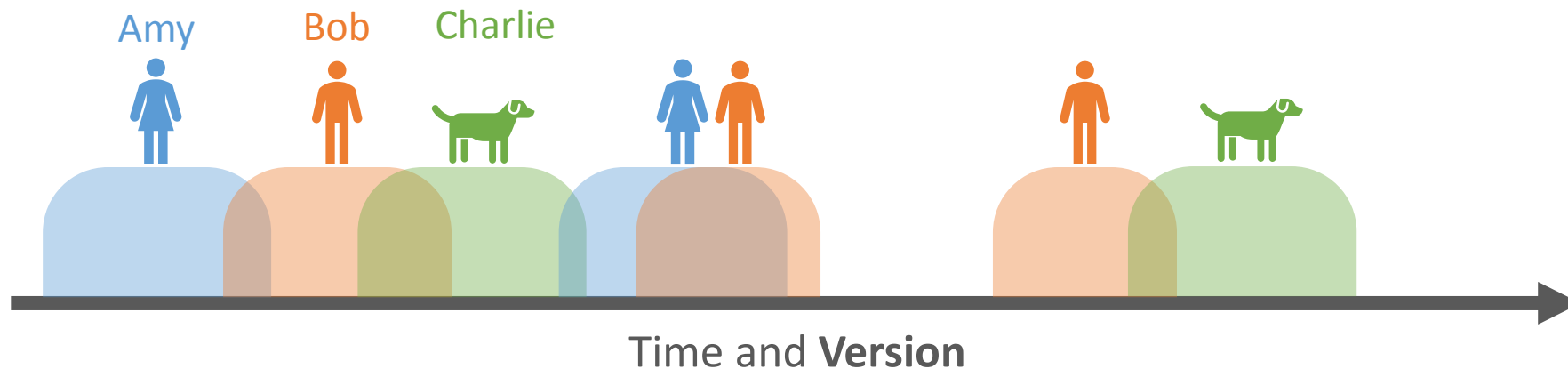




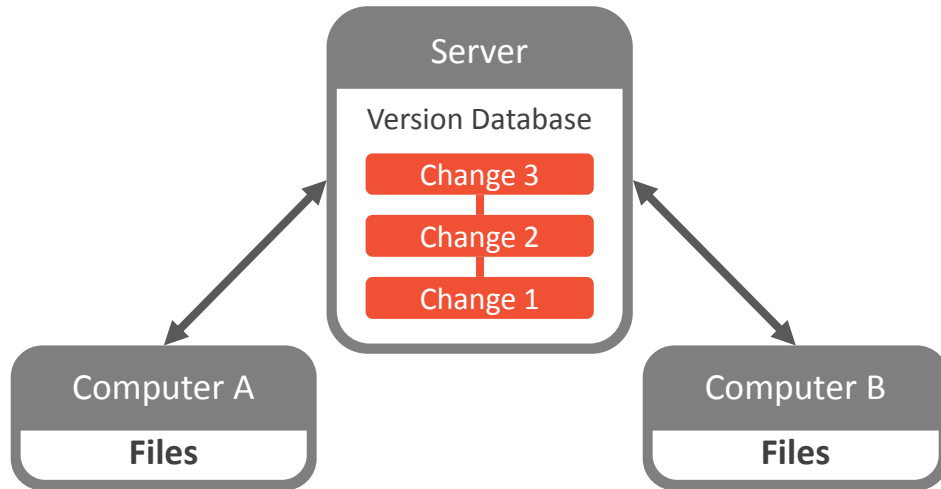
A Tutorial

Version Control

- Tracks **changes** in computer files
- **Documentation** of project
- Efficient **collaboration** with colleagues

