



Introduzione a Git e GitHub

Progetto ICT Forma

Istituto Statale di Istruzione Superiore "Gobetti - Volta"



Istituto Tecnico
Informatica e Telecom
GobettiVolta



Ciao!

Massimiliano Atzori

- programmatore dal 1999
- lavoro in Develer
- mi occupo di progetti software e sviluppo Agile

Come contattarmi

- social [@amaxis](#) (Twitter)
- email massimiliano.atzori@gmail.com



Cosa faremo

1. Installiamo Git e creiamo un account su GitHub
2. Breve introduzione a Git
3. Primi passi con Git: clone, add, remove, commit
4. Portare le modifiche in remoto
5. Conflitti e merge
6. Riepilogo e consigli utili



Photo by Brendan Steeves on Unsplash

Cosa portare a casa

1. Capire che i programmatori non lavorano **mai** da soli
2. Scoprire che esistono Git e GitHub
3. Imparare i comandi base più semplici
4. Iniziare a studiare i concetti più avanzati
5. Collaborare ad un progetto open source



1. Installare Git e account su GitHub



Installare Git sulla propria macchina

Installare Git

Windows <https://git-scm.com/download/win>

MacOS <https://git-scm.com/download/mac>

Linux

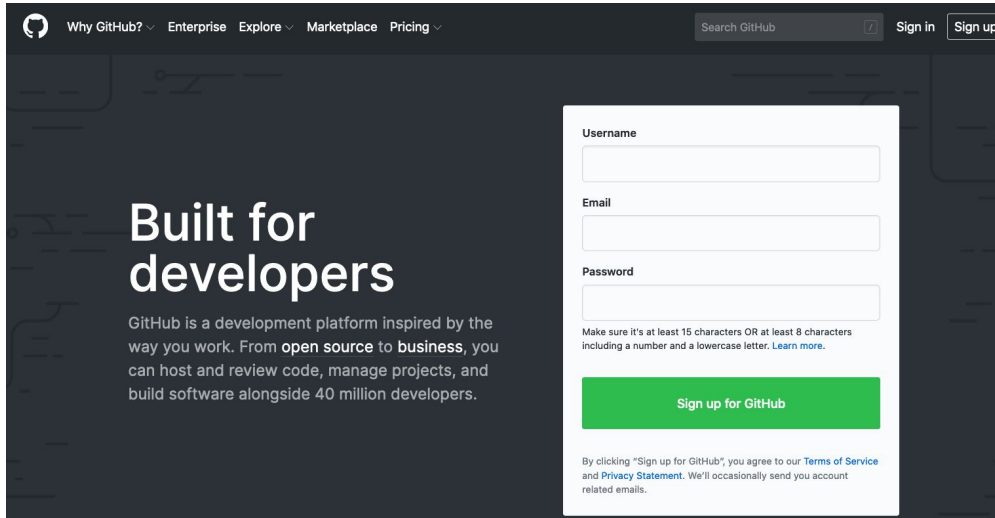
```
sudo apt-get install git
```

Debian

```
sudo yum install git
```

Fedora

2 - Account su GitHub



The image shows the GitHub sign-up page. At the top, there is a navigation bar with links: 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar and 'Sign in'/'Sign up' buttons are also present. The main content area has a dark background with the text 'Built for developers' and a description of GitHub as a development platform. On the right, there is a white sign-up form with fields for 'Username', 'Email', and 'Password'. Below the password field, there is a note about password requirements and a link to 'Learn more'. A green 'Sign up for GitHub' button is at the bottom of the form. At the very bottom, there is a small disclaimer about agreeing to the 'Terms of Service' and 'Privacy Statement'.

Why GitHub? Enterprise Explore Marketplace Pricing

Search GitHub Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

Cos'è GitHub

Account su GitHub

Gratuito per progetti Open Source

<https://github.com/>

<https://github.com/amaxis/git-gobetti-volta>

2. Breve introduzione a Git

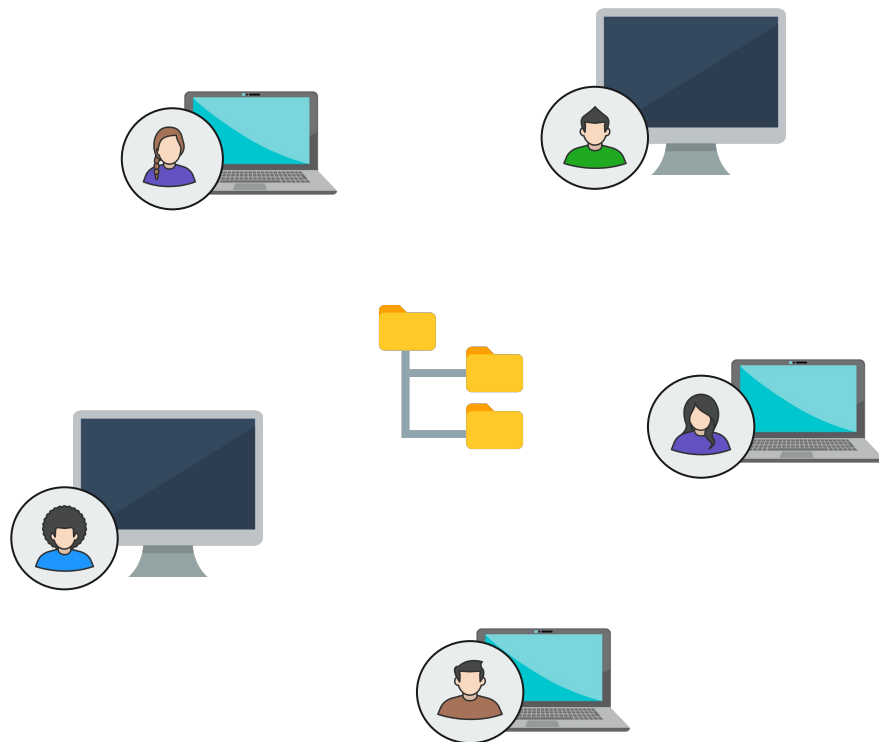
Cos'è Git?

È un **software di controllo di versione (VCS)**

Serve a:

- Riportare i file ad uno stato precedente
- Riportare un progetto ad uno stato precedente
- Confrontare modifiche fatte nel tempo
- Capire chi ha modificato cosa

In poche parole, è un insieme di strumenti per evitare di fare danni con il codice



Come nasce Git?

Linus Torvalds, Aprile 2005 (4 giorni!)

Junio Hamano, Luglio 2005 - oggi

Prende il nome da una canzone dei Beatles
(I'm so tired)

GitHub nasce nel 2008

Viene acquisito da Microsoft nel 2018



Linus Torvalds

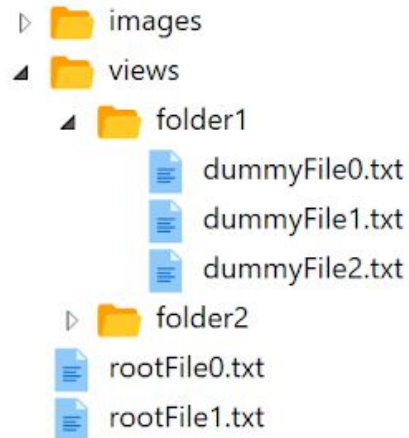


Junio Hamano

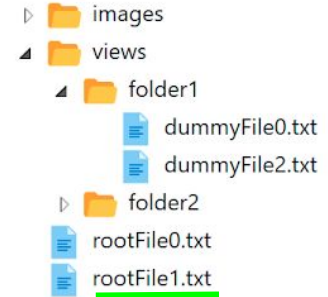
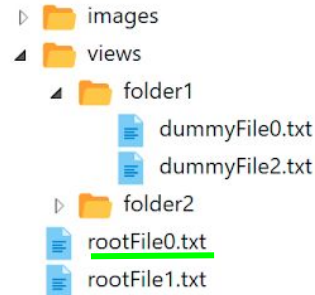
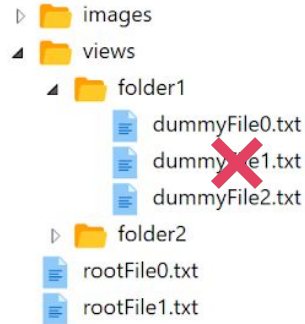
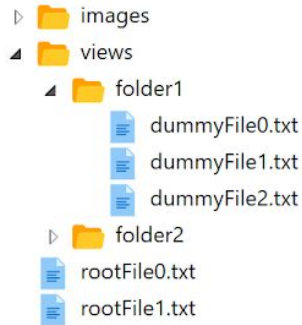
Repository



