Git/GitHub/GitLab tutorial
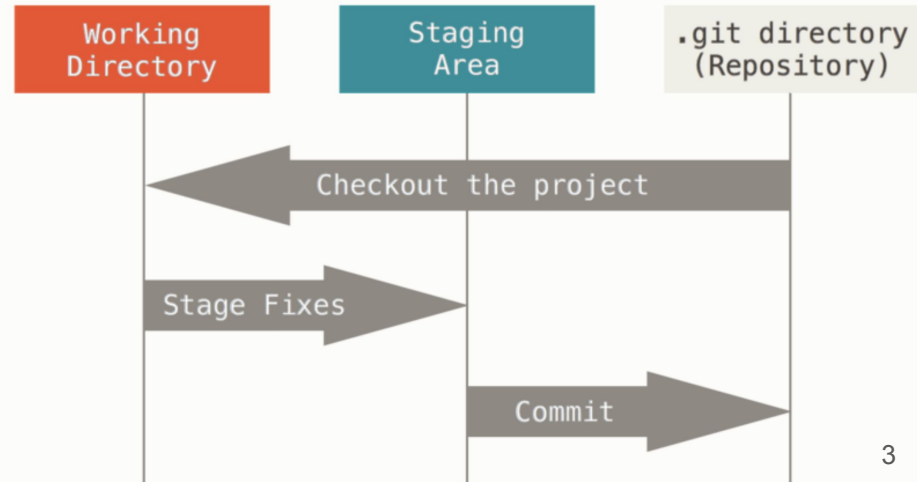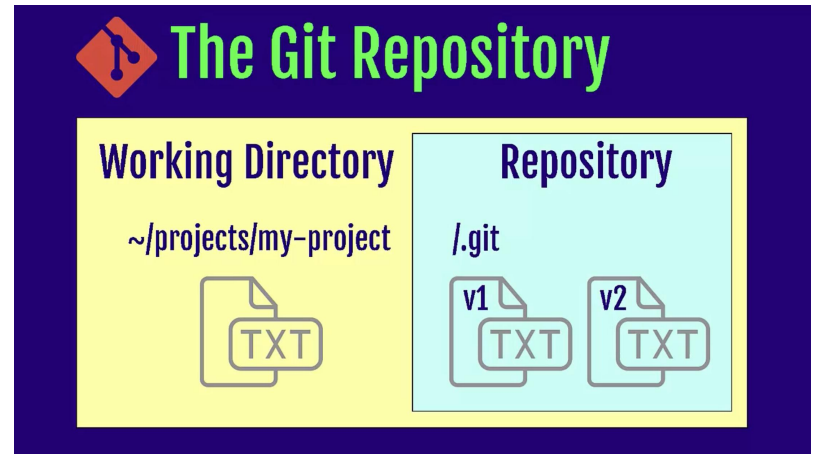
Introdução à análise de dados em FAE

What is Git?