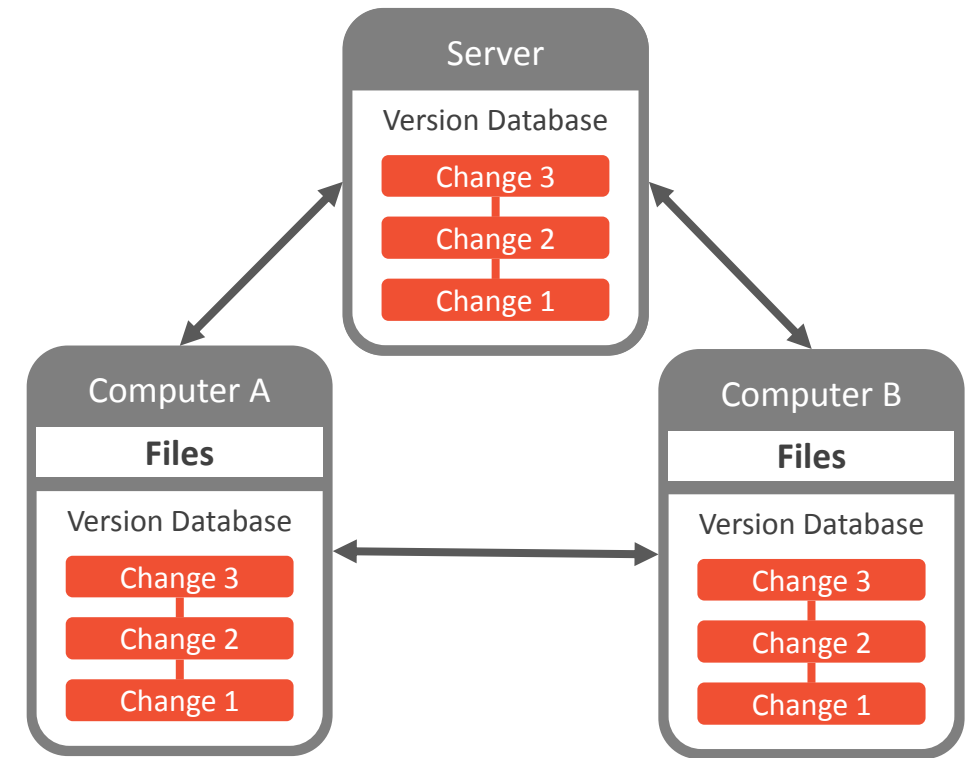


Version Control Systems



Centralized SVN, CVS, Perforce

+ manage access rights



Distributed Git, Mercurial

+ local version database

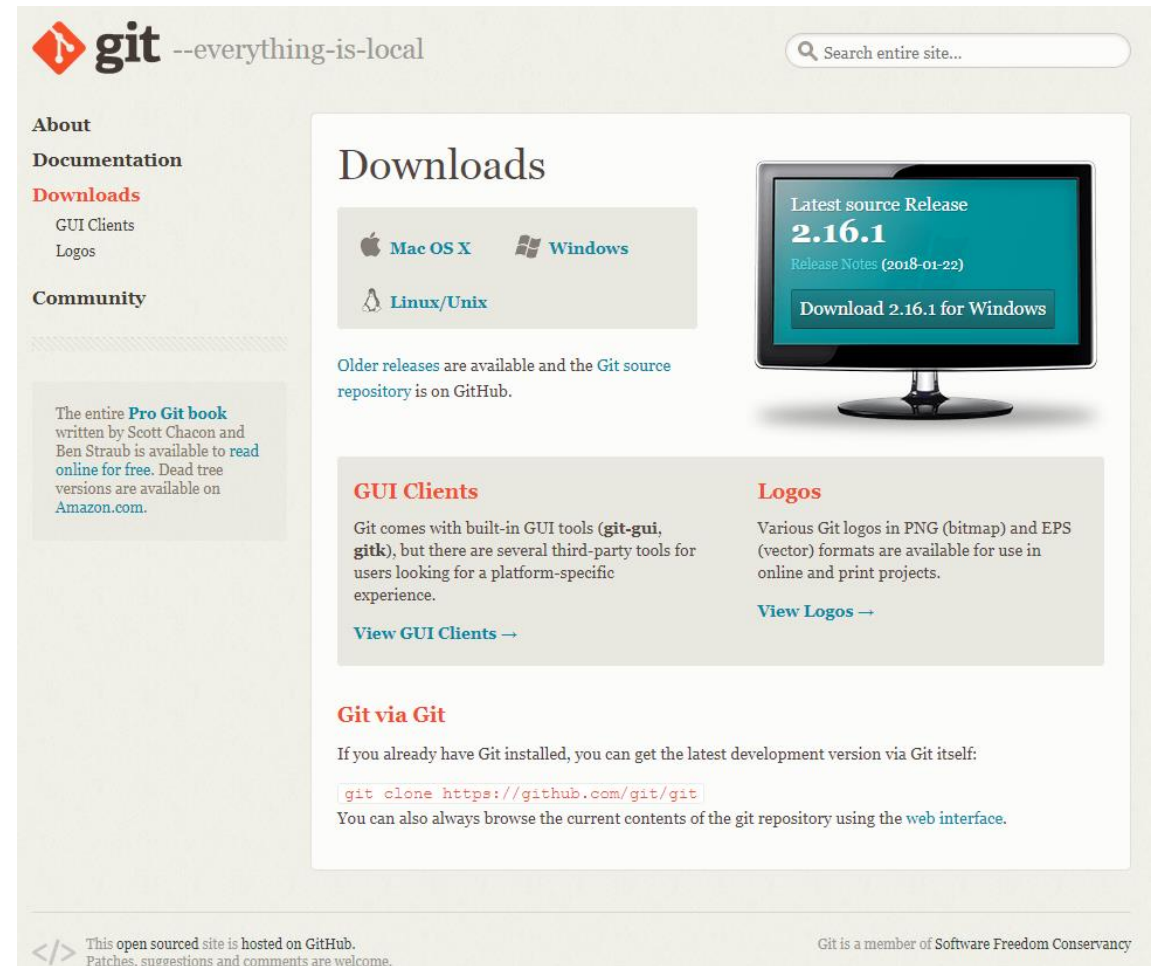
+ backup on collaborator machines

About Git

- Git is a **distributed** version control system for local and remote repositories
- Git is advertised with the following features
 - Offers **review** functionality
 - Work **offline** anywhere
 - **Fast** and lightweight
 - **Journal** of changes rather than a backup
 - Serves the needs of **beginners** and **advanced** users equally well

Installation

- **Download** at <https://git-scm.com/downloads>



The screenshot shows the Git website's Downloads page. At the top, the Git logo is followed by the tagline "--everything-is-local". A search bar is in the top right corner. On the left, a navigation menu includes links for "About", "Documentation", "Downloads" (which is highlighted), "GUI Clients", "Logos", and "Community". Below the menu, a text box mentions the "Pro Git book" by Scott Chacon and Ben Straub, available online for free. The main content area is titled "Downloads" and features a large monitor graphic displaying the "Latest source Release 2.16.1" with a "Download 2.16.1 for Windows" button. Below the monitor, a text box states that older releases are available on GitHub. To the left of the monitor, there are links for "Mac OS X", "Windows", and "Linux/Unix". Further down, there are sections for "GUI Clients" and "Logos", each with a brief description and a link to view more. At the bottom, a "Git via Git" section provides a command to clone the repository and a link to the web interface. The footer contains a note about the site being open sourced and hosted on GitHub, and a statement that Git is a member of the Software Freedom Conservancy.

git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows
Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.16.1
Release Notes (2018-01-22)
Download 2.16.1 for Windows