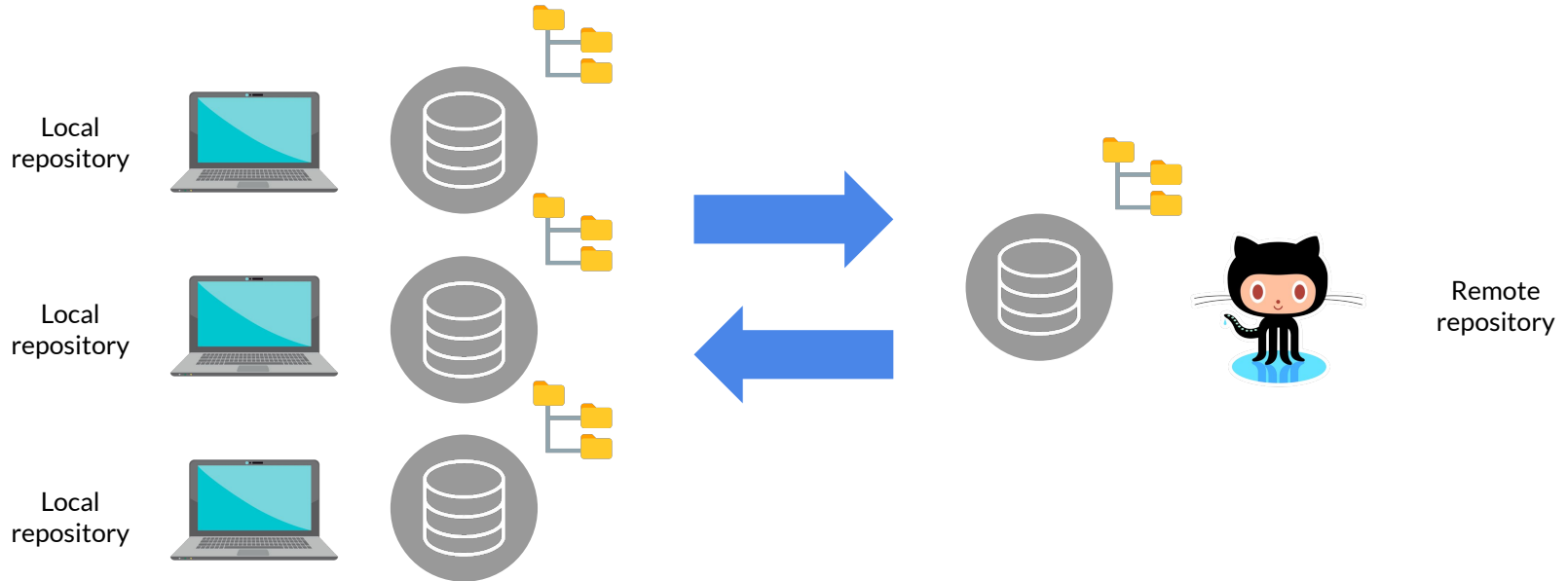
git-gobetti-volta



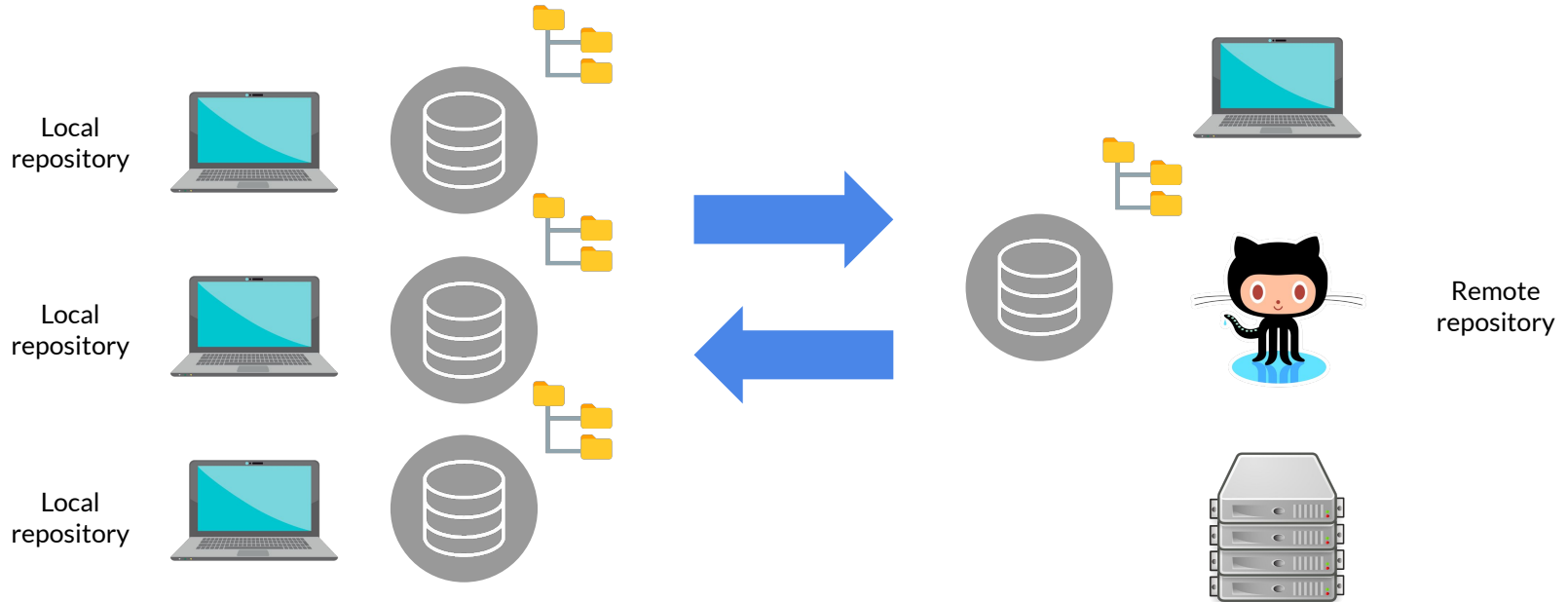
Repository



Repository



Repository



Primi passi con Git: clone, add, remove, commit



Primi passi con Git

Si può “parlare” con Git in tre modi:

- da linea di comando
- tramite programmi con interfaccia grafica (TortoiseGit, SourceTree, GitHub desktop e simili)
- direttamente dall’IDE (Eclipse, VSCode, SublimeText, QtCreator e simili)

Lo useremo solo da linea di comando

Per dare istruzioni a Git

```
git <comando>
```




Primi comandi

Verificare la presenza di Git

```
git --version
```

Configurare nome e email

```
git config --global user.name "Massimiliano Atzori"  
git config --global user.email massimiliano.atzori@gmail.com
```

Output con colori

```
git config --global color.ui.auto
```

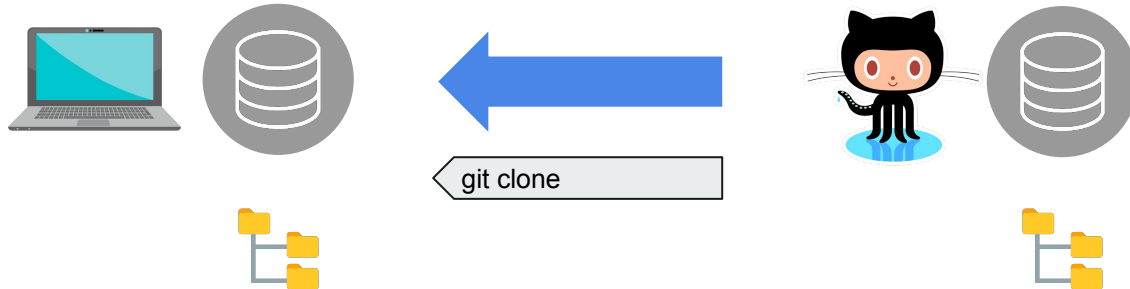
Verificare la configurazione

```
git config --list
```

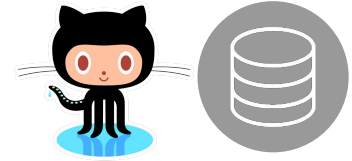
Proviamo!