- Git is a Distributed Version Control System (DVCS)
- Invented by Linus Torvalds (creator of Linux)
- Other version control systems you may know:
  - Centralized: Concurrent Versions System (CVS), Subversion (SVN)
  - Distributed: Mercurial (Hg)
  - Copy/paste files and folders on your computer
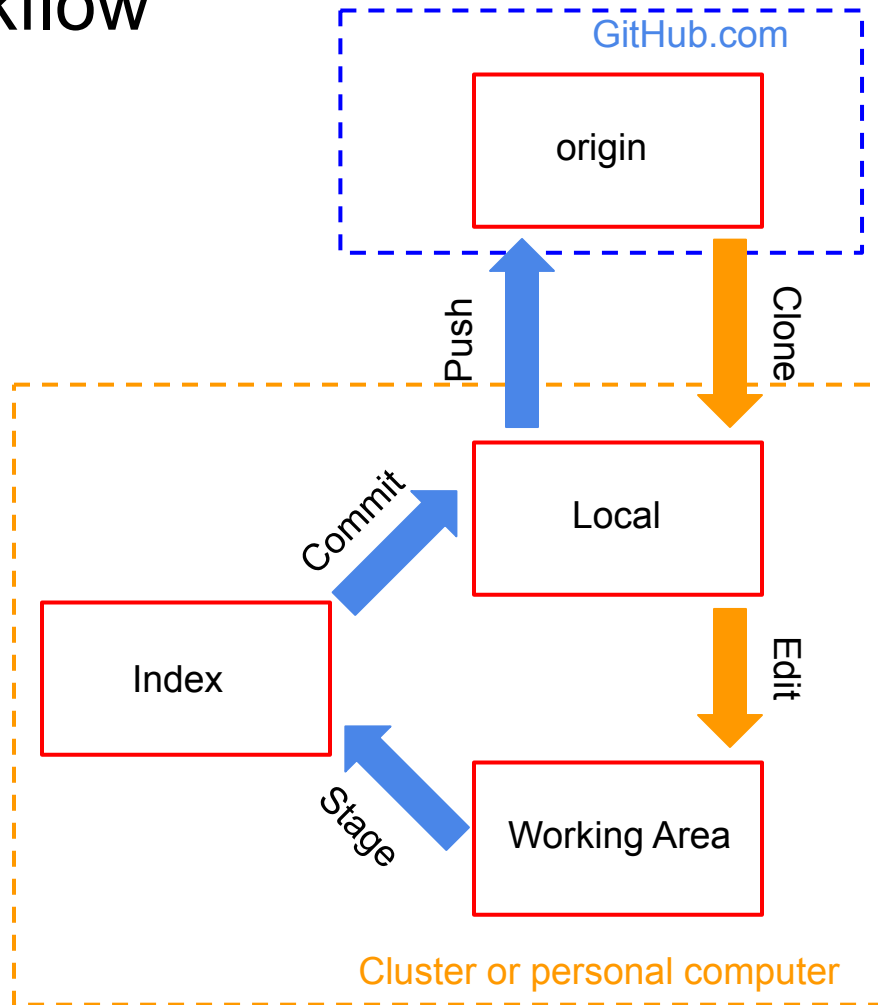
Git workflow

- Repository contains files, history, config managed by Git
- The files can be:
  - Tracked
  - Untracked
  - Staged
  - Committed
- Three States of Git
  - Working directory
  - Staging area - pre-commit holding area:
  - Repository- Git Repository (history)

The Git Repository

Working Directory	Repository
~/projects/my-project	/.git
TXT	v1 TXT v2 TXT

Working Directory — Staging Area — .git directory (Repository)

Checkout the project

Stage Fixes

Commit

Remote Repository Workflow

- Working area: actual source code being edited on your local machine
- Index: an intermediate area between the working area and the local repository
- Local: .git folder on your local machine
- Origin: repo on a remote server

GitHub.com

origin

Push

Clone

Local

Commit

Edit

Index

Stage

Working Area

Cluster or personal computer

4

GitHub Setup

- Make an account:
  https://github.com/join
- Generate an SSH key and add it to your account:
  https://help.github.com/articles/generating-an-ssh-key/

GitHub.com / Authentication / Connecting to GitHub with SSH

Article version:
GitHub.com ⌄

Connecting to GitHub with SSH

You can connect to GitHub using SSH.

About SSH

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username or password at each visit.

Checking for existing SSH keys

Before you generate an SSH key, you can check to see if you have any existing SSH keys.

Generating a new SSH key and adding it to the ssh-agent

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

Adding a new SSH key to your GitHub account

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Testing your SSH connection

After you've set up your SSH key and added it to your GitHub account, you can test your connection.

Working with SSH key passphrases

You can secure your SSH keys and configure an authentication agent so that you won't have to reenter your passphrase every time you use your SSH keys.

Repositories

- Repository: a software project, the unit of development on Git
- Often abbreviated as "repo"
- Repositories may contain:
  - Source code
  - README text files (GitHub uses [Markdown](#) markup languages; [full spec](#))
  - Input files (of various types)
  - `.gitignore`: list of file types (or filenames) to ignore automatically
- **Do not** store large binary files (e.g. ROOT files) in a Git repository! EVER!
  - Size of the repository can explode
  - Operations on the repository can become slow

Make a repository

- Create a new repository from scratch: `git init`
- Copy a repository (e.g. from GitHub): `git clone` git@github.com`:user/repo.git`
- Both of these commands creates a ".git" folder with everything Git needs
- Remember: Every local copy of a repository is fully-featured!
- You can also create a new repository directly on GitHub
- A few clone features:
  - Change directory name: `git clone` git@github.com`:user/foo.git bar`
  - Start on a specific branch: `git clone` git@github.com`:user/foo.git -b new`

