

0. Come condividere risultati e informazioni

Insegnamento di Informatica

Elisabetta Ronchieri

Corso di Laurea di Economia, Università di Ferrara

I semestre, anno 2014-2015



Argomenti

Sistema di controllo di versione (Version Control System)

- Di cosa si tratta

- Perché sono necessari

- Vantaggi

- Tipologie

Git

- What is Git

- How to install and configure it

- How to clone the class repository and update your clone

- How to create your own Bitbucket repository

- Basic Git commands

Online Storage Service

- Di cosa si tratta

- Esempi di applicazioni gratuite



Argomenti

Sistema di controllo di versione (Version Control System)

- Di cosa si tratta

- Perché sono necessari

- Vantaggi

- Tipologie

Git

- What is Git

- How to install and configure it

- How to clone the class repository and update your clone

- How to create your own Bitbucket repository

- Basic Git commands

Online Storage Service

- Di cosa si tratta

- Esempi di applicazioni gratuite



Definizione

*Un sistema di controllo di versione (Version Control System (VCS)) é una combinazione di tecnologie e procedure per **tenere traccia**, nel tempo, e **controllare** i cambiamenti ad un file o ad un insieme di file, recuperabili successivamente richiamando una versione specifica [Chacon, 2009].*

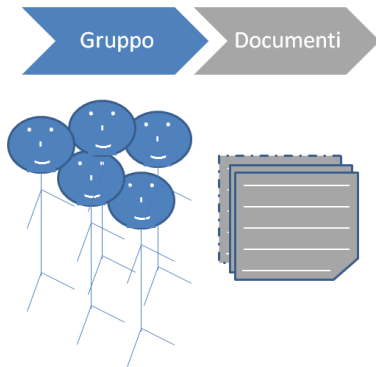
- I file possono contenere documenti, codice, pagine Web e altri insiemi di informazione

http://en.wikipedia.org/wiki/Revision_control.



Di cosa si tratta

- ▶ VCS aiuta in ogni aspetto della gestione di un progetto.
- ▶ Automatizza una serie di operazioni. Per esempio:
 - ▶ tracciare chi ha effettuato una modifica su un file;
 - ▶ unire le modifiche una volta completate;
 - ▶ ripristinare un file ad una versione precedente;
 - ▶ segnalare eventuali conflitti;
 - ▶ revisionare le modifiche fatte nel tempo.
- ▶ Particolarmente importante quando il progetto é condiviso tra più persone.



Perché sono necessari

- ▶ Originariamente, i VCS sono stati sviluppati per aiutare lo sviluppo di grandi progetti software con la partecipazione di molti sviluppatori su porzioni di codice correlate tra di loro.
- ▶ Oggi sono usati:
 - ▶ non solo da gruppi di persone che lavorano insieme. Anche da singoli sviluppatori.
 - ▶ non solo per la gestione di progetti software. Anche durante la scrittura di articoli e documenti da parte di ricercatori.

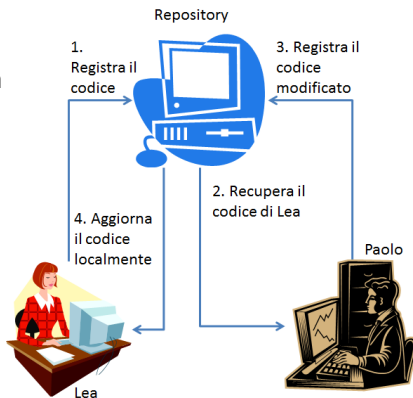


Come funziona: idea di base

Supponiamo di voler lavorare su un file.

Supponiamo di avere definito un'area di lavoro sul proprio desktop.

- ▶ Si recupera localmente il file da una repository.
- ▶ Si modifica il file e si registra nella repository.



La repository tiene traccia di tutte le modifiche e di chi le ha fatte:

- ▶ associa ad ogni modifica del file registrato una versione.
- ▶ per ogni registrazione tiene traccia della differenza tra versioni.