Clonare un repository da GitHub

```
git clone https://github.com/amaxis/git-gobetti-volta.git
```



Le aree di lavoro di Git



Workspace



Index
(staging area)



Local
repository



Remote
repository



Le aree di lavoro di Git



Workspace

- Nuovi file
- File modificati
- File “eliminati”



Index (staging area)

- Modifiche **pronte**
per essere
registrate



Local repository

- Modifiche
registrate



Remote
repository

- Modifiche
condivise



Controllare lo stato del nostro repo

```
git status
```

Cosa ci dice

- se abbiamo fatto modifiche ai file rispetto al contenuto del local repository
- in quali stage si trovano i file che abbiamo modificato
- se ci sono conflitti da risolvere
- se ci sono altre situazioni particolari



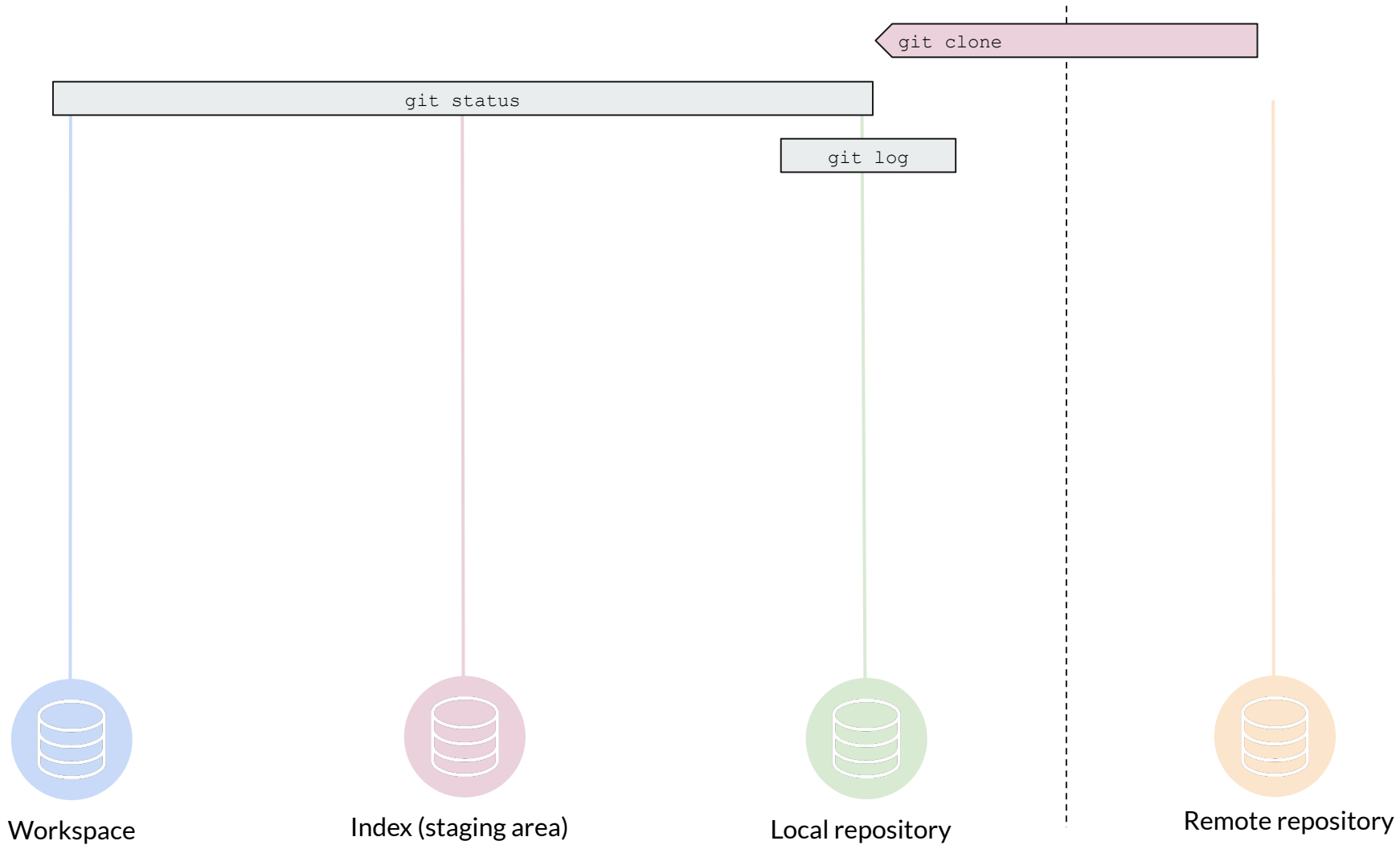
Controllare lo stato del nostro repo

```
git log
```

Cosa ci dice

- ci fa vedere la lista delle modifiche permanenti fatte al repository locale
- due opzioni utili
 - `-n`: mi fa vedere solo le ultime `n` modifiche
 - `--oneline`: mi fa vedere una lista più compatta

```
git log -n <numero di linee> --oneline
```

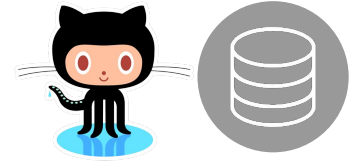


Proviamo!

**Modificare i file e portarli in
remoto**

—

Le aree di lavoro di Git



Workspace



Index
(staging area)



Local
repository



Remote
repository



Un primo semplice flusso di lavoro con Git

Clono un repository (o lo creo in locale)

```
git clone / git init
```

Faccio delle modifiche: aggiungo, modifico, elimino dei file

```
git add / git rm
```

Tengo d'occhio lo stato dei file

```
git status / git log
```

Quando arrivo ad un punto importante, registro le modifiche nel repository locale

```
git commit
```

Quando sono pronto a condividere le modifiche, le porto sul repository remoto

```
git push
```