Basic Terminology

- **Branch**: development area for a project
  - *One* repository can have *multiple* branches
- **Commit**: a set of changes to a project (modify, add, remove files)
  - A commit is just a *diff* (list of lines changed, added, removed)
  - Each commit is identified by a unique *hash* (SHA-1) based on metadata: *parent*(previous commit), date/time, author, committer, message, etc.
- **Tag**: specific version (snapshot) of a project
- **Release**: GitHub ass-on to a tag (release notes, source code tarball, etc.)
- **Tracked**: Git keeps track of a *tracked* file
- **Untracked**: Git does not keep track of an *untracked* file

Basic Commands: Commits & Files

All commands should be preceded by `git`, e.g. `git add` …

- `commit -a`: make a snapshot of all changes
  - **Always** leave a descriptive commit message with `-m "message"`

File operation commands:

- `add [file]`: start tracking a file
  - `add -A`: track all files
- `rm [file]`: delete a file and stop tracking it
- `mv [old] [new]`: rename a file
- All three of these commands can be used with `-n`, equivalent to `--dry-run`: shows what will happen without actually doing it
- They can also be used with `-f` or `--force`, to override any warnings or problems (e.g. file is ignored in `.gitignore`)

9

Basic Commands: Branches

All commands should be preceded by `git`, e.g. `git checkout …`

- `Checkout [branch/commit]`: switch to specific branch or commit
  - This command affect your current *working area*
  - `checkout -b [branch]`: create a new branch and switch to it
  - `checkout -b [branch] [ref]`: create a new branch to *follow* ref (use ref as starting point for the new branch) (ref can be another local branch, a remote branch, even a commit)
  - `checkout [ref] [file]`: switch to specific file to the version in ref
- `branch`: list local branches
  - This command affects Git's *list of branches*
  - `branch -a`: list all branches (including remote branches)
  - `branch [branch]`: make a new branch
  - `branch [branch] [ref]`: make a new branch to *follow* ref
  - `branch -m [old] [new]`: rename a branch
  - `branch -d [branch]`: delete a branch

Basic Commands: Status

- `status`: see the current state of all uncommitted changes (ignores all file or type specified in `.gitignore`)
  - `-s`: summarize status
  - `-suno`: summarize status (`s`) of *only* tracked files (`uno`)
  - `[file]`: see status of specified file
- `diff`: see the actual changes in *diff* output format
  - Many possibilities: specific file (`[file]`), specific commits (`[commit1] [commit2]`), etc.

Git Setup

In your home directory:

```
.gitconfig
[user]
      name = Sheila Amaral
      email = sheila.silva.do.amaral@cern.ch
      github = ssilvado
```

Commands to set these:

```
git config --global user.name [Name]
git config --global user.email [Email]
git config --global user.github [Account]
```

To see the configuration:

```
git config --list
```

To see the global configuration:

```
cat ~/.gitconfig
```

Create Repository on GitHub

Create a new Repository Locally

```
● ● ●                    📁 hellogit — bash — 93×49
sheilamarass@amaral:~ $ mkdir git-projects
sheilamarass@amaral:~ $ cd git-projects/
sheilamarass@amaral:~/git-projects $ pwd
/Users/sheilamarass/git-projects
sheilamarass@amaral:~/git-projects $ ls
sheilamarass@amaral:~/git-projects $ git init hellogit
Initialized empty Git repository in /Users/sheilamarass/git-projects/hellogit/.git/
sheilamarass@amaral:~/git-projects $ ls
hellogit
sheilamarass@amaral:~/git-projects $ cd hellogit/
sheilamarass@amaral:~/git-projects/hellogit $ ls
sheilamarass@amaral:~/git-projects/hellogit $ ls -al
total 0
drwxr-xr-x   3 sheilamarass  staff  102 31 Mar 22:19 .
drwxr-xr-x   3 sheilamarass  staff  102 31 Mar 22:19 ..
drwxr-xr-x  10 sheilamarass  staff  340 31 Mar 22:19 .git
sheilamarass@amaral:~/git-projects/hellogit $ cd .git/
sheilamarass@amaral:~/git-projects/hellogit/.git $ ls
HEAD        config      hooks       objects
branches    description info        refs
sheilamarass@amaral:~/git-projects/hellogit/.git $ cd ..
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $
```

Create a new repository from scratch:
```
cd git-projects
git init hellogit
```

Check the .git folder
```
cd hellogit
ls .git
```

Check the status
```
git status
```

14

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Create a new file test.txt
`echo "It is just a demo file" >> test.txt`