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

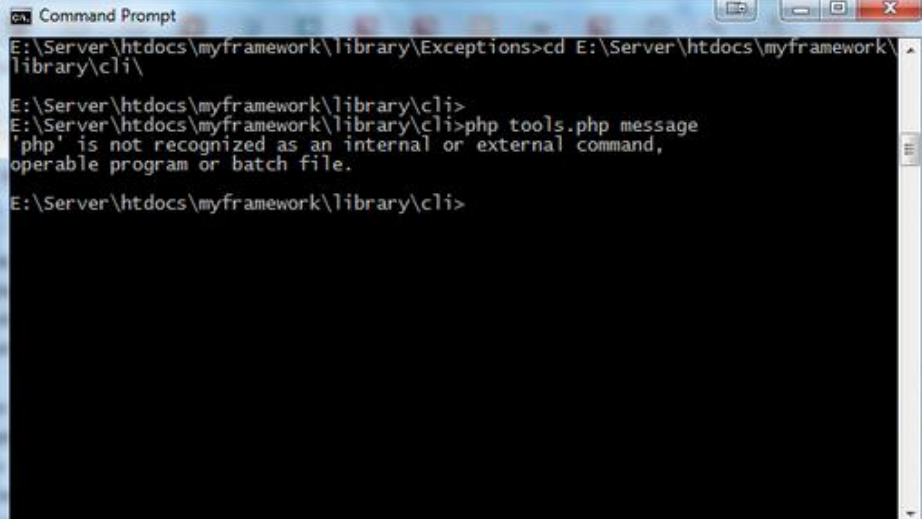
</> This open sourced site is hosted on GitHub.
Patches, suggestions and comments are welcome.

Git is a member of Software Freedom Conservancy

Interfaces

Command Line Interface (CLI)

- Syntax: `git [verb] [options]`
- Allows for **customized** commands
- Commands are **platform-independent**



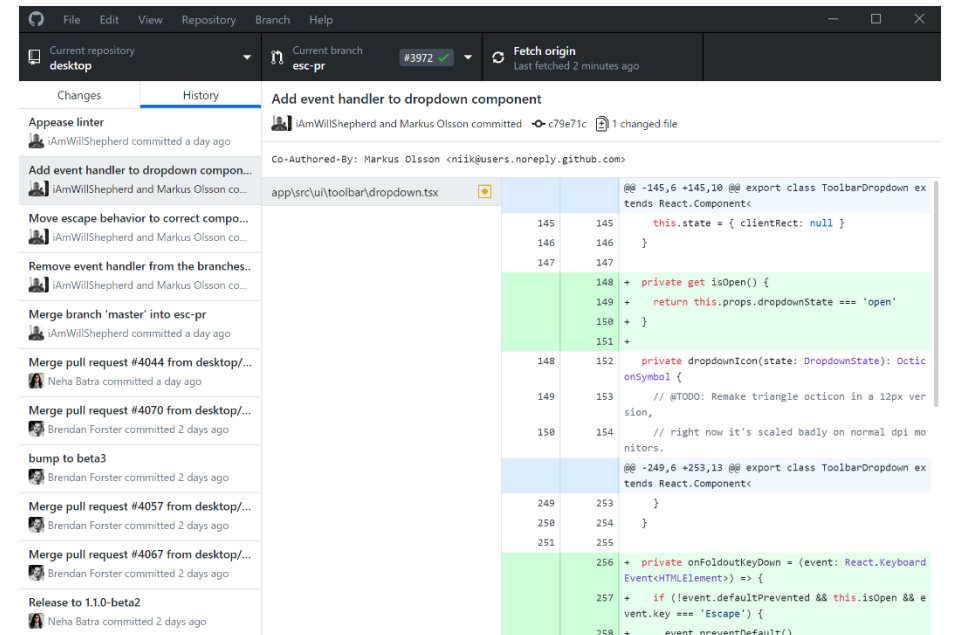
```
Command Prompt
E:\Server\htdocs\myframework\library\Exceptions>cd E:\Server\htdocs\myframework\library\cli\
E:\Server\htdocs\myframework\library\cli>
E:\Server\htdocs\myframework\library\cli>php tools.php message
'php' is not recognized as an internal or external command,
operable program or batch file.
E:\Server\htdocs\myframework\library\cli>
```

Graphical User Interface (GUI)

- Saves **typing**
- Lower **barrier-of-entry**
- **Graphically** represents collaboration

Integrated Development Environment (IDE)

- Integrates version control with **projects**



Configure Git

Verify installed git **version**

```
git --version
```

Configure git

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

Open **documentation** of verb

```
git help [verb]  
git [verb] --help
```

Local Repositories

The local repository is the project folder under version control. It contains the .git subfolder and all project related files and subfolders.

