Corso Git & GitHub - Lezione 1

Introduzione a Git e Github

Stefano Faccio, Elisabetta Ferri, Giorgio Micaglio

Associazione Italiana Studenti di Fisica
Comitato Locale di Trento

02/04/2025

Outline del corso

La principale fonte che abbiamo usato è l'ottimo libro Pro Git, 2nd edition di Scott Chacon e Ben Straub, disponibile gratuitamente in lingua inglese al seguente link: `https://git-scm.com/book/en/v2`

- Lezione 1: cos'è Git, installazione e primi passi.
- Lezione 2: utilizzo di Git in locale
- Lezione 3: utilizzo di Git in remoto tramite GitHub e collaborazione

Ci teniamo a ringraziare anche Gianmarco Puleo e Michele Tognoni per il contributo alla realizzazione del corso e in particolare di queste slide.

Overview

Che cos’è Git?

Version Control Systems

- Git è un sistema che tiene traccia di tutte le modifiche che vengono effettuate ad un insieme di files detto **Repository**. Un tale sistema è chiamato *VCS* ossia *Version Control System*.
- L'analogia corretta è quella di una **"macchina del tempo"**: si può ritornare **facilmente** a qualsiasi versione precedente del progetto, senza perdere dati.
- Git è il *de facto standard* per lo sviluppo di progetti software grandi e piccoli sia a livello industriale che universitario.

Git in dettaglio

Idea di base

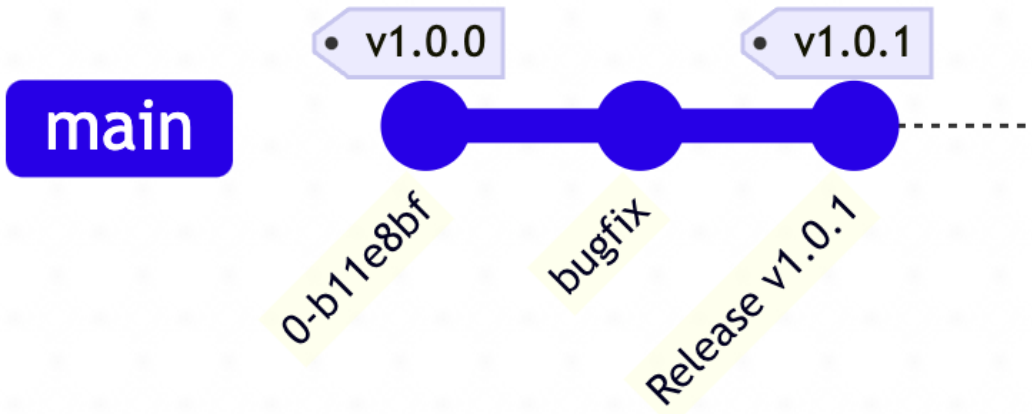
È come fare delle fotografie a dei file in una cartella, con la possibilità di visualizzare tutte le fotografie del passato in ogni momento.

Commit

"**commit**" è sia l'atto di "fotografare" dei file (verbo) che la foto stessa (sostantivo). Ogni commit è identificato con un numero esadecimale univoco, esempio:

```
3c552345a5613a94e5f4704a8311919071fc410d
```

Idea di base

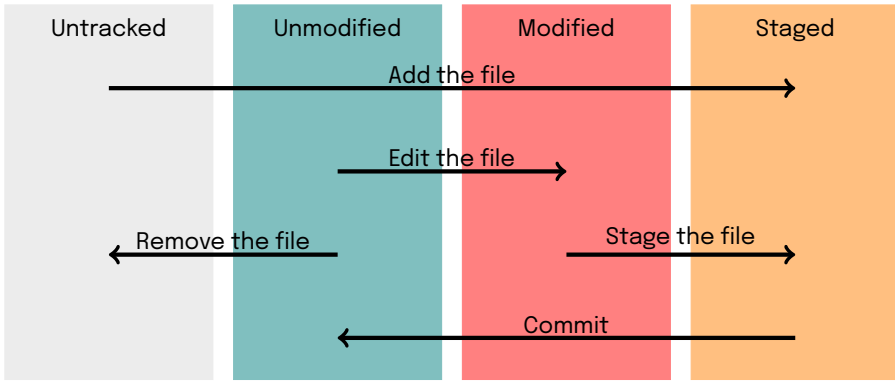


I possibili stati di un file

Un file in una repo può trovarsi in 4 diversi stati:

- **untracked**: è lo stato di partenza di tutti i nuovi file che create. Un file untracked viene completamente ignorato da Git.
- **staged**: significa che il file è identificato da Git come “pronto per essere fotografato” con un *commit*
- **commited**: significa che il file è stato “fotografato” (dunque è tracciato da Git), e dopo l’ultimo commit non è stato modificato.
- **modified**: significa che il file è stato “fotografato” (dunque è tracciato da Git), ma dopo l’ultimo commit è stato modificato.

