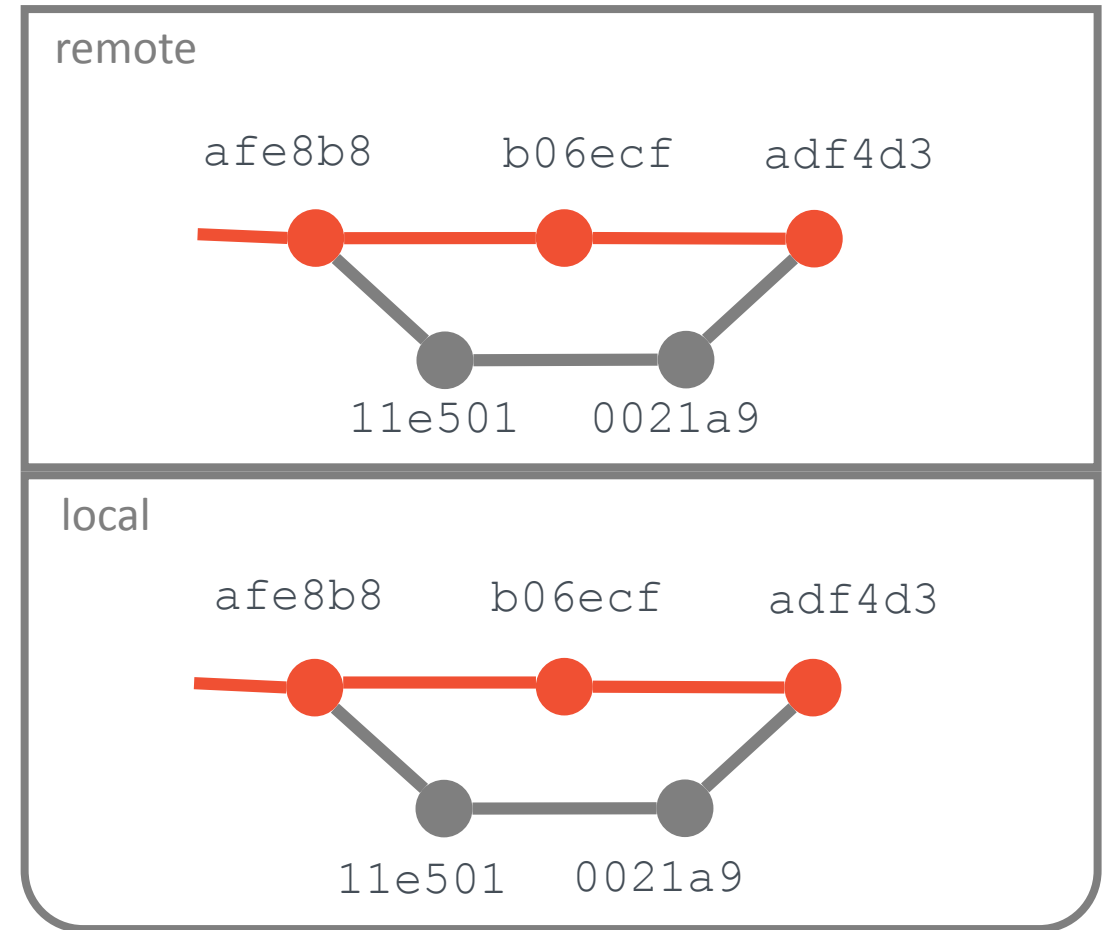
```
git branch --merged
```

Merge branch into master

```
git merge <branch>
```

Push merge to repository

```
git push origin master
```



Removing Branch

Show **Merged** branches

```
git branch --merged
```

Remove branch

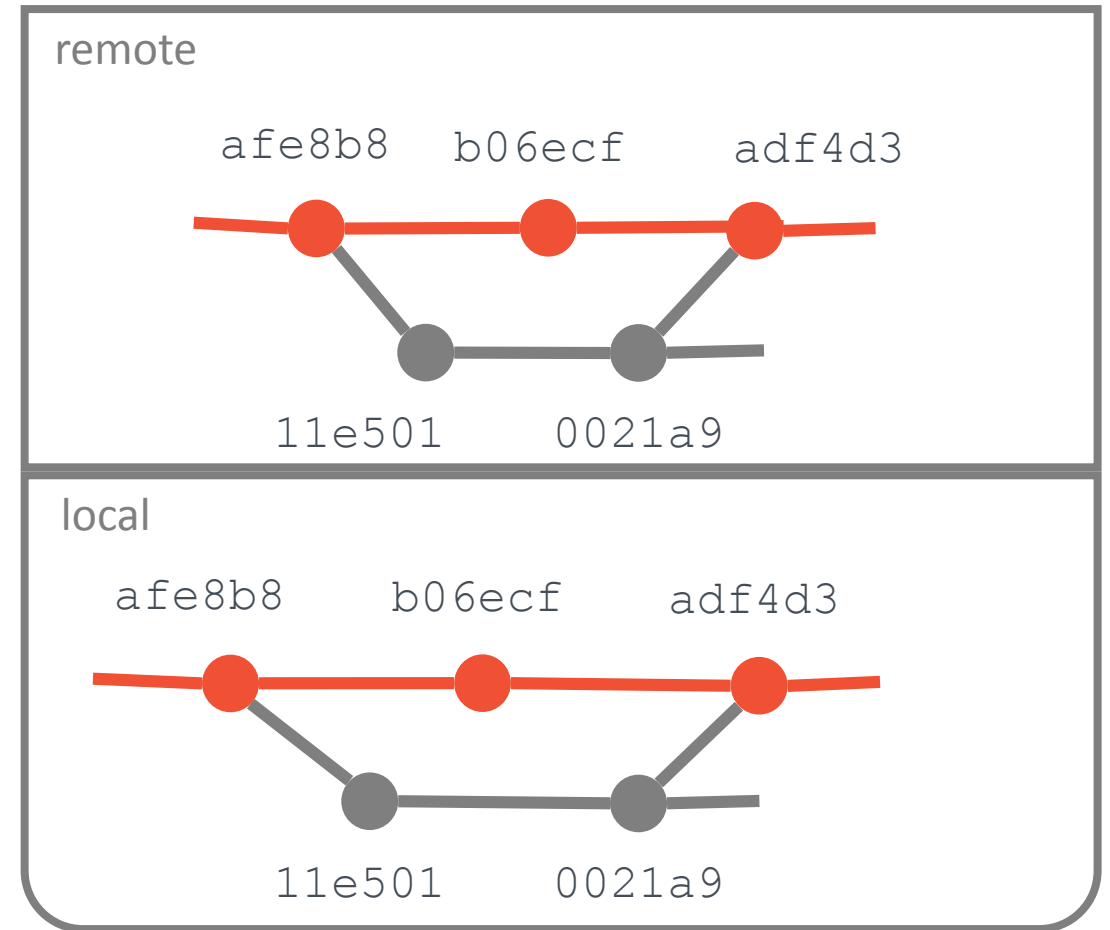
```
git branch -d <branch>
```

List **remote** branches

```
git branch -a
```

Remove branch from **remote** repository

```
git push origin -delete <branch>
```



Merge Conflicts

The conflicting file will contain <<< and >>> sections to indicate where Git was unable to resolve a conflict:

```
<<<<<< HEAD:index.html
<div id="footer">todo: message here</div>
=====
<div id="footer">
  thanks for visiting our site
</div>
>>>>>> SpecialBranch:index.html
```

} branch 1's version

} branch 2's version

Find all such sections, and edit them to the proper state (whichever of the two versions is newer / better / more correct).

Useful

Show **Differences** after modifying

```
git diff
```

Tag a commit

```
git tag
```

Set **Editor** for commit messages - *default* vim

```
git config --global core.editor <editor>
```



.gitignore

References

Linus Torvald's **Portrait** from <https://github.com/torvalds>