Check the status
`git status`

Start track the file test.txt
`git add test.txt`
`git status`

Commit the changes to the staging area
`git commit`
`git status`

Check the register
`git log`

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▎
```
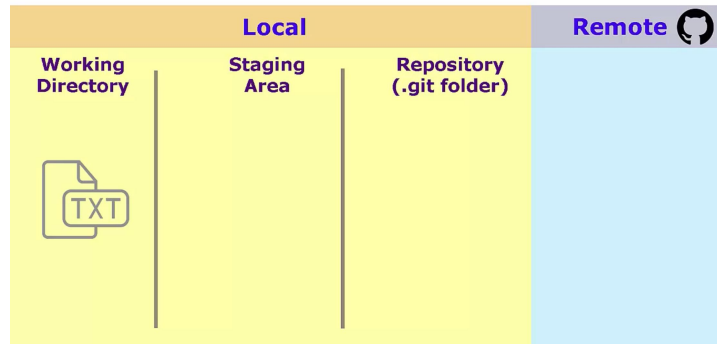
Basic Git Workflow Life Cycle

Local			Remote
Working Directory	**Staging Area**	**Repository (.git folder)**	

Start track the file hipster.txt
`git add test.txt`
`git status`

Commit the changes to the staging area
`git commit`
`git status`

Check the register
`git log`

16

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▮
```

Create a new file test.txt
`echo "It is just a demo file" >> test.txt`

Check the status
`git status`

Start track the file test.txt
`git add test.txt`
`git status`

Commit the changes to the staging area
`git commit`
`git status`

Check the register
`git log`

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ 
```
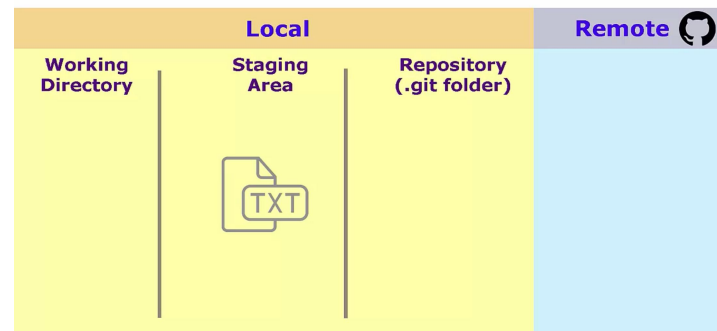
Create a new file test.txt
`echo "It is just a demo file" >> test.txt`

Basic Git Workflow Life Cycle

	Local			Remote
Working Directory	Staging Area	Repository (.git folder)	

area
`git commit`
`git status`

Check the register
`git log`

18

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Create a new file test.txt
`echo "It is just a demo file" >> test.txt`

Check the status
`git status`

Start track the file test.txt
`git add test.txt`
`git status`

Commit the changes to the staging area
`git commit`
`git status`

Check the register
`git log`

Create a new Repository Locally

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```
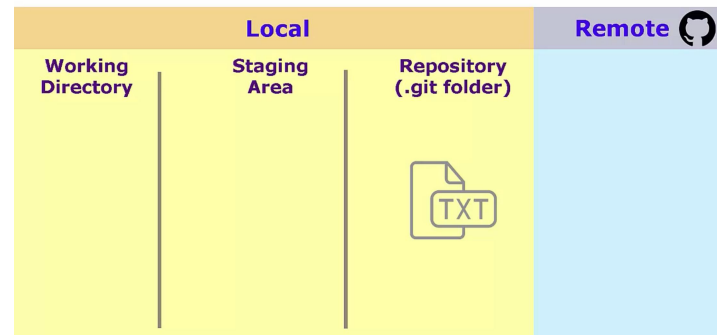
Create a new file test.txt
```
echo "It is just a demo
file" >> test.txt
```

Basic Git Workflow Life Cycle

	Local			Remote
Working Directory	**Staging Area**	**Repository (.git folder)**		
		TXT		

area
```
git commit
git status
```
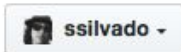
Check the register
```
git log
```

First push

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner   **Repository name** *

ssilvado ▾   /   hellogit   ✓

Great repository names are short and memorable. Need inspiration? How about **sturdy-telegram**?

Description (optional)

[   ]

○ 📖 **Public**
      Anyone can see this repository. You choose who can commit.

○ 🔒 **Private**
      You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
      This will let you immediately clone the repository to your computer.

[Add .gitignore: **None** ▾]   [Add a license: **None** ▾] ⓘ

Create repository

---

ew file   **Upload files**   **Find file**   **Clone or download** ▾

Clone with HTTPS ⓘ   Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/ssilvado/hellogit. 📋

Open in Desktop   **Download ZIP**

First push

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote add origin
https://github.com/ssilvado/hellogit.git
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
origin  https://github.com/ssilvado/hellogit.git (fetch)
origin  https://github.com/ssilvado/hellogit.git (push)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ssilvado/hellogit.git
 * [new branch]      master -> master
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▊
```