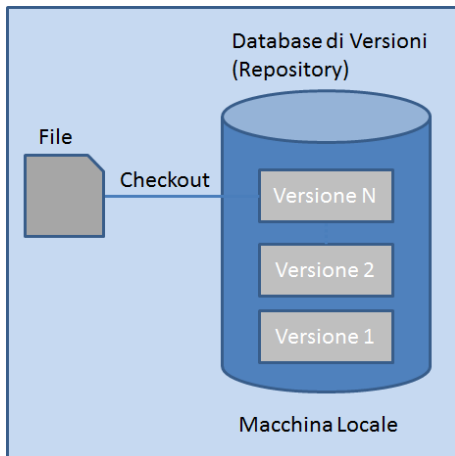
Vantaggi

- ▶ É possibile **ritornare** alla versione precedente di un file, nel caso di modifiche sbagliate.
- ▶ Si possono **confrontare** versioni differenti per vedere i cambiamenti introdotti.
- ▶ É possibile **registrare** i dati e il codice che servono per un articolo:
 - ▶ mantenere traccia dei dati prodotti;
 - ▶ garantire riproducibilità.
- ▶ Se si lavora usando diverse macchine, dal desktop al portatile, VCS permette di mantenere le informazioni **sincronizzate tra le varie macchine**.



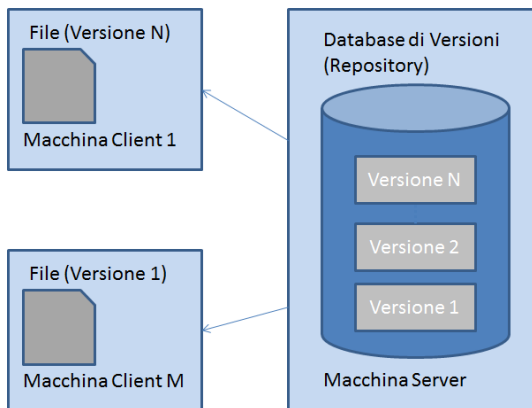
VCS locale

- ▶ Un database (detto **repository**) mantiene tutti i cambiamenti dei file sotto controllo di revisione.
- ▶ Il file viene recuperato sulla stessa macchina del database.



VCS centralizzato: modello client-server

- ▶ Una macchina server contiene il repository.
- ▶ 1 o più macchine client da cui recuperare i file dalla repository.
- ▶ Solo la repository conserva l'intera storia di tutte le modifiche.



Sistemi come Concurrent Version System (CVS) e SubVersion (SVN) rientrano in questa tipologia di VCS.

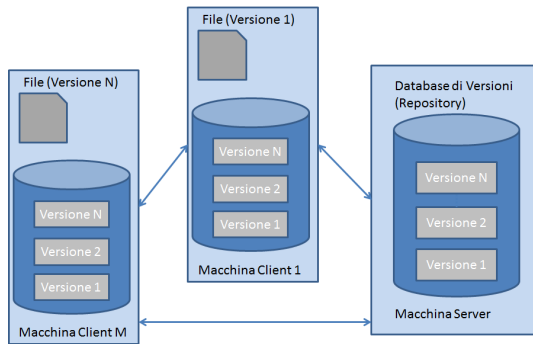
Ulteriori informazioni

<http://software-carpentry.org/v4/vc/>.



VCS distribuito

- ▶ Sulle macchine client si ha una copia dei file piú recenti (detto anche **snapshot**).
- ▶ Ma anche una copia completa della repository.
- ▶ Ogni operazione di recupero effettua una copia completa di tutti i dati.



Sistemi come Git e Mercurial rientrano in questa tipologia di VCS.

Ulteriori informazioni <http://mercurial.selenic.com/>.



Comparazione tra le varie tipologie

Caratteristica	Locale	Centralizzato	Distribuito
Tracciabilità delle modifiche.	✓	✓	✓
Collaborazione tra più persone.		✓	✓
Controllo sugli accessi.		✓	✓
Facile amministrazione.		✓	✓
Singolo punto di fallimento ¹ .	✓	✓	
Perdita dei dati ² .	✓	✓	
Copia repository client.			✓
Recupero repository.			✓

¹La macchina server é andata giù.

²Disco del server si é corrotto.