Aggiungiamo un file nel repository Git locale

Creiamo un file nella directory

```
git-gobetti-volta/classi/
```

Chiamiamo il file

```
nome-cognome.txt
```

Scriviamo nel file un artista che ci piace

```
Joe Bonamassa
```

Vediamo lo status

```
git status
```

Chiediamo a Git di aggiungerlo

```
git add nome-cognome.txt
```

Vediamo lo status

```
git status
```



Proviamo a togliere il file

Chiediamo a Git di non considerare più il file

```
git rm nome-cognome.txt
```

Vediamo lo status

```
git status
```

Aggiungiamolo di nuovo

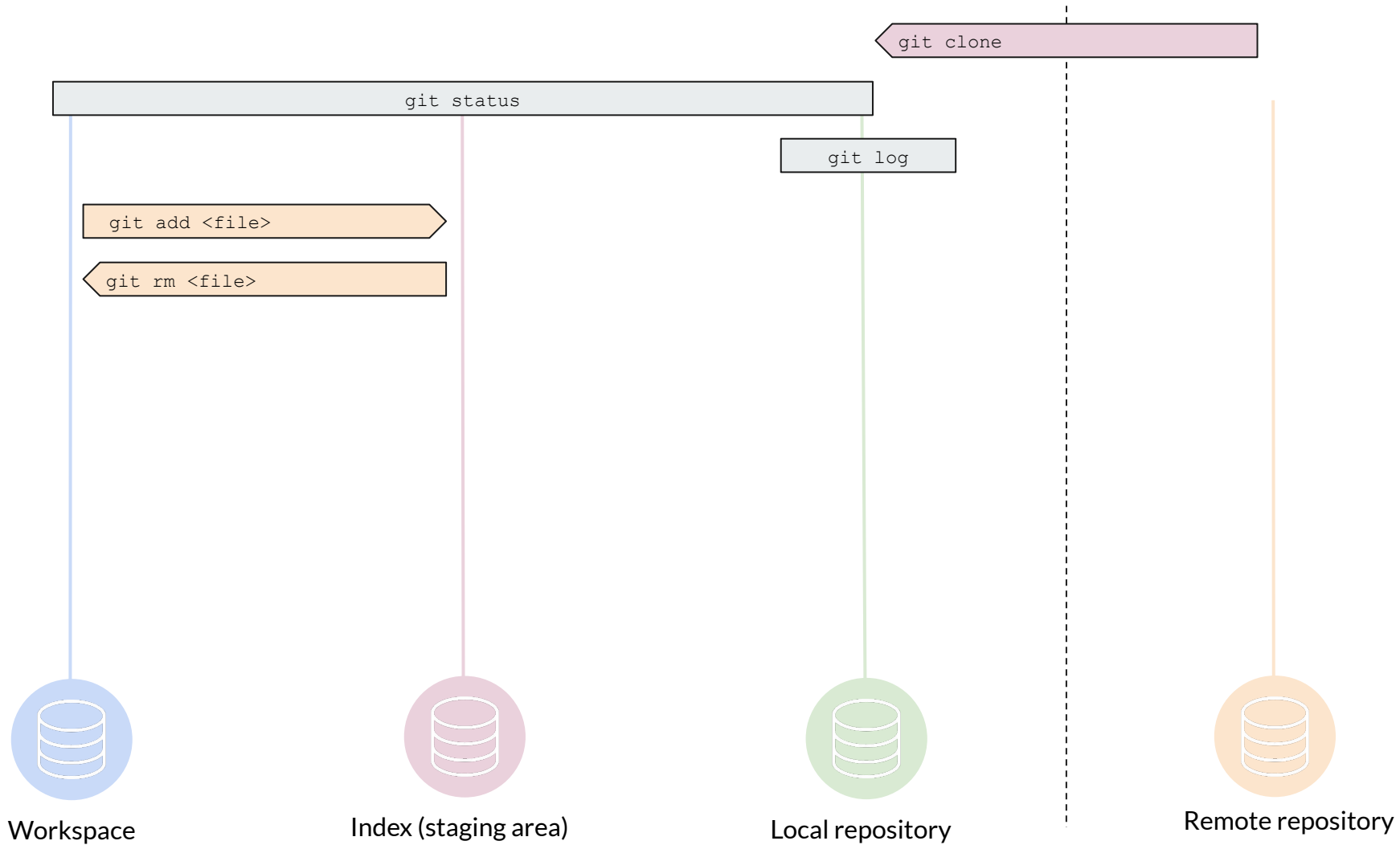
```
git add nome-cognome.txt
```

Vediamo lo status

```
git status
```

Vediamo la situazione del repository

```
git log --oneline
```





Aggiorniamo il repository locale

Vediamo lo status

```
git status
```

Aggiorniamo il repository

```
git commit -m "Aggiunta artista preferito"
```

Vediamo lo status

```
git status
```

Vediamo la situazione del repository

```
git log --oneline
```




Cos'è successo?

Abbiamo spostato tutte i file nella staging area nel repository locale, creando un commit

Un commit è come un'etichetta associata ad un gruppo di modifiche

Ogni commit è uno snapshot permanente

Abbiamo associato un messaggio al commit con -m

Tutte le modifiche sono ancora in locale sulla nostra macchina



Portare le modifiche in remoto

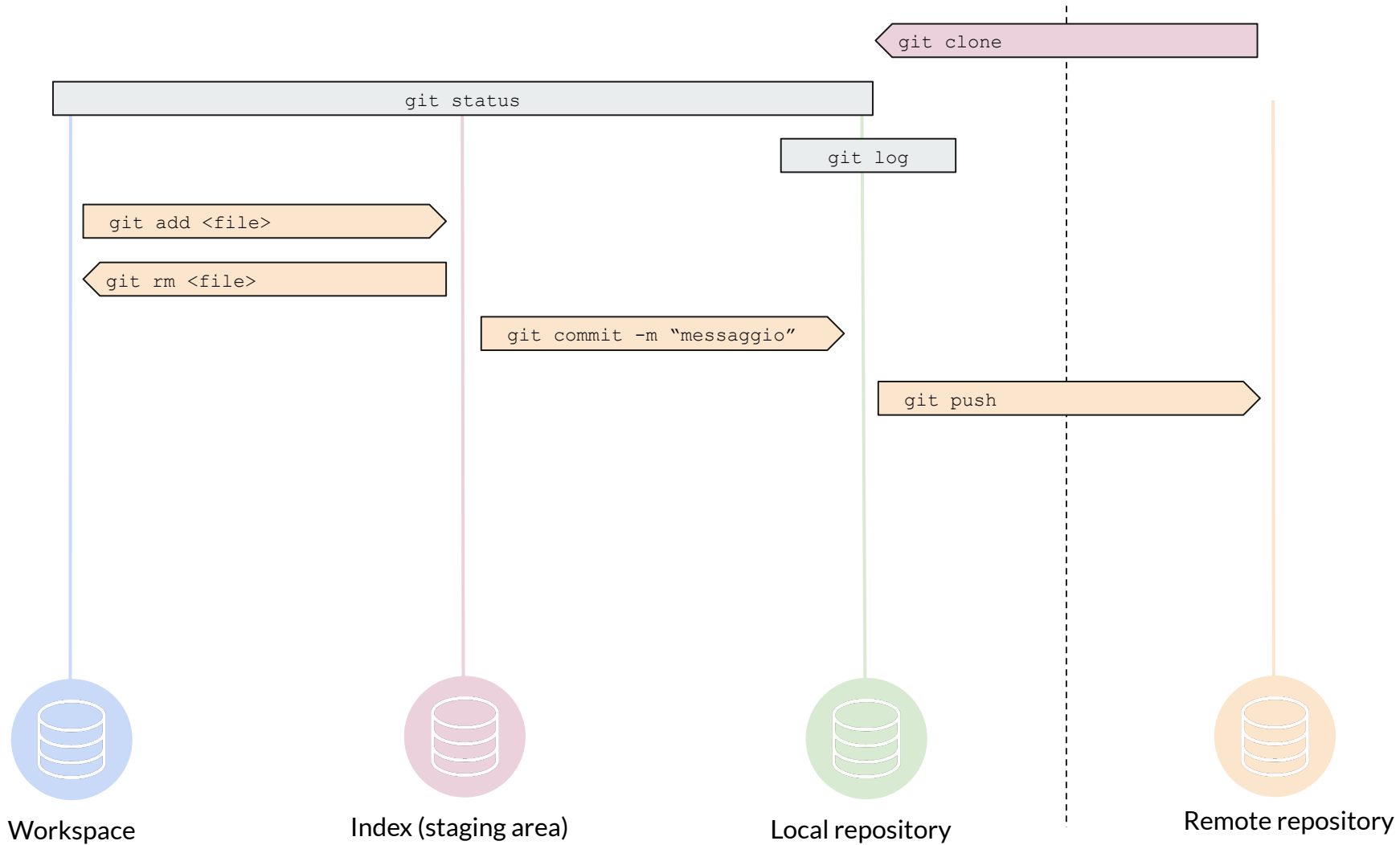
Per portare le modifiche sul repository remoto

```
git push
```