Show list of remote repos
`git remote -v`

Add new remote
`git remote add origin`
https://github.com/ssilvad
o/hellogit.git

Show list of remote repos
`git remote -v`

Send local ref (master) to specific remote (origin)
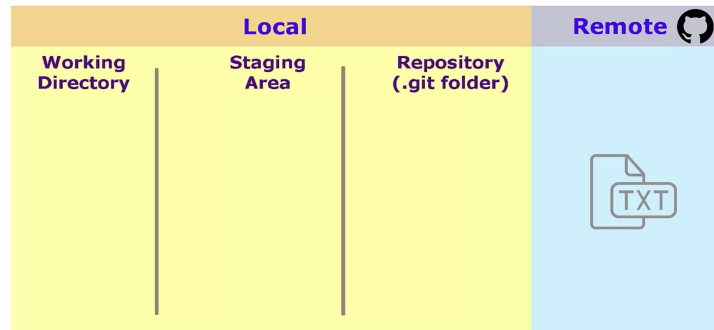`git push origin master`

First push

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote add origin
https://github.com/ssilvado/hellogit.git
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
origin  https://github.com/ssilvado/hellogit.git (fetch)
origin  https://github.com/ssilvado/hellogit.git (push)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ssilvado/hellogit.git
 * [new branch]      master -> master
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▊
```

Show list of remote repos
`git remote -v`

Add new remote

Basic Git Workflow Life Cycle

Local			Remote ⬤
Working Directory	Staging Area	Repository (.git folder)	
			TXT

remote (origin)
`git push origin master`

Branch: master ▾	New pull request			Find file	Clone or download ▾

ssilvado A simple commit		Latest commit 1f67f82 16 minutes ago
☐ test.txt	A simple commit	16 minutes ago

Create a new Branch

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -v
* master 1f67f82 A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch work
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout work
Switched to branch 'work'
sheilamarass@amaral:~/git-projects/hellogit (work) $ git log --all
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
```

Check in which branch we are now
`git branch -v`

Create new branch named work
`git branch work`

Switch to the branch work
`git checkout work`

Display logs of all branches and tag
`git log --all`

Git merge

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout -b testing master
Switched to a new branch 'testing'
sheilamarass@amaral:~/git-projects/hellogit (testing) $ echo "Working with branches and merge" >>
test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git add test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git commit -m "New test"
[testing d144418] New test
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git checkout master
Switched to branch 'master'
sheilamarass@amaral:~/git-projects/hellogit (master) $ git merge testing
Updating c37981c..d144418
Fast-forward
 test-branches-merge.txt | 1 +
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -d testing
Deleted branch testing (was d144418).
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log --graph
* commit d14441841907e1e6de826c163f1be14edb4aea25
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:14:42 2020 -0400
|
|     New test
|
* commit c37981c5ad4899824fa428322e400c8900758df5
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:10:17 2020 -0400
|
|     Start feature
|
* commit 1f67f82ff927fb82fd8050b1059527929edbe447
  Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
  Date:   Tue Mar 31 22:21:53 2020 -0400

      A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ 
```

Start a new feature

`git checkout -b testing master`

Create new file test-branches-merge.txt
and commit change