Argomenti

Sistema di controllo di versione (Version Control System)

- Di cosa si tratta

- Perché sono necessari

- Vantaggi

- Tipologie

Git

- What is Git

- How to install and configure it

- How to clone the class repository and update your clone

- How to create your own Bitbucket repository

- Basic Git commands

Online Storage Service

- Di cosa si tratta

- Esempi di applicazioni gratuite



What is Git

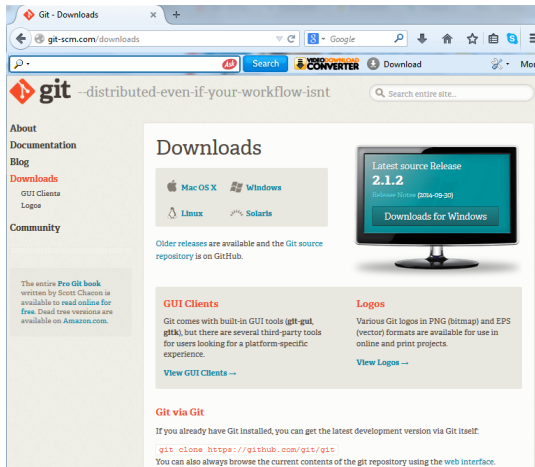
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

<http://git-scm.com>



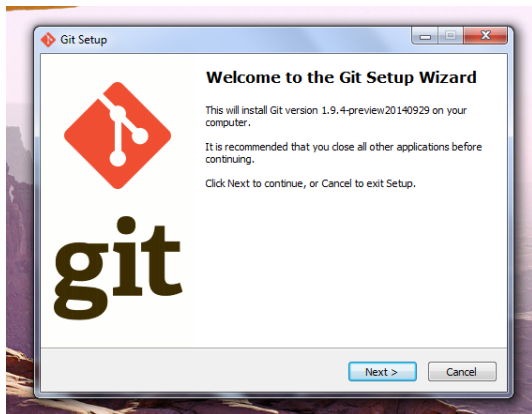
Steps to install Git

- 1 Go to the website.
<http://git-scm.com/downloads>
- 2 Click on the download link for your Operating System (OS), such as Mac, Windows and Linux.



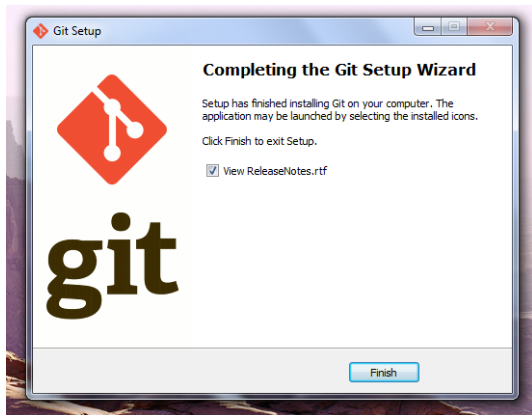
Steps to install Git

- 3 Once the file is done downloading (selected Windows), open it up to begin the Git installation.
- 4 Click on **Next**.



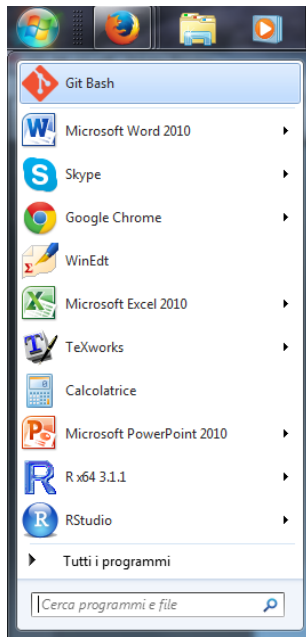
Steps to install Git

- 5 Go with the default options at each step of the installation.
 - ▶ This is a recommendation.
- 6 Once the install is complete, click on **Finish**.