Initialize local **repository**

```
git init [project-name]
```

Check **status** of local repository

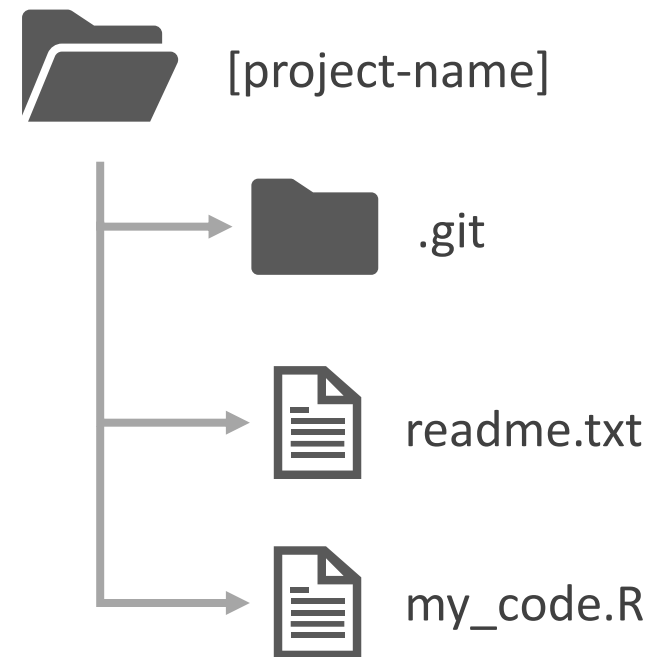
```
git status
```

Create empty file on Windows

```
copy NUL [file-name]
```

Create empty file on MacOs, Linux

```
touch [file-name]
```



File Status Lifecycle

Stage untracked/modified file

```
git add [file-name]
```

Commit staged files

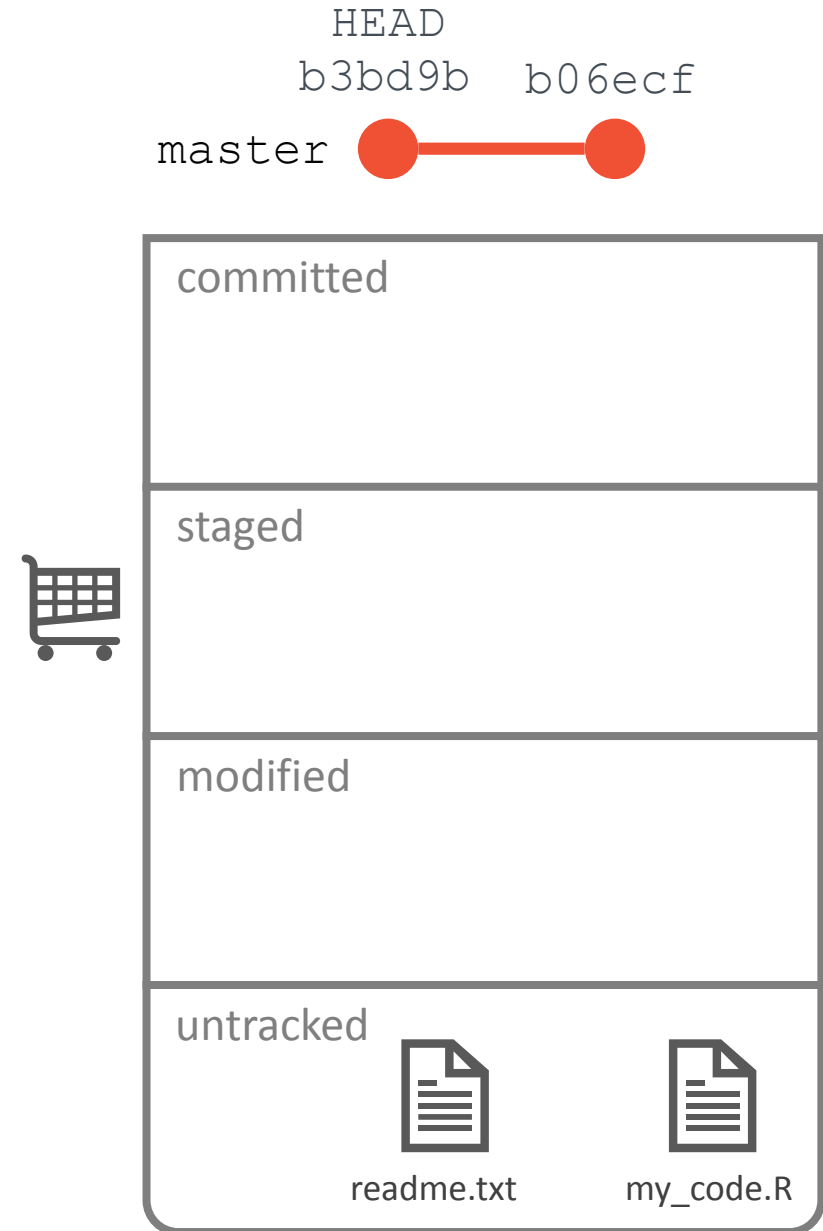
```
git commit -m "[message]"
```

Show modification

```
git diff HEAD
```

Stage all files

```
git add -A
```



Log



Show **logged** commits

```
git log
```

Show command **options**

```
git help log
```

Show **formatted** log message and **graph**

```
git log --graph --oneline
```

Restore previous version

```
git checkout [commit-hash]
```

Commit message

- Use imperative mode for summary line. Start with "Fix", "Add", instead of "Fixed", "Added".
- Leave the second line blank
- Line break the commit message (at about 72 characters)
- Don't end the summary line with a period, as it is a title

Short (50 chars or less) summary of changes

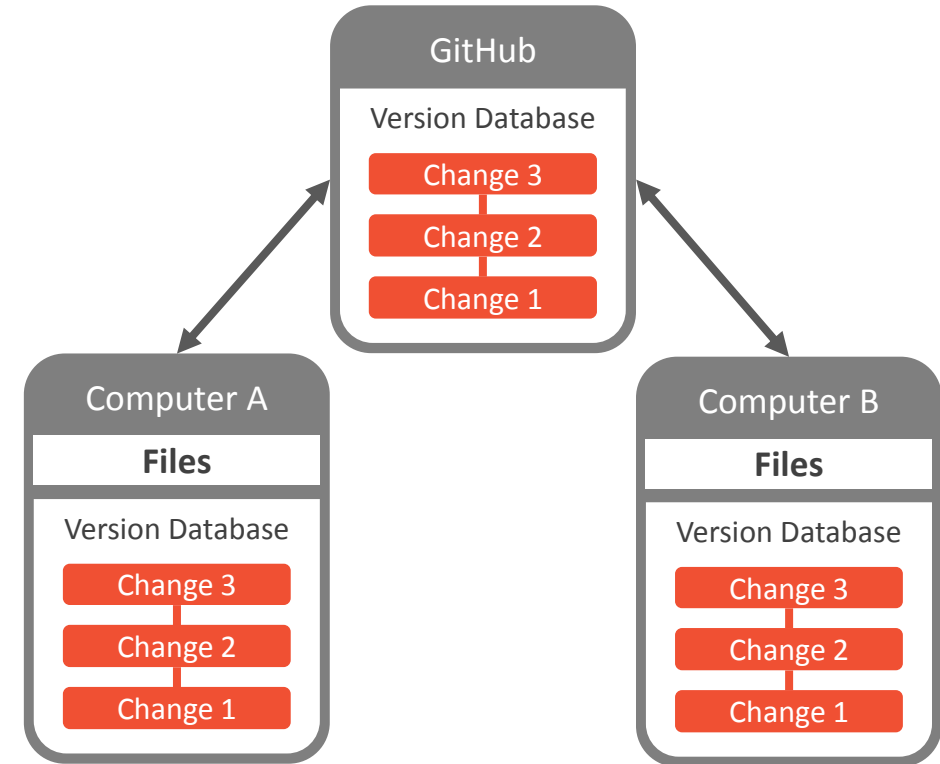
More detailed explanatory text, if necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of an email and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); tools like rebase can get confused if you run the two together.