```
echo "Working with branches
and merge" >>
test-branches-merge.txt
git add
test-branches-merge.txt
git commit -m "New test"
```
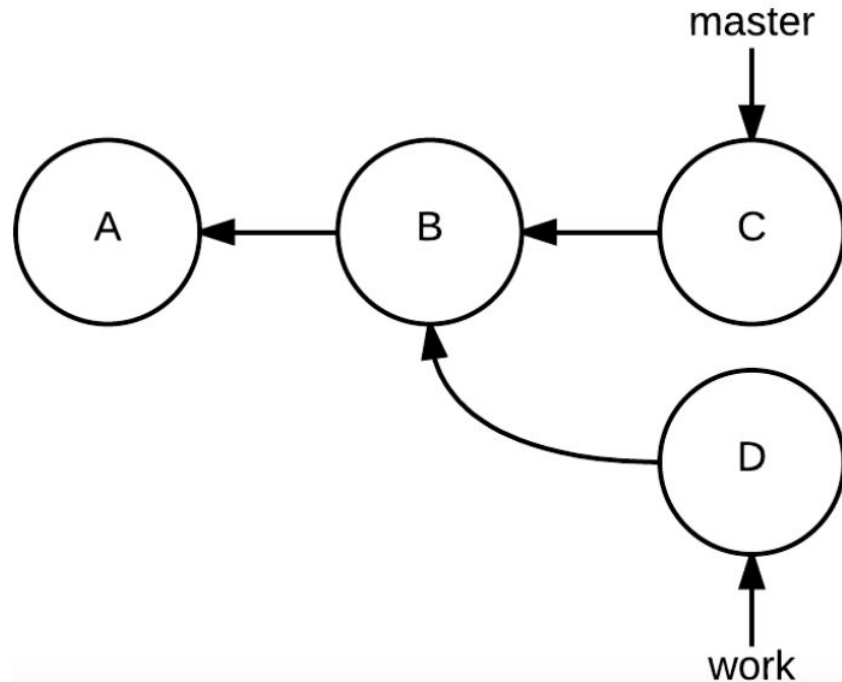
Merge in the testing branch

```
git checkout master
git merge testing
git branch -d testing
```

Git merge

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout -b testing master
Switched to a new branch 'testing'
sheilamarass@amaral:~/git-projects/hellogit (testing) $ echo "Working with branches and merge" >>
test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git add test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git commit -m "New test"
[testing d144418] New test
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git checkout master
Switched to branch 'master'
sheilamarass@amaral:~/git-projects/hellogit (master) $ git merge testing
Updating c37981c..d144418
Fast-forward
 test-branches-merge.txt | 1 +
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -d testing
Deleted branch testing (was d144418).
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log --graph
* commit d14441841907e1e6de826c163f1be14edb4aea25
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:14:42 2020 -0400
|
|     New test
|
* commit c37981c5ad4899824fa428322e400c8900758df5
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:10:17 2020 -0400
|
|     Start feature
|
* commit 1f67f82ff927fb82fd8050b1059527929edbe447
  Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
  Date:   Tue Mar 31 22:21:53 2020 -0400

      A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▎
```

Git merge

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout -b testing master
Switched to a new branch 'testing'
sheilamarass@amaral:~/git-projects/hellogit (testing) $ echo "Working with branches and merge" >>
test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git add test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git commit -m "New test"
[testing d144418] New test
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git checkout master
Switched to branch 'master'
sheilamarass@amaral:~/git-projects/hellogit (master) $ git merge testing
Updating c37981c..d144418
Fast-forward
 test-branches-merge.txt | 1 +
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -d testing
Deleted branch testing (was d144418).
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log --graph
* commit d14441841907e1e6de826c163f1be14edb4aea25
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:14:42 2020 -0400
|
|     New test
|
* commit c37981c5ad4899824fa428322e400c8900758df5
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:10:17 2020 -0400
|
|     Start feature
|
* commit 1f67f82ff927fb82fd8050b1059527929edbe447
  Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
  Date:   Tue Mar 31 22:21:53 2020 -0400

