Git Basics Workshop
CodeAcademy

IBM

Who are we?

- Diego Tabares
  Web Developer
  +10 years in IBM

- Nicolás Gomez
  Web Developer
  +10 years in IBM

- Pablo Hernandez
  Web Developer
  9 years in IBM

Agenda

- What is Version Control?
- What is Git?
- How does Git work?
- Git workflow
- Configuring git
- Cloning an existing repo
- Creating a repo
- First Steps
- Gitignore
- Git Log
- Branching and Merging
- Solving merge conflicts
- Ooops I fu**ed it up

What is Version Control?

- System that records changes to a file or group of files
- Revert to a previous state
- Track and compare changes
- Types: Local , Centralized and Distributed

What is Git?

- Open source project
- Distributed Version Control System
- Performance
- Security
- Flexibility

How does Git work?

- Track differences
- Nearly every operation are Local
- Integrity control
- Files stages

Git workflow

- New project
  - Create a new repository
  - Create the necessary branches
  - Setup of the project permissions (if using GitHub for example)

- Existing project
  - Checkout a repository
  - Create a local branch
  - Make the changes
  - Add the required files
  - Merge the local branches
  - Push the changes

Configuring git

- Create ssh key
  - $ mkdir /home/student/.ssh
  - $ cd /home/student/.ssh
  - $ ssh-keygen -t rsa -C "<mail>"
  - $ chmod 400 id_rsa
  - Copy the pub key content to GitHub
- Identity
  - $ git config --global user.name "John Doe"
  - $ git config --global user.email johndoe@example.com
- Editor
  - $ git config --global core.editor vi
- Colors
  - $ git config --global color.ui true

Clonning an existing repo

- $ git clone git@github.com/user/repo.git

- $ git clone <source> <destination>

Creating a repo

- Create the remote repository in Github
- cd to the project's folder
- git init
- git remote add origin <URI>

First Steps

- $ git status: tracked, untracked, ignored
- Gitignore
- $ git add <file> , git add –u <file>
- $git checkout <file>
- $ git commit –am “”
- $ git pull
- $ git push

Gitignore

- Syntax
  - /db/*.sqlite3

- Common uses
  - compiled code
  - build output directories, such as /bin, /out, or /target
  - personal IDE config files, such as .idea/workspace.xml

Git Logs

- $ git log
- $ git log -p -N
  - Show the changes introduced in the last N commits with their changes (p)
- $ git log --pretty=oneline
  - Show each commit on a single line
- $ git log --pretty=format:"%h - %an, %ar : %s”
  - Customize output by using --pretty=format
- $ git log --since=2.weeks
  - Limit log output

Branching and Merging

- **$ git branch**
  - List local branches
- **$ git branch -r**
  - List remote branches
- **$ git branch -a**
  - List ALL branches
- **$ git fetch**
  - Update the remote branches list
- **$ git checkout -b <new_branch>**
  - Create the local branch
- **$ git checkout <branch>**
  - Change the branch
- **$ git merge <branch>**
  - Merges with the other branch
- **$ git branch -d <branch_to_delete>**
  - Delete a local branch

Solving merge conflicts

- $ git status

On branch branch-b

You have unmerged paths.

(fix conflicts and run "git commit")

- ### Markups ###
- > <<<<<<< HEAD
- > <local version>
- > =======
- > <remote version>
- > ->>>>>>> COMMIT-ID

Ooops I fu**ed it up

- $git checkout <commit id>
  - Return to certain commit
- $git checkout <branch_name>
  - Return to the latest commit
- $git reset --hard <commit id>
  - Return to certain commit (removing the next ones)
- $ git checkout .
  - Returns to the original state

Muchas gracias

CodeAcademy
T&I Dev Team