Porta i commit aggiunti nel local repository sul remote repository

Attenzione

- per l'operazione di push, ci vogliono le autorizzazioni sul repository remoto
- portare i commit sul remote può causare **conflitti**





Altri comandi utili

Riportare i file da staging a workspace

```
git reset
```

Ritornare ad una versione precedente

```
git reset --hard
```

Differenza

- la prima riporta i file da staging a workspace, **il contenuto rimane invariato**
- la seconda riporta i file dal local repository a workspace, **il contenuto viene sovrascritto**



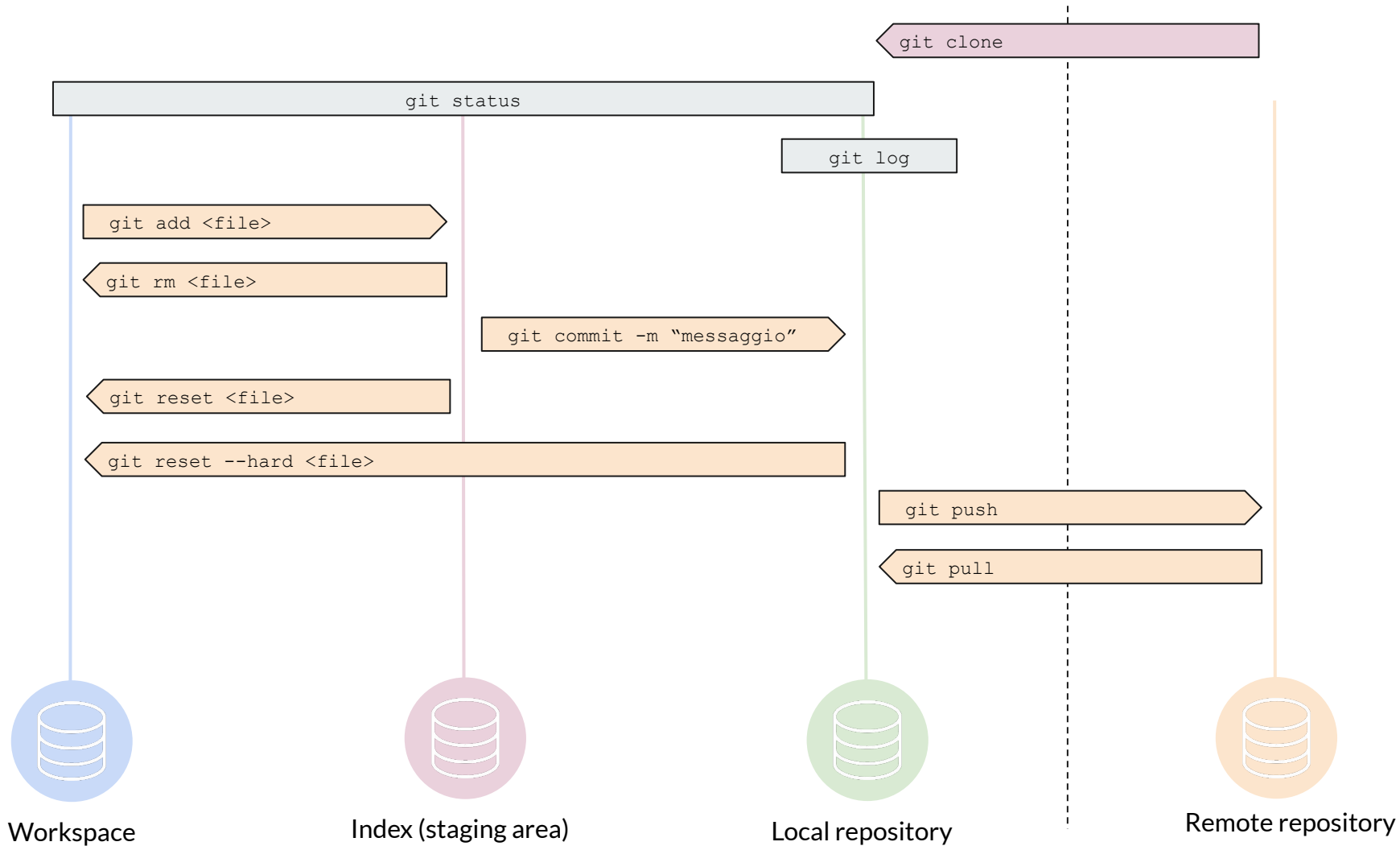
Altri comandi utili

Prende i nuovi commit dal repository remoto

```
git pull
```

Da notare

- il comando `git clone` fa una copia esatta del repository e lo collega a quello remoto
- il comando `git pull` si esegue su un repository già collegato ad uno remoto e copia solo i commit che non si hanno già



Repository e commit: un esempio

tensorflow / tensorflow

Used by ▾

63.9k

Watch ▾

8.5k

★ Star

141k

Fork

79.9k

<> Code

Issues 3,113

Pull requests 242

Actions

Projects 1

Security

Insights

An Open Source Machine Learning Framework for Everyone <https://tensorflow.org>

tensorflow

machine-learning

python

deep-learning

deep-neural-networks

neural-network

ml

distributed

78,898 commits

33 branches

0 packages

103 releases

2,397 contributors

Apache-2.0

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



tensorflow-gardener Go: Update generated wrapper functions for TensorFlow ops. ...

✗ Latest commit 2d94eb1 33 minutes ago

.github/ISSUE_TEMPLATE Fix GitHub issue templates. Actual fix for #36721

2 days ago

tensorflow Go: Update generated wrapper functions for TensorFlow ops.

30 minutes ago

<> Code

! Issues 3,113

🔗 Pull requests 242

▶ Actions

📁 Projects 1

🛡 Security

📊 Insights

Branch: master ▾

🔗 Commits on Feb 16, 2020

Go: Update generated wrapper functions for TensorFlow ops. ...

👤 tensorflow-gardener committed 33 minutes ago ✖



2d94eb1



Fix typo in tf.stack ...

👤 tensorflow-gardener committed 1 hour ago ✖



00cb358



Break dependency of profiler_backends on core:core_cpu_lib ...

👤 jbaiocchi authored and tensorflow-gardener committed 2 hours ago ✖



9f86c8c



Automated rollback of commit a86cf76 ...