      A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ ▏
```

Start a new feature
```
git checkout -b testing master
```

Create new file test-branches-merge.txt and commit change
```
echo "Working with branches
and merge" >>
test-branches-merge.txt
git add
test-branches-merge.txt
git commit -m "New test"
```

Merge in the testing branch
```
git checkout master
git merge testing
git branch -d testing
```

Collaboration on GitHub

- Common development situation:
  - Group software project (e.g. ntuple production code)
  - A few people may be "in charge" - release managers
  - Many people may contribute code - objects, algorithms, configurations, etc.
- Several options:
  - Everyone commits to the same repository ❌
  - Each contributor **forks** the repo and sends **pull requests** with new feature ✓ ⇒"Fork and Pull development model
- What is a Pull Request?

Pull requests are a feature specific to GitHub. They provide a simple, web-based way to submit your work (often **called** "patches") to a ("upstream") project. It's **called a pull request** because you're asking the project to **pull** changes from your fork.
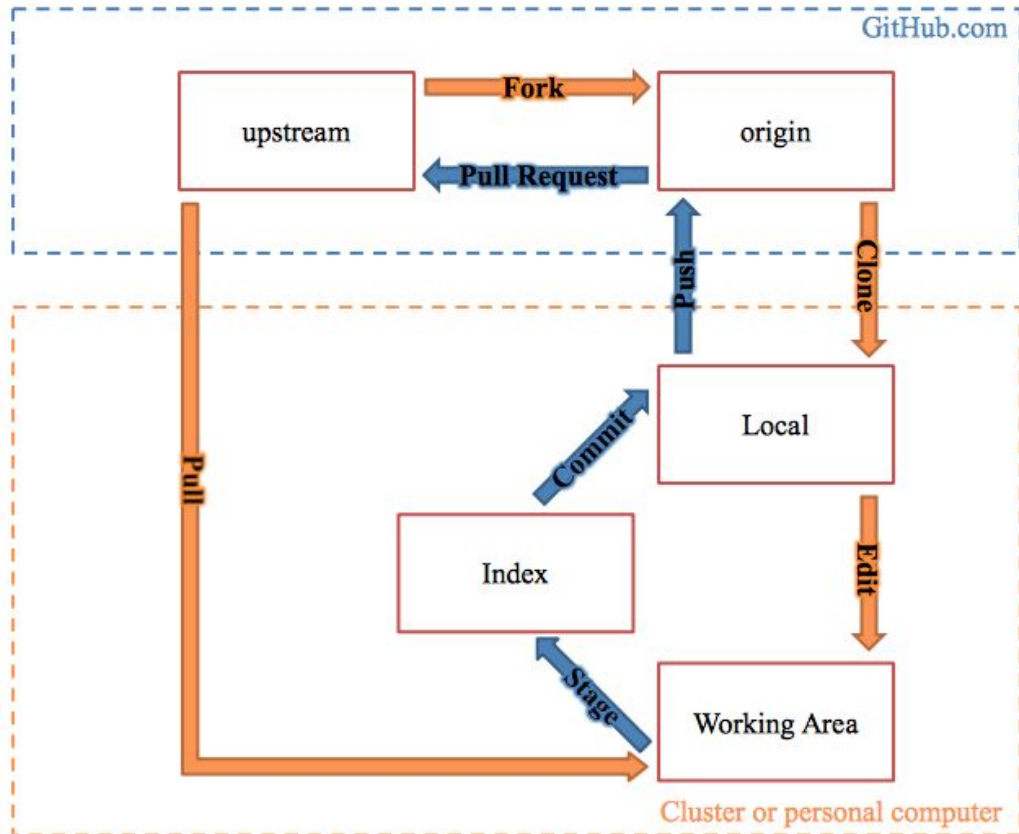
"Fork and Pull" workflow

1. Fork repository: this creates a new copy of the repo under your GitHub user account
2. Clone the repository to your local computer: `git clone` https://github.com/useraccount/demo
3. Create a new branch by issuing the command: `git checkout -b new_branch`
4. Create a new remote for the upstream repo with the command: `git remote add upstream` https://github.com/useraccount/demo
   a. In this case, "upstream repo" refers to the original repo you created your fork from.
5. Now you can make change to the code.
   a. E.g.: Make any change `(echo "some test file" >> test ) >` `git status > git add test > git commit -m "Adding a test file to new_branch"`
6. Push the changes to the remote: `git push -u origin new_branch`
7. Once you push the changes to your repo, the Compare & pull request button will appear in GitHub.

Undo in Git

It depends on the stage of development

- ● Undo local changes
  - ○ Unstaged local changes (before `git add` and `git commit`): `git checkout .`
  - ○ Stages local changes (before `git commit`): `git reset`
  - ○ Committed local changes
    - ■ Without modifying history: `git revert HEAD`
    - ■ With history modification: (*)
  - ○ Redoing the Undo (*)
- ● Undo remote changes
  - ○ without changing history (*)
  - ○ with history modification (*)

(*) check https://docs.gitlab.com/ee/topics/git/numerous_undo_possibilities_in_git/

Exercício

- I created a web project* on my GitHub account and you should change any file and submit a Pull Request, based on the "Fork and Pull" method
- The repository is: https://github.com/ssilvado/web-project
- It should be done until April 16th, 2020.

*This HTML5 project was created using http://www.initializr.com/, which is a template generator