

Versionamento del codice sorgente

Introduzione a **git**



www.inmagik.com

Ing. Mauro Bianchi
mauro.bianchi@inmagik.com

Perchè versionare il codice

- Tenere traccia dei cambiamenti
- Mantenere un backup
- Lavorare in team ad un progetto
- Poter sperimentare senza paura di “rompere” il codice
- Semplificare il deployment

Git

- Veloce
- Distribuito
- Permette di lavorare offline
- Incoraggia il “branch”
- Molti servizi e applicativi supportati
 - Github
 - Bitbucket
 - Gitlab
- Multi piattaforma
- Anche tools GUI

Basi di git

— — —

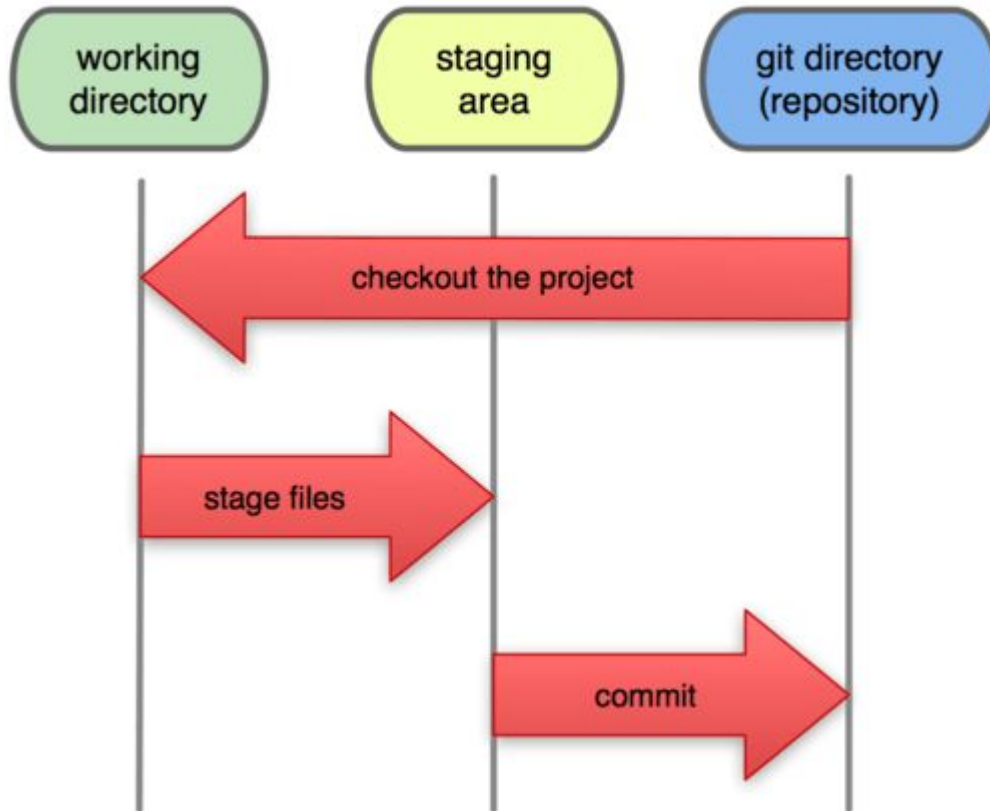
Riferimento

<https://git-scm.com/book/it/v1/Per-Iniziare>

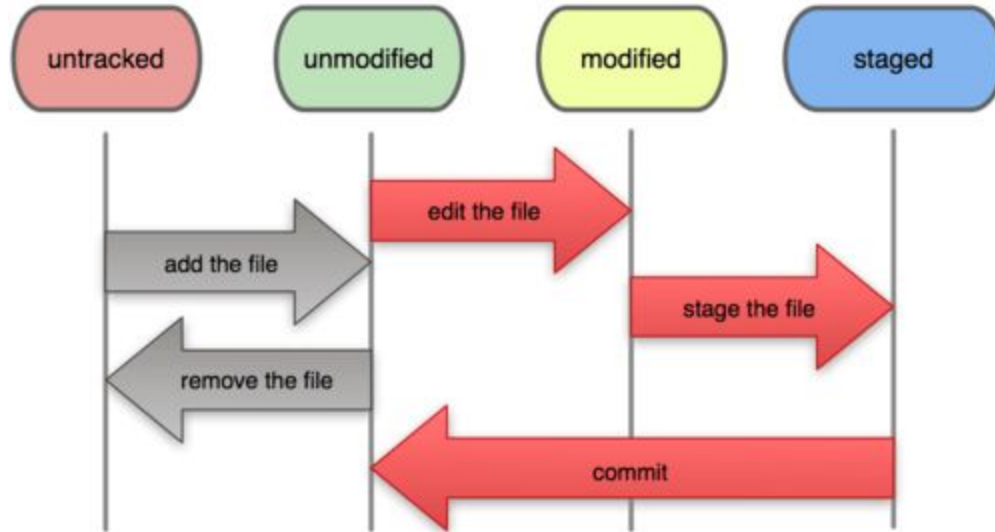
- Modello di funzionamento distribuito
- Configurazione

Local Operations

— — —



File Status Lifecycle



Git: comandi base

- `init`: inizializza un nuovo repository
- `status`: visualizza lo stato corrente del repository
- `clone`: clona un repository da un nodo remoto
- `add`: aggiunge uno o più files al prossimo commit
- `commit`: memorizza lo snapshot delle modifiche fatte (anche `--amend`)
- `log`: visualizza gli ultimi commit
- `reset`: riporta lo stato della working directory al commit precedente

git: comandi di gestione branch

branch

diff: mostra differenze tra due branch o due commit

merge: porta commit di un'altro branch all'interno del branch corrente

checkout: sposta la head tra branch o commit

git: interazione con i remote

clone: clona un repository remoto via ssh o http

remote: gestisce i server remoti per un repository

push: invia modifiche locali ad un remote

pull: ottiene modifiche da un remote

git: files speciali

`.gitignore`: contiene i files che git non deve considerare quando analizza la working directory

collezione di files `.gitignore` per vari linguaggi:

<https://github.com/github/gitignore>

`.gitkeep`: contiene eventuali eccezioni a files ignorati da `.gitignore`

`.git/config`: configurazione del repository git

git: servizi online

— — —

github: <https://github.com/>

bitbucket: <https://bitbucket.org>

gitlab: <https://about.gitlab.com/>

- Forniscono server remoti per repository git
- Aggiungono funzionalità aggiuntive (ex gestione issues, collaborazione, webhooks, integrazione altri servizi)

<https://opensource.com/life/16/8/how-construct-your-own-git-server-part-6>

git: branching models

— — —

<http://nvie.com/posts/a-successful-git-branching-model/>

<https://git-scm.com/book/it/v2/Git-Branching-Branching-Workflows>

riferimenti

— — —

editor

<https://code.visualstudio.com/>

atom

<https://atom.io/>