👤 tensorflow-gardener committed 4 hours ago ✖



992b5eb



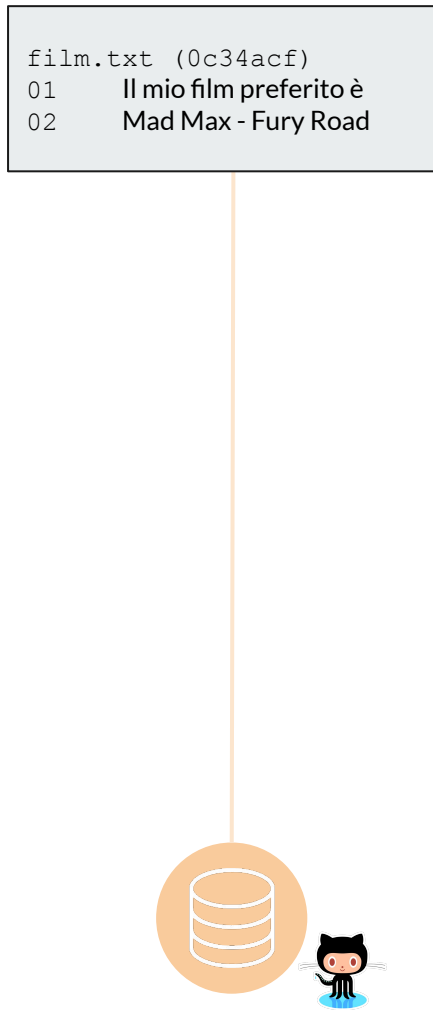
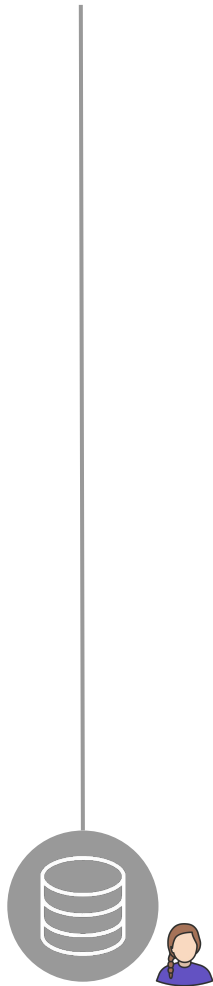
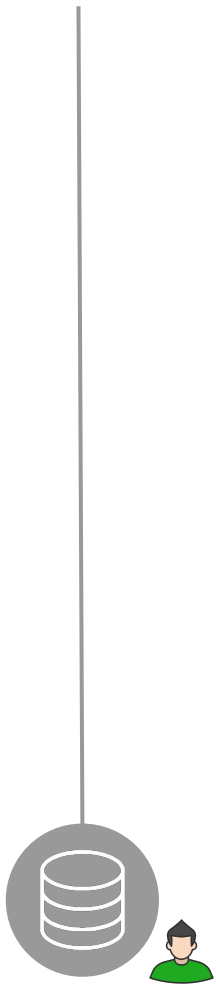
Go: Update generated wrapper functions for TensorFlow ops. ...