I possibili stati di un file



Repository Git

Un progetto Git è composto da due elementi:

1. **working directory**: la versione del vostro lavoro su cui state lavorando, ovvero l'insieme di file visibili nella cartella in cui lavorate.
2. **.Git directory**: è una cartella *NASCOSTA* in cui Git opera e salva tutte le versioni del vostro progetto. Contiene tutte le informazioni della vostra repository. NON MODIFICARE I FILE IN QUESTA CARTELLA

Si noti che

Git è generalmente usato assieme ad altri strumenti come *GitHub* per sincronizzare la Repository su un **server remoto** per facilitare la **collaborazione contemporanea** di più utenti allo stesso progetto.

GitHub

1. GitHub è una piattaforma che **si serve** di Git e permette di ospitare le proprie repository su server remoti.
2. Facilita le interazioni tra utenti
3. Ha un'interfaccia user-friendly di facile comprensione ed utilizzo

Nota bene

Useremo GitHub per configurare le repository remote nella lezione 3.

Esempi

Git + GitHub sono usati in collaborazioni scientifiche e aziende, che rendono disponibili anche diversi codici open-source:

- CERN: `https://github.com/CERN`
- Meta: `https://github.com/facebook`
- Python: `https://github.com/python`
- AISF: `https://github.com/ai-sf/`

Fun Fact

Git è tracciato da Git: `https://github.com/git`

Prerequisiti e installazione

Prerequisiti

Useremo la **shell UNIX**

Linux e Mac

Le distribuzioni **Linux e Mac** sono basati su UNIX, non c'è bisogno di fare nulla

Windows 10 / 11

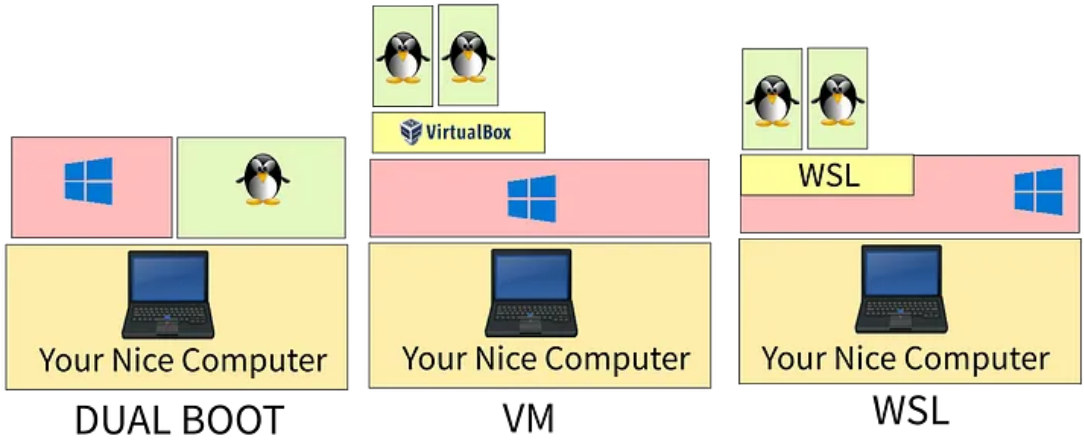
Su **Windows 10 / 11** chiediamo di installare la **WSL - Windows Subsystem for Linux** seguendo questa guida:
`https://learn.microsoft.com/en-us/windows/wsl/install`

Installazione di Git

Installazione di Git da terminale:

- Su Windows WSL o Ubuntu:
  - `sudo apt update`
  - `sudo apt install git-all -y`
- Su Mac:
  `git --version`, se Git non è presente vi dira di installarlo.

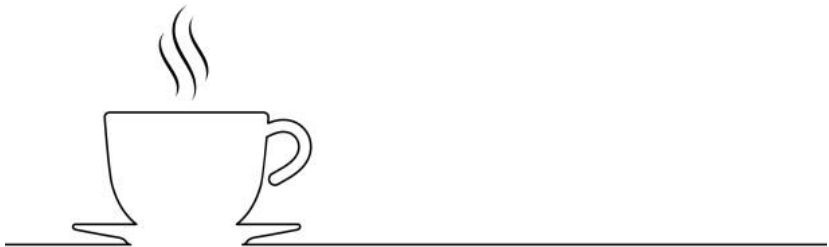
Approfondimento: Che cos'è la WSL?