Further paragraphs come after blank lines.

- Bullet points are okay, too
- Typically a hyphen or asterisk is used for the bullet, preceded by a single space, with blank lines in between, but conventions vary here

Remote Repositories

- **Bare repository** only contains version database
- **Working repository** generates working directory from version database



Distributed Version Control System

GitHub

- **Repositories** store all files of a particular project
- **Visualize** changelogs
- **Social network** infrastructure

The screenshot shows the GitHub interface for the repository **OHDSI / PatientLevelPrediction**. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name is displayed along with statistics: 67 Watchers, 23 Stars, and 23 Forks. A tab bar shows 'Code' as the active tab, with other tabs for Issues (9), Pull requests (2), Projects (0), and Insights.

The repository description states: "An R package for performing patient level prediction." Below this, a summary bar shows 305 commits, 9 branches, 5 releases, and 6 contributors. Action buttons include 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'.

A recent commit by **PRijnbeek** is highlighted: "Added link to adding existing models vignette." Below this, a list of files and folders is shown with their commit messages and timestamps:

File/Folder	Commit Message	Time Ago
.settings	Nesting the vignette study in a diabetes population to reduce size. S...	3 years ago
R	delayed files	13 days ago
data	New simulation profile, with two outcomes instead of one, for better ...	2 years ago
demo	Fixes #27	10 months ago
extras	added function for existing model	13 days ago
inst	added function for existing model	13 days ago
man-roxygen	Auto code formatting	2 years ago
man	delayed files	13 days ago
src	Fixed some more check problems, including Rcpp's new issue with funct...	5 months ago
tests	removed multianalyses	2 months ago
vignettes	added function for existing model	13 days ago
.Rbuildignore	temporarily stop using drat	8 months ago
.cproject	Nesting the vignette study in a diabetes population to reduce size. S...	3 years ago
.gitattributes	silly windoz cr/lf issues	3 years ago
.gitignore	- merged Logger feature in	2 years ago
.project	Nesting the vignette study in a diabetes population to reduce size. S...	3 years ago
.travis.yml	removing uploading git token	6 months ago

Remote Repositories

A project copy, which is hosted on a remote server: github.com/mi-erasmusmc/GitTutorial

Clone a remote repository

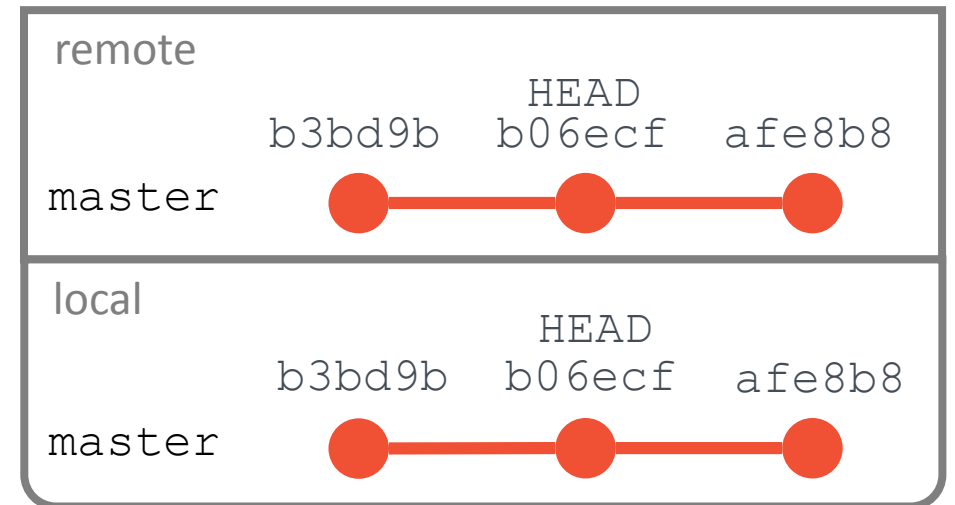
```
git clone [url] [where-to-clone]
```

Lists remote repository Information

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

Push changes to remote repository

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



Branches

Diverge from the main line of development to work independently.