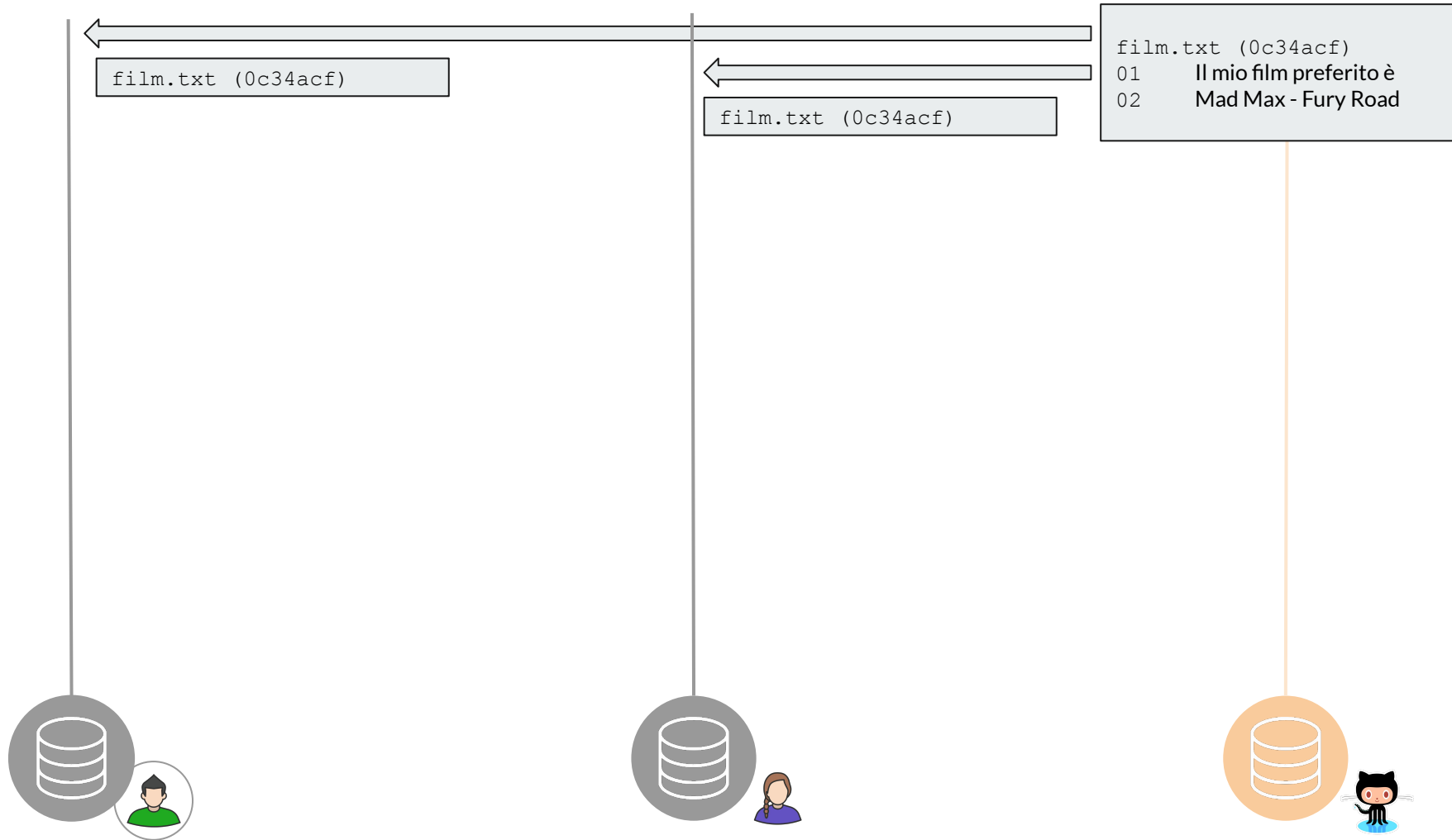


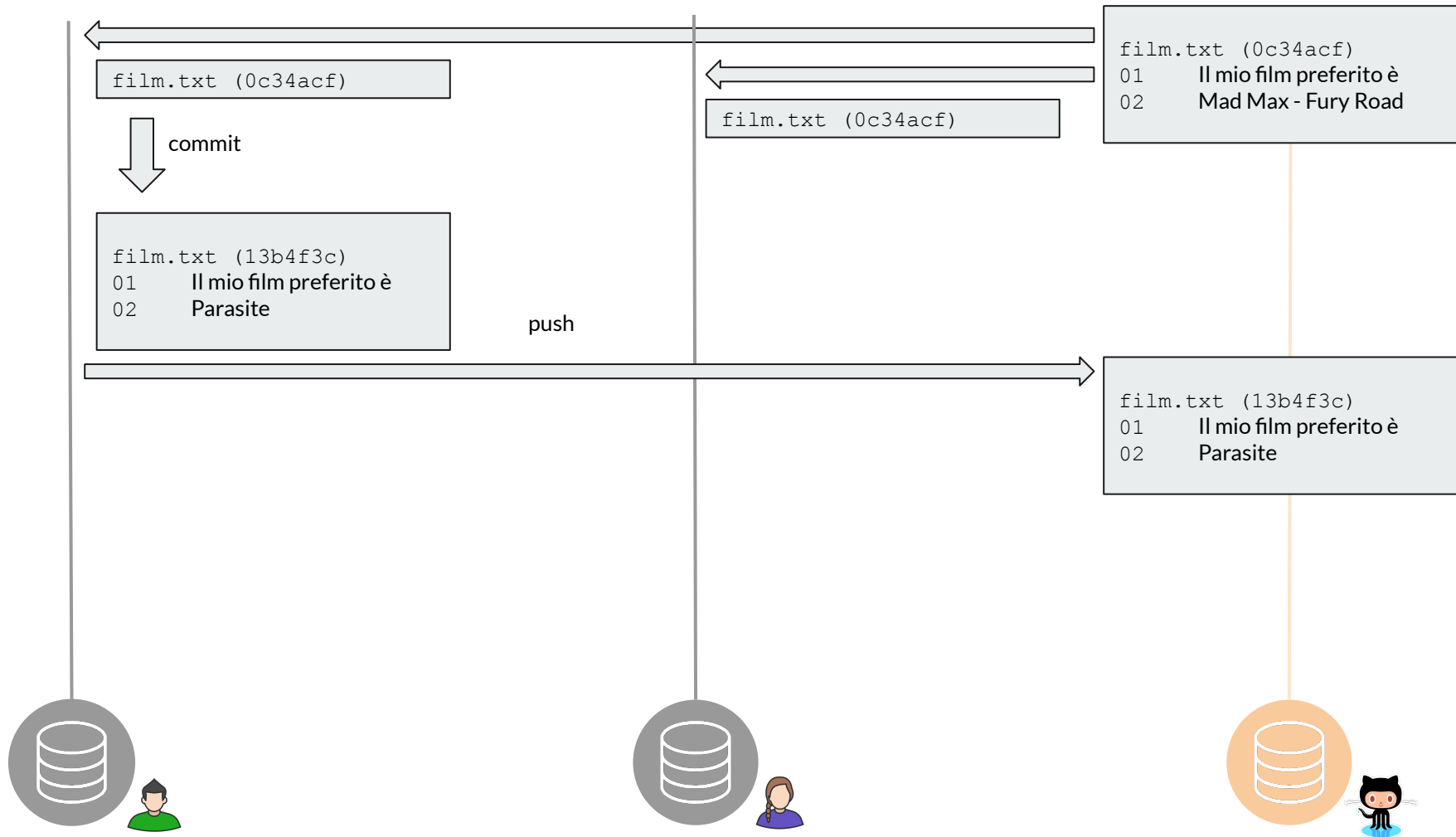
a515b1f

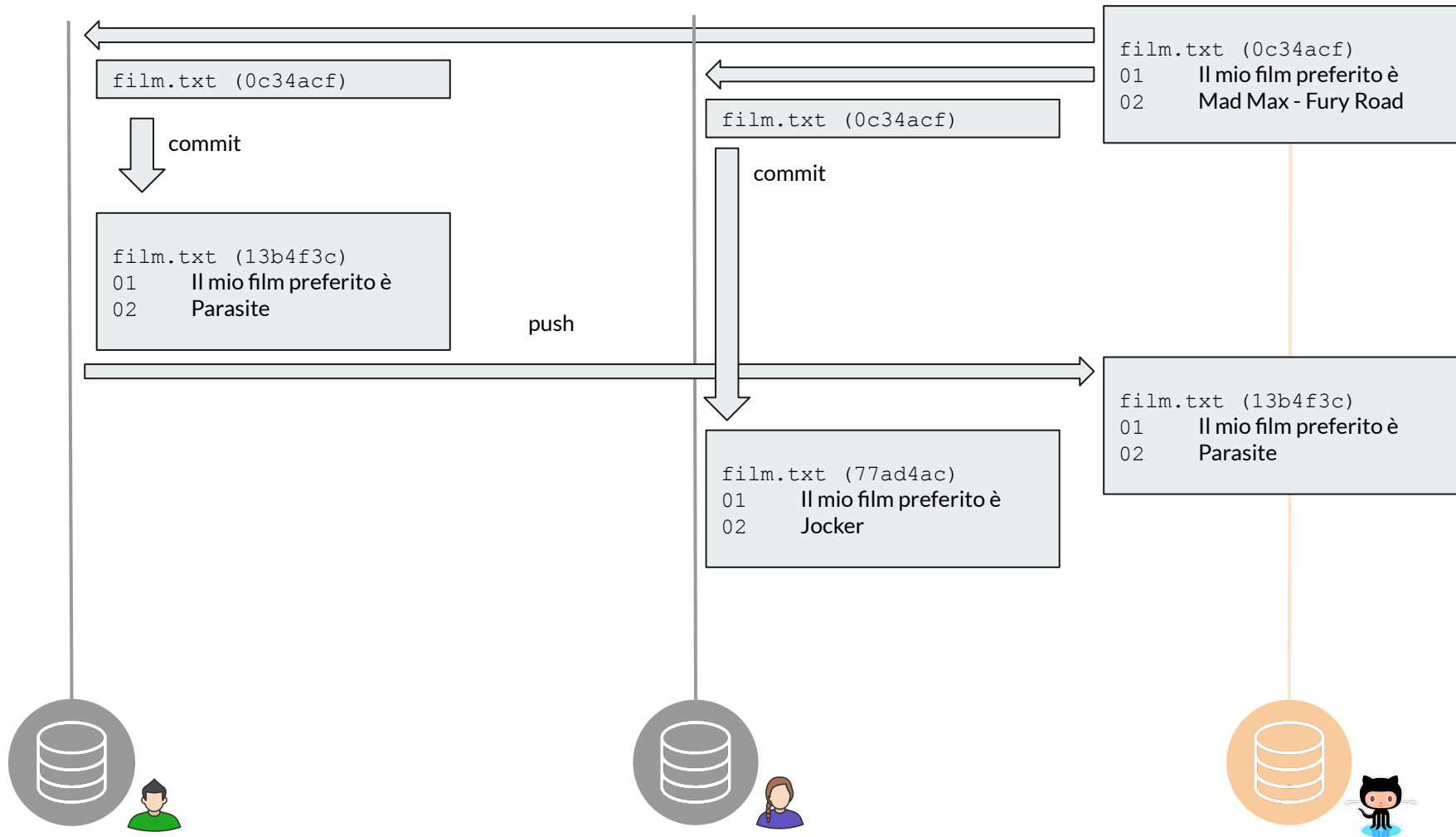


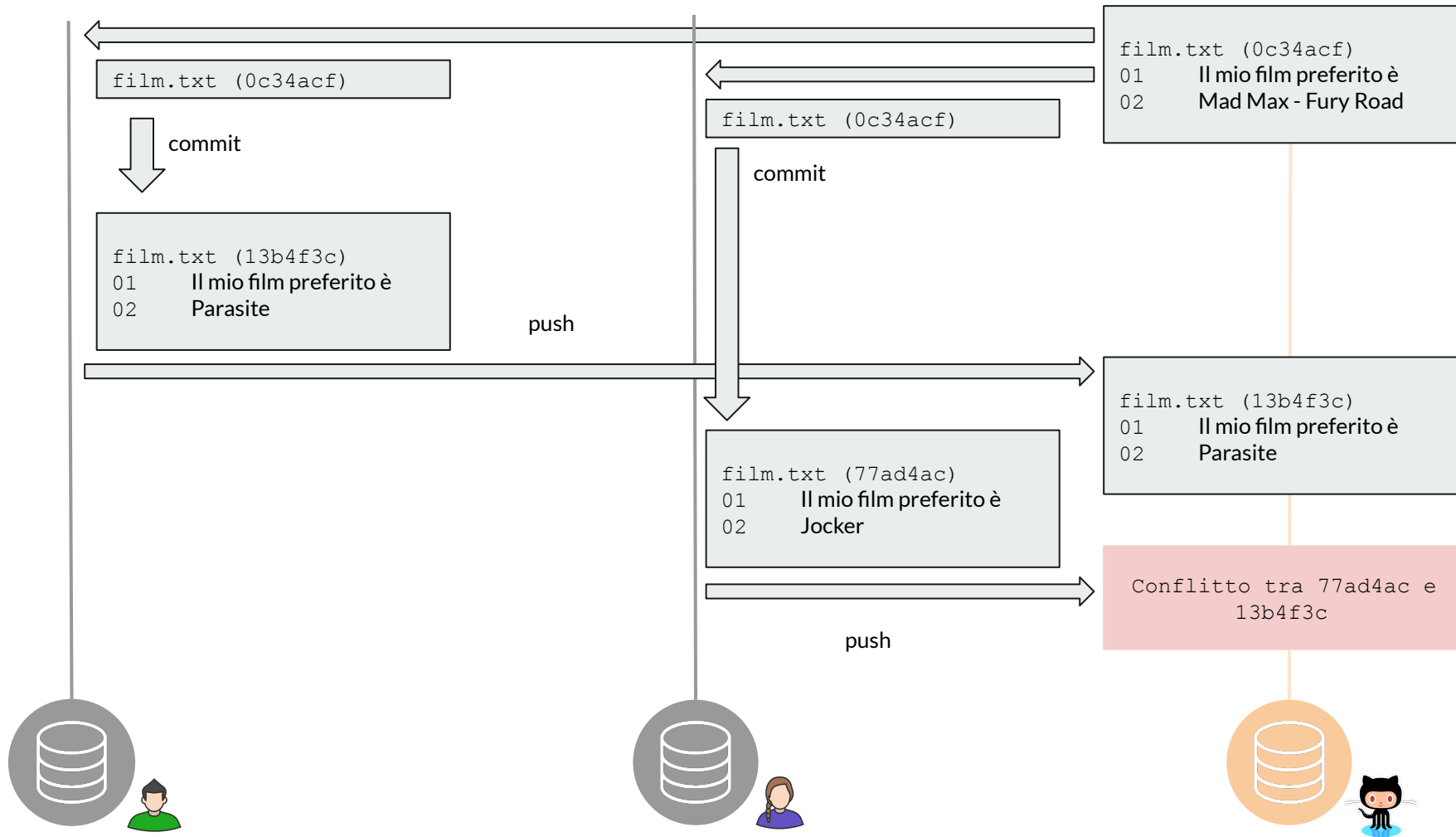
Conflitti e merge











Proviamo!



Cos'è successo?

Git ha identificato un conflitto: ci sono due modifiche, fatte a partire dallo stesso commit, che sono in contraddizione

Non tutte le modifiche sono classificate automaticamente come conflitti: quelle che non di pestano i piedi, vanno bene (per esempio, aggiunte)

Ogni volta che si fa un'operazione di push o pull, **si possono verificare dei conflitti**



Cosa fare in caso di conflitto

Quando c'è un conflitto, Git scrive nei file quali parti sono in contraddizione. Per risolvere il conflitto, basta scrivere qual è la versione corretta

```
film.txt (77ad4ac)
01 Il mio film preferito è
02 <<<<<<< HEAD
03 Parasite
04 =====
05 Jocker
06 >>>>>>> 77ad4ac
```

Cosa fare in caso di conflitto

Quando c'è un conflitto, Git scrive nei file quali parti sono in contraddizione. Per risolvere il conflitto, basta scrivere qual è la versione corretta

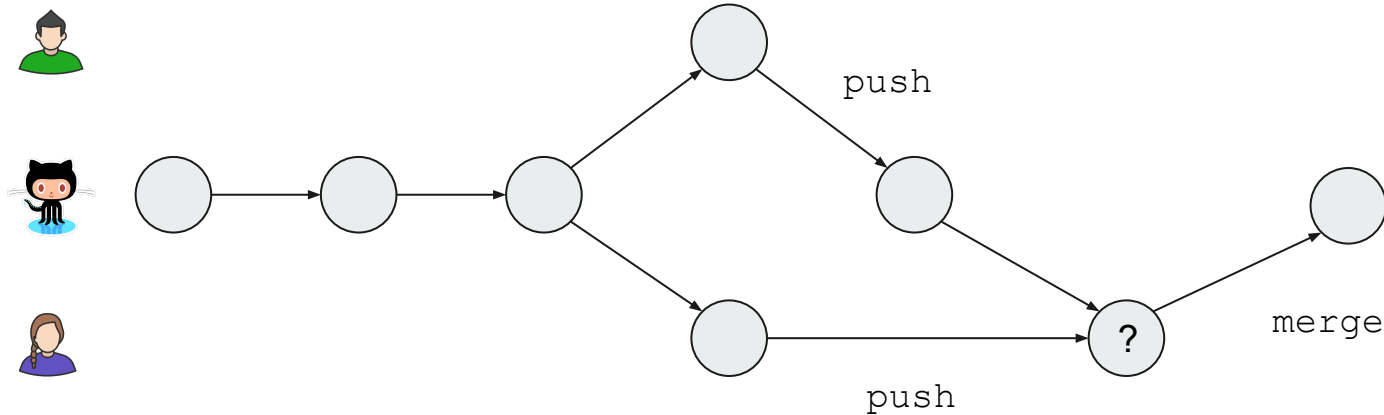
```
film.txt (77ad4ac)
01 Il mio film preferito è
02 <<<<<<< HEAD
03 Parasite
04 =====
05 Jocker
06 >>>>>>> 77ad4ac
```



```
film.txt (77ad4ac)
01 Il mio film preferito è
02 Jocker
```