Steps to open Git Bash

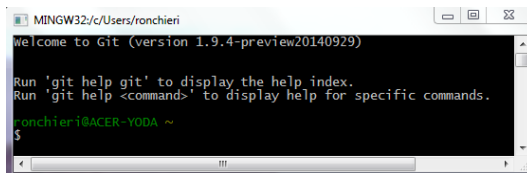
- 1 Find a program called **Git Bash**.
 - ▶ Git Bash is the environment for interacting with Git.
 - ▶ It is located in the installation path: for Windows OS, it is in the Start Menu.



Steps to open Git Bash

2 Once Git Bash opens, you'll see:

- ▶ a Welcome message;
- ▶ the name of your computer;
- ▶ a dollar sign (you can type a command).



```
MINGW32:/c/Users/ronchieri
Welcome to Git (version 1.9.4-preview20140929)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

ronchieri@ACER-YODA ~
$
```



Steps to configure Git locally

- 1 Set your username:

```
$ git config --global user.name "Your Name"
```

- 2 Set your e-mail:

```
$ git config --global user.email "email@domain"
```

- 3 Specify the previous commands the first time.

- ▶ However, you change them on the way with the same commands.

- 4 Verify your changes:

```
$ git config --list
```

- 5 Close Git Bash:

```
$ exit
```



Cloning the class repository

- ▶ All the materials are in a Git repository hosted at Bitbucket.
<https://bitbucket.org/joda70/informatica-economia.unife>
- ▶ Via the same link, you can monitor all changesets, issues and so on.



Steps to clone the class repository

- 1 Once Git Bash opens, move to the directory where you want your copy to reside called **Corso**.

```
$ cd Corso
```

- 2 Clone the repository:

```
$ git clone \  
> https://bitbucket.org/joda70/informatica-economia.unife.git
```

- ▶ This will download the entire repository as a new subdirectory called **informatica-economia.unife**, residing into your home directory.

- 3 Clone the repository in a non-default directory called **Informatica**:

```
$ git clone \  
> https://bitbucket.org/joda70/informatica@economia.unife.git \  
> Informatica
```

- 4 Move to the repository directory:

```
$ cd Informatica
```



Steps to update your clone

1 Move to the repository directory:

```
$ cd Informatica
```

2 Fetch any changes from **origin**:

```
$ git fetch origin
```

- ▶ **origin** points to the remote Bitbucket repository that you originally cloned from.

3 Merge any changes retrieved into the files in your current directory:

```
$ git merge origin/master
```

- ▶ **origin/master** refers to the master branch in the repository.
- ▶ The master branch is the only branch in the repository (at the moment).

4 Update origin and master copies:

```
$ git pull
```



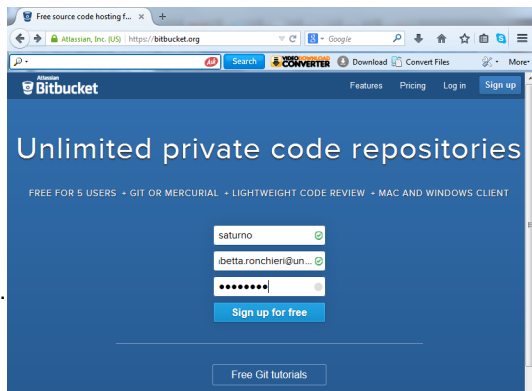
Why it is important

- ▶ You should learn how to use Bitbucket.
<https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>
- ▶ You'll create your own repository on Bitbucket for the project of Informatica.
- ▶ You'll give the teacher permission to clone your repository.
- ▶ You'll remove the permissions to the teacher after the exam.



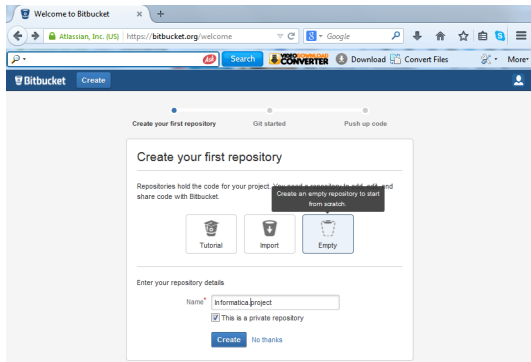
Steps to create your own Bitbucket repository

- 1 Configure Git locally (see slide 19).
- 2 Go to the website.
<https://bitbucket.org/>
- 3 Click on **Sign up**, if you do not have already an account.
- 4 Fill in the form.