Approfondimento: Che cos'è la WSL?

- **Dual Boot**: al momento dell'avvio del pc, è possibile scegliere tra più sistemi operativi installati; i diversi *SO* (Sitemi Operativi) possono essere avviati **solo alternativamente**.
- **VirtualBox**: Software di virtualizzazione completa che permette di eseguire sistemi operativi completi (Windows, Linux, macOS, ecc.) in una *VM* (macchina virtuale)
  - Emula un'intera macchina con CPU, RAM, disco, ecc
  - Puoi eseguire qualsiasi SO, non solo Linux
  - Utilizza virtualizzazione più pesante e meno efficiente rispetto a WSL2
- **WSL2**: Essenzialmente una VM leggera che esegue un kernel Linux all'interno di Windows. Microsoft ha **integrato profondamente** WSL2 dentro Windows rendendolo molto più veloce delle VM tradizionali.

Pausa caffè



Prendiamo confidenza con la shell UNIX

Cheat sheet sui comandi della shell di Linux essenziali

`https://ai-sf.it/trento/downloads/git/unix_essential.pdf`

Editor testo da terminale

Su tutti i terminali bash sono presenti vari editor di testo da terminale. Questi tools risualtano spesso comodi per fare veloci modifiche ad un file. Alcuni dei più famosi sono:

- `vim`
  Difficile da utilizzare
- `nano`
  Già installato con Ubuntu
- `gedit`
  Installazione: `sudo apt install gedit`

Si noti che

E' sempre possibile utilizzare un editor di testo con interfaccia grafica come VSCode

Esercizi

https://ai-sf.it/trento/downloads/git/es1.pdf

Configurazione di Git + GitHub

Creazione di un account

D'ora in avanti useremo GitHub.

1. Se non lo avete già, create un account: `www.github.com/signup`.
2. Usate il vostro indirizzo email @unitn o personale.

L'account serve per caricare le proprie repository su un server. Lo vedremo nella lezione 3 in dettaglio.

Configurazione di Git

Ora ci concentriamo sulla configurazione locale. Git dovrà “parlare” con il server GitHub, perciò dobbiamo dire a Git chi siamo: digitare

```
git config --global user.name ilvostrousernamegithub
git config --global user.email lavostraemail@esempio.com
```

Senza l’opzione `--global`, questi dati verranno settati solo per la repository su cui state lavorando.

Si noti che

Non è permesso autenticarsi mediante password. Per accertarsi che siamo davvero noi GitHub usa le **chiavi SSH**.

Approfondimento: Chiavi SSH

Le chiavi SSH utilizzano la **crittografia asimmetrica** per consentire l'autenticazione sicura tra un client e un server.
Nella crittografia asimmetrica sono presenti 2 chiavi:

- **Chiave Privata** mantenuta segreta sul computer dell'utente
- **Chiave Pubblica** condivisa con il server

Perchè ci complichiamo la vita?

Rispetto all'autenticazione basata su password, le chiavi SSH sono più sicure in quanto **non vengono trasmesse mediante la rete informazioni riservate**

Generazione di una coppia di chiavi SSH

Per generare una coppia di chiavi ssh utilizzare i seguenti comandi:

- `cd .ssh` (oppure `cd ~/.ssh`)
- `ssh-keygen -t ed25519 -f nomechiave -C your@email.com`

Quando richiesta una passphrase, lasciare vuoto premendo invio.

Una volta create, usando il comando `ls` vedrete le due chiavi:
`nomechiave.pub` ← chiave pubblica
`nomechiave` ← chiave privata

Caricare la chiave pubblica su GitHub

E' ora il momento di caricare la nostra chiave pubblica appena generata su GitHub

1. Aprite la vostra chiave pubblica con un editor testo o con il comando `cat` e copiate l'intero contenuto.
2. Entrate su GitHub, click in alto a destra → **Settings** → **SSH and GPG keys** → **New SSH key**.
3. Incollate l'intero contenuto della chiave pubblica nel riquadro. Inserite un titolo per identificare la chiave e confermate.

Ora siete pronti per usare Git anche in remoto! Riprenderemo questi concetti nella terza lezione.

Thank you for your attention

Stefano Faccio, Elisabetta Ferri, Giorgio Micaglio

Associazione Italiana Studenti di Fisica
Comitato Locale di Trento

02/04/2025