List local branches

```
git branch
```

Create a branch

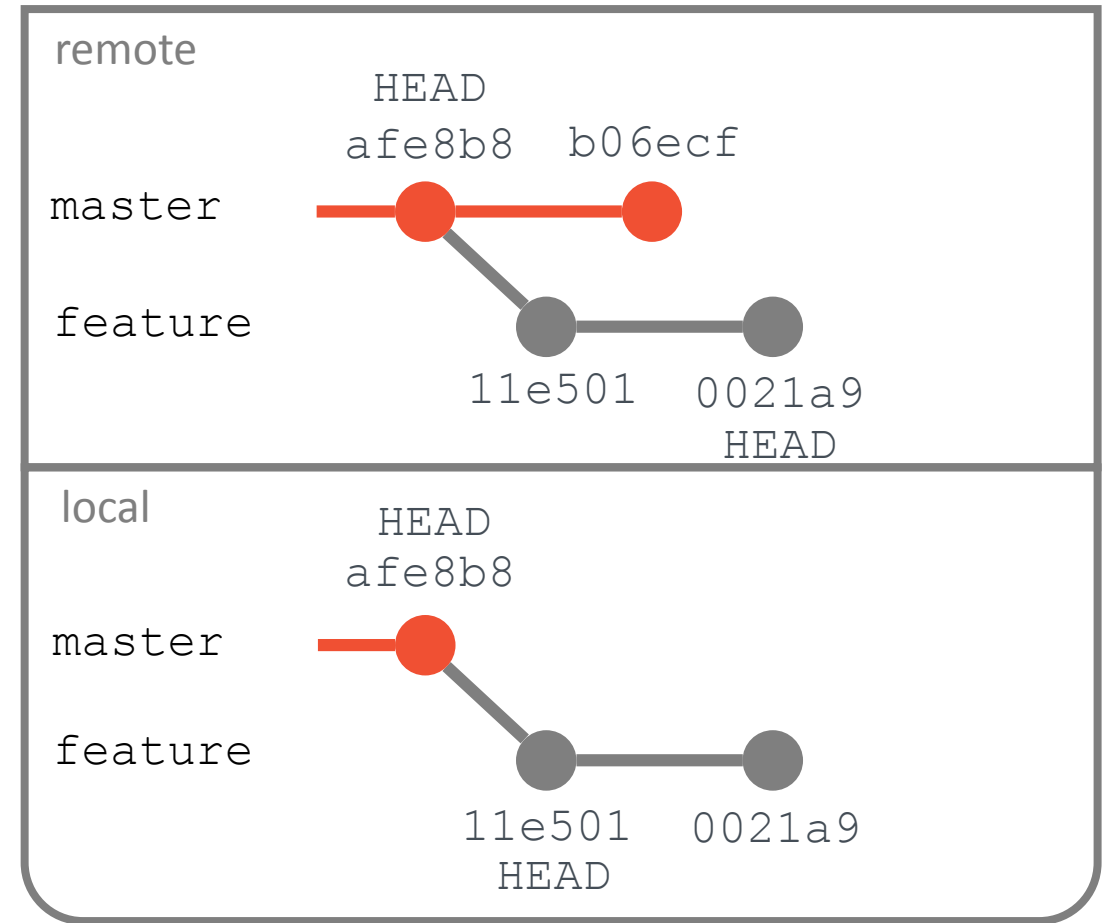
```
git branch [branch-name]
```

Enter a branch

```
git checkout [branch-name]
```

Push branch to remote repository

```
git push -u origin [branch-name]
```



Merging

Combine two lines of development after independent work is completed.

Pull **Master** branch changes

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

Show **Merged** branches

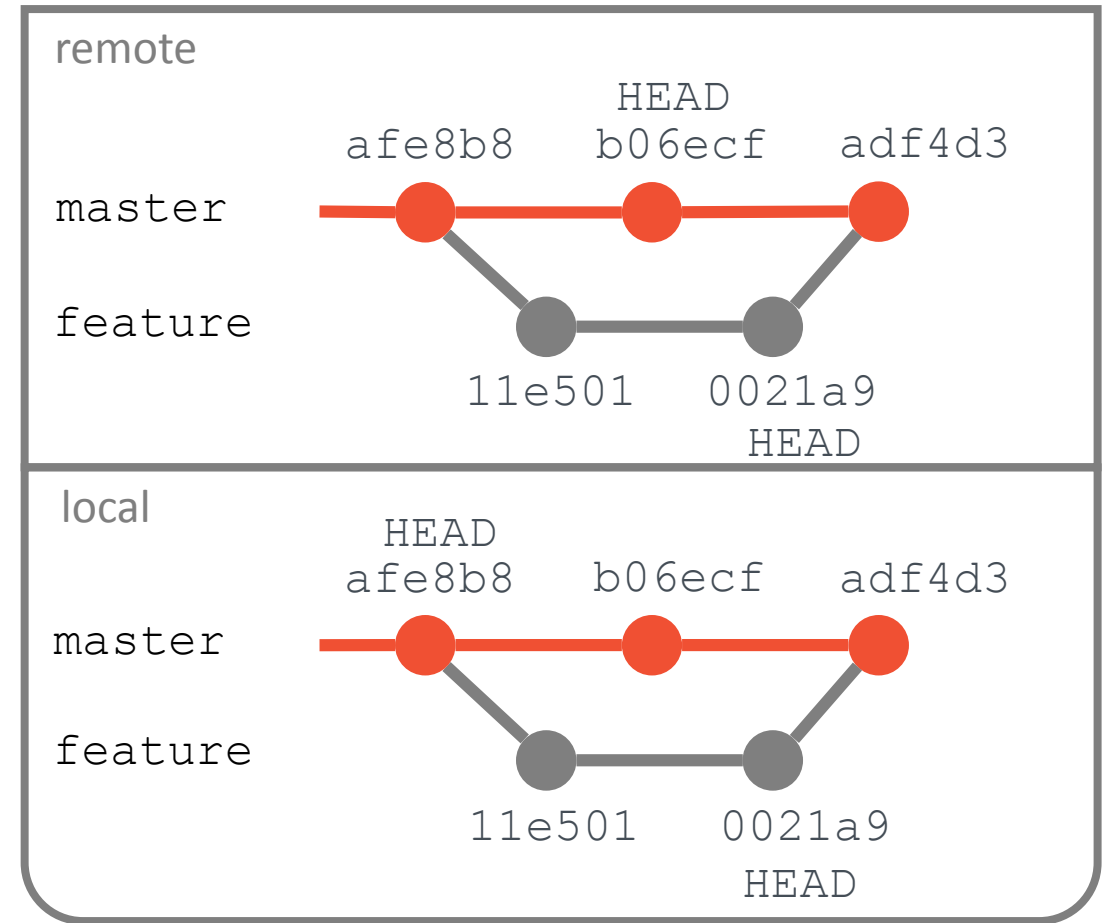
```
git branch --merged
```

Merge branch into master

```
git merge [branch-name]
```

Push merge to remote repository

```
git push origin master
```



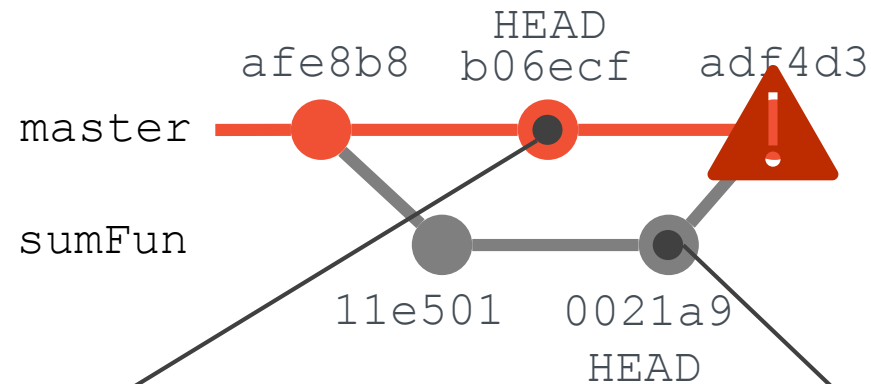
Merge Conflicts

```
# Take sum of vector
sum <- function(x) {
  sum <- 0

  # Take sum of vector
  sum <- function(x) {
    sum <- 0
    if (any(is.na(x))) {
      stop("x has NA values")
    }
    for (val in x) {
      sum <- sum + val
    }
    sum
  }
}
```