Cos'è successo?

Albero dei commit (snapshot)





Esempio

1. Modifico un file

```
README.md
```

2. Lo aggiungo in staging

```
git add README.md
```

3. Lo aggiungo al repository (faccio un commit)

```
git commit -m "Modifica nome istituto"
```

4. Provo a fare un push delle modifiche

```
git push
```

5. Conflitto: apro il file e faccio le modifiche

```
README.md
```

6. Rifaccio un commit

```
git commit -m "Modifica nome istituto"
```

7. È andata! 🎉

Proviamo!



Riepilogo

1. Installiamo Git e creiamo un account su GitHub
2. Breve introduzione a Git
3. Primi passi con Git: clone, add, remove, commit
4. Portare le modifiche in remoto
5. Conflitti e merge
6. Riepilogo e consigli utili

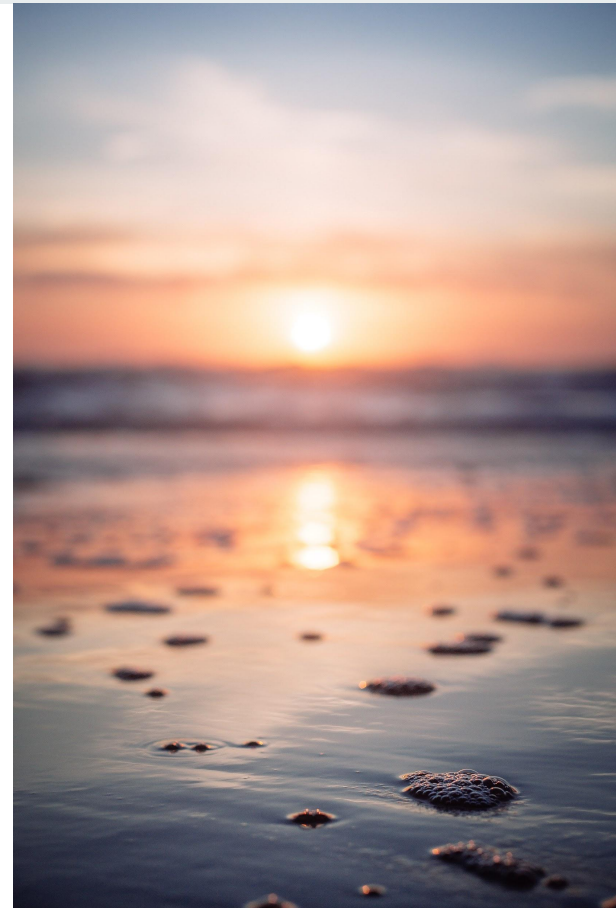


Photo by Cole Keister on Unsplash

Cosa manca

Branch, tag, remotes

Merge, rebase

Git tools (comandi avanzati)

Git workflow



Photo by Joshua Earle on Unsplash



Link e risorse

Libro

<https://git-scm.com/book/en/v2>

Slides e video

<https://channel9.msdn.com/Series/Introduzione-a-Git>

<https://www.slideshare.net/valix85/introduzione-a-git-ita-2017>

Google :)

Questa presentazione

<https://github.com/amaxis/git-gobetti-volta>

`git clone https://github.com/amaxis/git-gobetti-volta`

Grazie

Massimiliano Atzori

massimiliano.atzori@gmail.com

@amaxis