

Basic Tools for System Management

Frederic.Mallet@unice.fr

Université Nice Sophia Antipolis
Membre de Université Côte d'Azur

Objectives

- Learn how to manage your own computer
 - Using the shell
 - Linux / Windows
 - Version control and git
 - Installing a Virtual Machine
 - Linux Mint 18.4
 - Using bash
 - Eclipse: JDT and Modeling

Continuous Improvements

- Continuous delivery of small batches
 - Fix bug, add feature
- Commits
 - Each commit contains all the project
 - Store only the difference with the previous commit
- History: sequence of commits, can go and back
- Branches
 - Can create separate independent branches
 - The master branch does not know about other branches
 - Merge can be done by pull requests (to be approved)

Version Control

- Complete history of the project available at any time
- Version Control Helps to collaborate
- Allows agility
 - Can undo whatever modification
- Git
 - Good for text-based files
 - Automated Tests, Documentation, web sites

Distributed Version Control

- A remote repository is stored in a data-center or cloud
- Each user has a local project history in his local repository
 - The local repository contains all the project
 - The user can continue working offline
 - This needs synchronization between local repositories and remote one

Git repository

- It contains a series of snapshots (called commits)
 - A new commit can be because of adding a file and/or modifying one
 - The user can go back to any of the snapshot at any time
- Installation
 - `sudo apt install git`

git [command] [--flags] [arguments]

■ Commands

- Status: what has been modified or add
 - **git status -short**
- Add: add a new file
 - **git add file.txt**
- Help
 - **git help**
 - **git help status**

git config

- Specify username and email used to see who has done what on the repository
 - `git config --global user.name "Frederic"`
 - `git config --global user.email "fmallet@unice.fr"`
- Configuration can be
 - Global: Your repositories only
 - Local: Only the local repository
 - System: every repository of every user
 - `local >> global >> system`
- Read: by forgetting the value (last parameter)

Git locations

- Project directory: contains the working tree and .git
 - Working Tree
 - Location on your computer that contains the directories and files
 - Staging Area / index
 - List of files planned to be included in the next commit (commit needs to be meaningful)
 - Local repository
 - All of the commits (version history)
- Remote Repository
 - Located on a remote computer

Create a local repository

```
~$ mkdir repos  
~$ cd repos  
repos$ mkdir myproj  
repos$ cd myproj  
repos/myproj$ git init  
repos/myproj$ ls -a
```

Commit to a local repository

- View status of files in the working tree and staging area
 - Tells you what has been modified and on which branch you are

```
fmallet@chevalerios ~/git/myProj $ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
fmallet@chevalerios ~/git/myProj $ touch file1.txt
fmallet@chevalerios ~/git/myProj $ git status
On branch master

Initial commit

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

        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Working Tree to Staging Area

- `git add <file-or-directory>`

```
fmallet@chevalerios ~/git/myProj $ git add file1.txt
fmallet@chevalerios ~/git/myProj $ git status
On branch master

Initial commit
(at will be committed)

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

        new file:   file1.txt
```

Working Tree to Staging Area

■ git add <file-or-directory>

```
fmallet@chevalerios ~/git/myProj $ git status
On branch master

Initial commit

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

        new file:   file1.txt

fmallet@chevalerios ~/git/myProj $ git status -s
A file1.txt
fmallet@chevalerios ~/git/myProj $ mkdir dir
fmallet@chevalerios ~/git/myProj $ touch dir/f1.txt
fmallet@chevalerios ~/git/myProj $ touch dir/f2.txt
fmallet@chevalerios ~/git/myProj $ git status -s
A file1.txt
?? dir/
fmallet@chevalerios ~/git/myProj $ git add dir
fmallet@chevalerios ~/git/myProj $ git status -s
A dir/f1.txt
A dir/f2.txt
A file1.txt
```

Modified files

- git add <file-or-directory>

```
fmallet@chevalerios ~/git/myProj $ echo "test" >> dir/f1.txt
fmallet@chevalerios ~/git/myProj $ git status -s
AM dir/f1.txt
A  dir/f2.txt
A  file1.txt
fmallet@chevalerios ~/git/myProj $ git add dir/f1.txt
fmallet@chevalerios ~/git/myProj $ git status -s
A  dir/f1.txt
A  dir/f2.txt
A  file1.txt
```

Staging area to local repository

- git commit
- git commit -m “short message”

```
fallet@chevaleros ~/git/myProj $ git commit -m "initial commit"
[master (root-commit) 8b9a96c] initial commit
3 files changed, 1 insertion(+)
create mode 100644 dir/f1.txt
create mode 100644 dir/f2.txt
create mode 100644 file1.txt
fallet@chevaleros ~/git/myProj $ git status -s
fallet@chevaleros ~/git/myProj $
```

Accessing the logs

- You can follow the commits by
 - git log

```
fmallet@chevaleries ~/git/myProj $ git log
commit 8b9a96ca096551f68c9003cdd04072a52636ecae
Author: Frederic Mallet <Frederic.Mallet@inria.fr>
Date:   Mon Sep 10 20:41:41 2018 +0200

    initial commit
fmallet@chevaleries ~/git/myProj $ git log --oneline
8b9a96c initial commit
fmallet@chevaleries ~/git/myProj $
```


Creating a remote repository

- Host in a data center or a cloud
 - <http://www.github.com>
 - <http://www.bitbucket.org>
 - <https://bitbucket.org/<login>/myproj.git>

Push to a remote repository

■ git clone

- If you do not have a local repository
- `git clone url/<login>/repo.git [localname]`
- `git remote -v` displays info about remote repository
- Alias name is “origin”

■ git remote

- If you do have a local repository
- `git remote add <name> <url>`
- Ex: `git remote add origin http://bitbucket.org/...git`

Cloning a remote repository

■ git clone



Local to Remote repositories

- All commits belong to a branch
- By default, a single branch called **master**
- **git push [-u] [repository] [branch]**
 - Commit the branch from local to remote
 - Repository can be shortcut (like origin) or url
 - Use -u the first time
 - Track origin to master
 - May need user name and password