sum.R



```
git merge sumFun
```

```
Auto-merging sum.R
```

```
CONFLICT (content): merge
conflict in sum.R
```

```
Automatic merge failed; fix
conflicts and then commit the
result.
```

```
git mergetool
```

```
# Take sum of vector
sum <- function(x) {
  sum <- 0

  if (any(is.na(x)) {
    stop ("x contains NA")
  }

  for (val in x) {
    sum <- sum + val
  }
  sum
}
```



sum.R

Removing Branch

Show **Merged** branches

```
git branch --merged
```

Remove branch

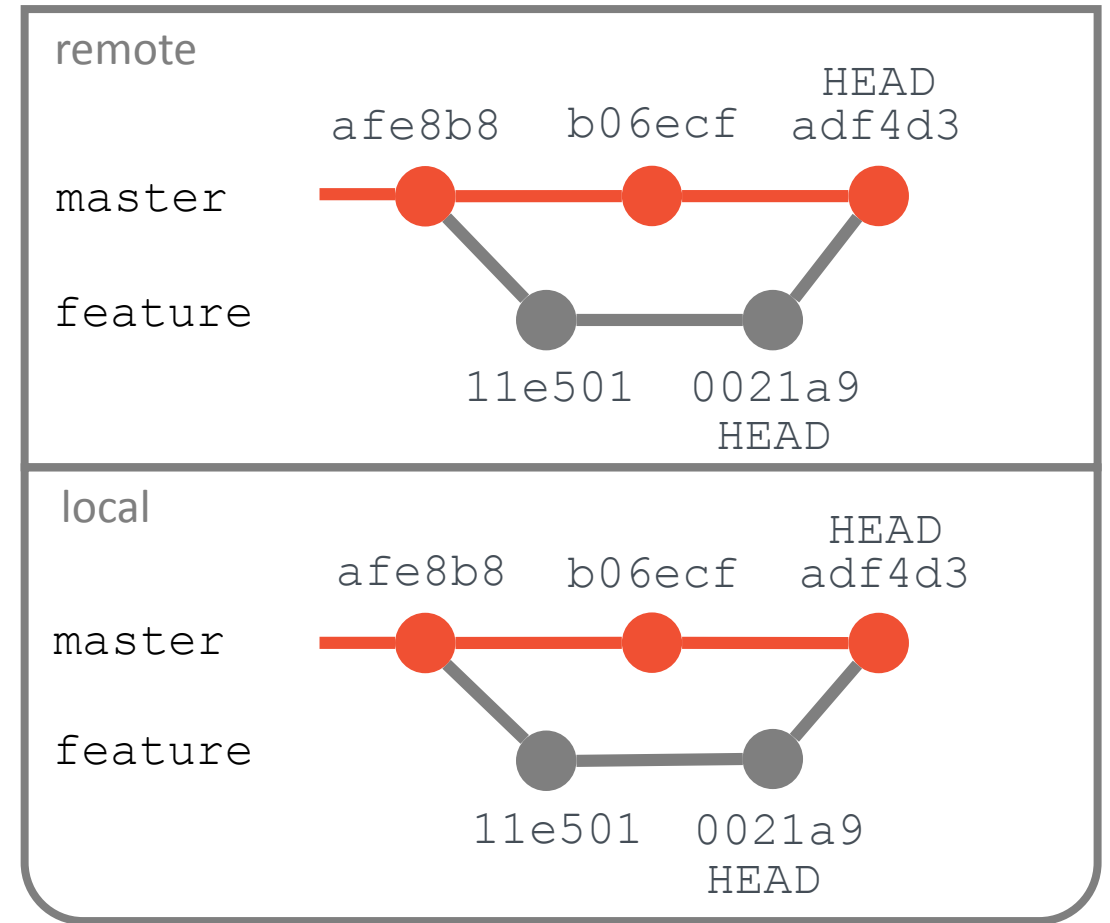
```
git branch -d [branch-name]
```

List **remote** branches

```
git branch -a
```

Remove branch from **remote** repository

```
git push origin -delete [branch-name]
```



Rebase

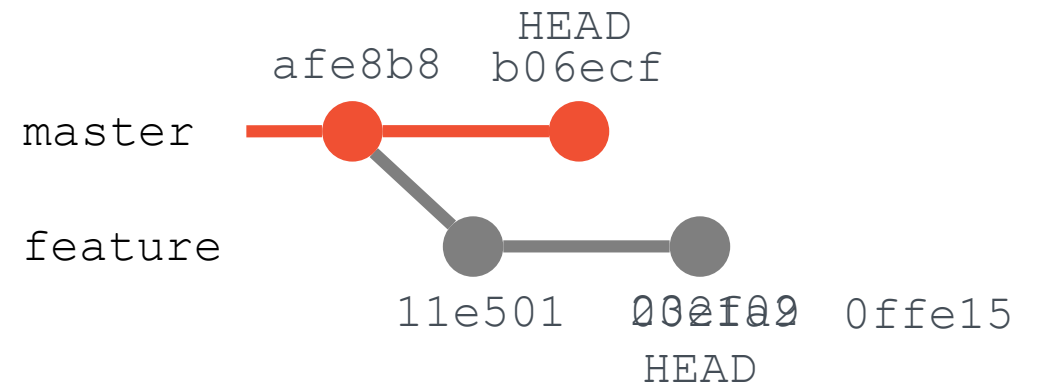
Changes base of a branch from one commit to another. *Moved* branch is composed of entirely new commits.

Rebase branch

```
git checkout [branch-name]
git rebase master
```

Fast-Forward merge

```
git checkout master
git merge [branch-name]
```



Useful

Stage and commit all files

```
git commit -am "[message]"
```

Tag a commit

```
git tag
```

Set **editor** for commit messages - default `vim`

```
git config --global core.editor [editor]
```



`.gitignore`

RStudio GUI

- **Visualize** branches and changes
- **Merging** functionality not available

The screenshot displays the RStudio Git interface. The top panel shows the commit history with a table of commits. The bottom panel shows a diff view for the commit with SHA aa2a4d20.

Subject	Author	Date	SHA
HEAD -> refs/heads/master Resolved merge conflict	Henrik John <l.john@erasmus>	2018-01-29	6417ecb0
Added a new error message	Henrik John <l.john@erasmus>	2018-01-29	aa2a4d20
errorMessage Added error message within branch	Henrik John <l.john@erasmus>	2018-01-29	c815a681
Initial commit.	Henrik John <l.john@erasmus>	2018-01-29	4452b7ee

Commits 1-4 of 4

SHA aa2a4d20
Author Henrik John <l.john@erasmusmc.nl>
Date 2018-01-29 10:14
Subject Added a new error message
Parent 4452b7ee

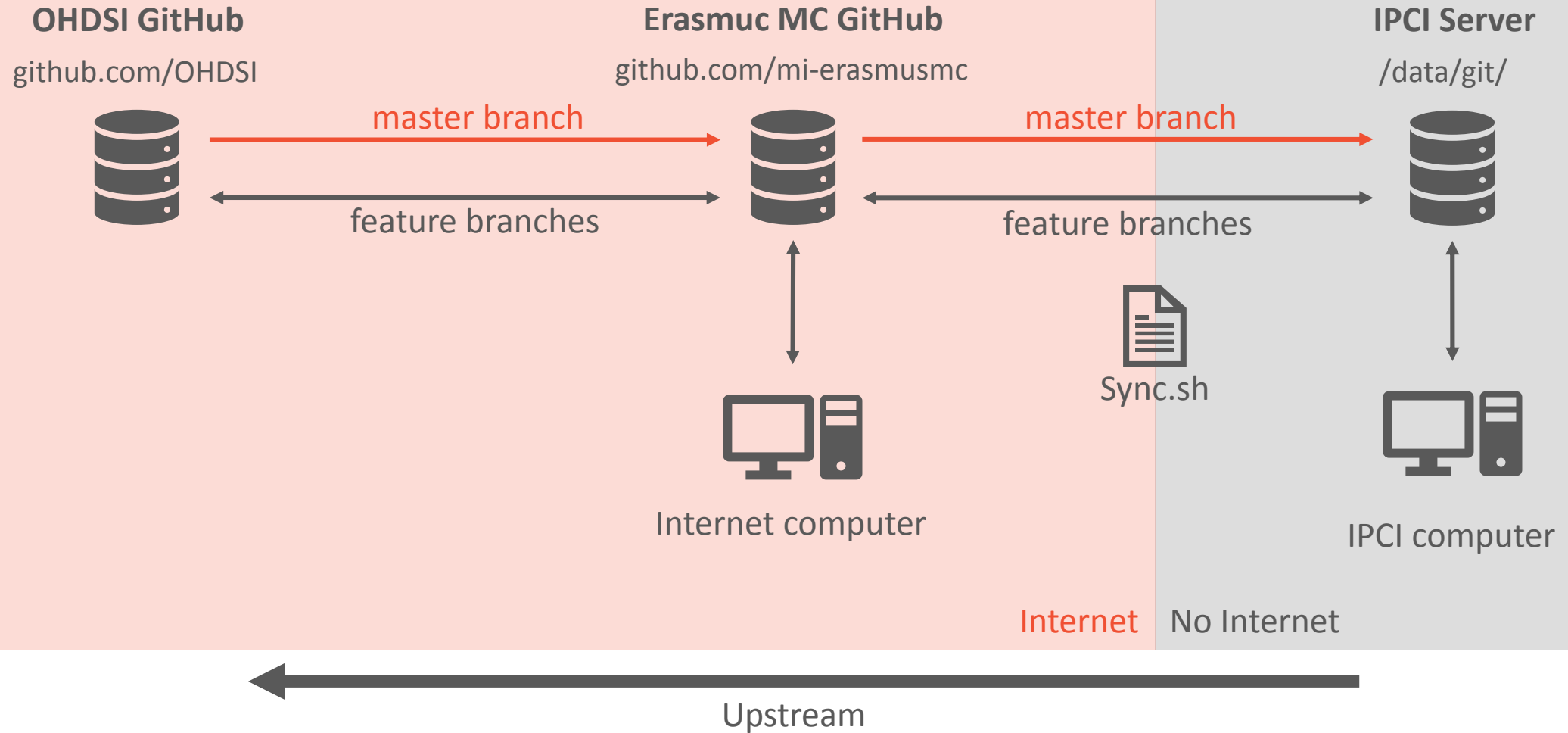
testFunction.R

testFunction.R View file @ aa2a4d20

```
@@ -1,3 +1,3 @@
1 1 test <- function(x) {
2  stop("ToDo")
2  stop("There appears to be an error.")
3 3 }
```

No newline at end of file

Medical Informatics Workflow



Assignment

1. **Clone** the project from <https://github.com/mi-erasmusmc/GitTutorial>
2. Create a new **branch**
3. Within the new branch modify `participants.txt` by adding your first name to the list
4. **Commit** the change to your branch
5. **Merge** your branch into the master
6. **Push** the master to the remote

Documentation

- **Installation** files <https://git-scm.com/downloads>
- Official **documentation** <https://git-scm.com/documentation>
- **DataCamp** course <https://www.datacamp.com/courses/introduction-to-git-for-data-science>
- **Crash** course https://www.youtube.com/watch?v=SWYqp7iY_Tc
- **RStudio** course
 - Part 1: <https://www.datacamp.com/courses/working-with-the-rstudio-ide-part-1>
 - Part 2: <https://www.datacamp.com/courses/working-with-the-rstudio-ide-part-2>