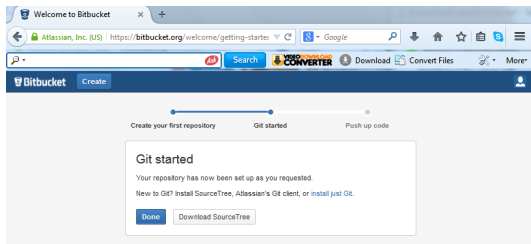
Steps to create your own Bitbucket repository

- 5 Create an empty repository to start from scratch.
- 6 Fill in your repository name, e.g. [Informatica.project](#).
- 7 Click on [Create](#).



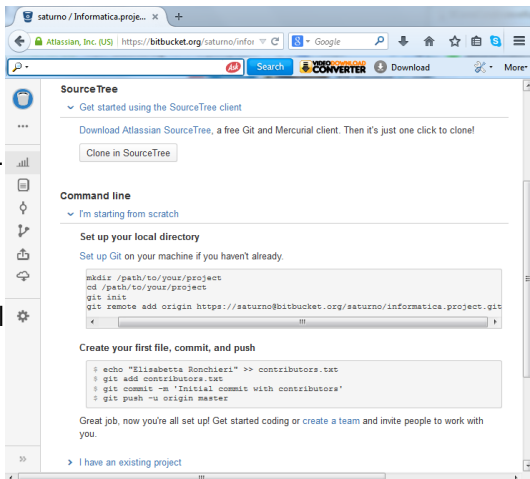
Steps to create your own Bitbucket repository

- 8 The Informatica.project has been set up in the private way.
- 9 Click on Done.



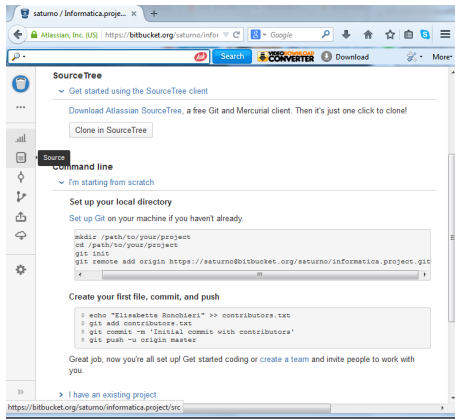
Steps to create your own Bitbucket repository

- 10 See a page with instructions.
- 11 Click on **I'm starting from scratch**.
- 12 Follow the instructions specified to set up your local directory by using Git Bash.



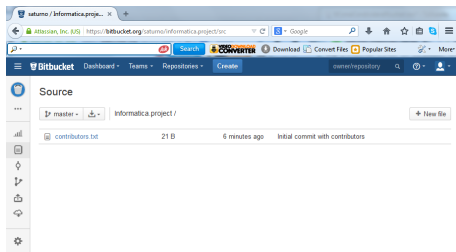
Steps to create your own Bitbucket repository

- 13 Once the push command ends, move on the Source icon and click on it.



Steps to create your own Bitbucket repository

- 14 Examine the class repository at <https://bitbucket.org/joda70/informatica-economia.unife> to see how to structure your project.



Adding

- ▶ Suppose you add new files to a local repository.
- ▶ Let Git know that they need to be tracked.
 - ▶ Add all new files:
`$ git add .`
 - ▶ Update tracking for files that changed names or were deleted:
`$ git add -u`
 - ▶ Do both of the previous operations:
`$ git add -A`



Committing

- ▶ Suppose you perform some changes in your local repository.
- ▶ Commit to save changes as an intermediate version:

```
$ git commit -m "message"
```

- ▶ This updates your local repository.



Pushing

- ▶ Suppose you have saved local changes.
- ▶ Update the remote repository:

```
$ git push
```



Branches

- ▶ Suppose you want to work on a testing version.
- ▶ Create a branch:
`$ git checkout -b "branch name"`
- ▶ See the active branch:
`$ git branch`
- ▶ Switch back to the master branch type:
`$ git checkout master`

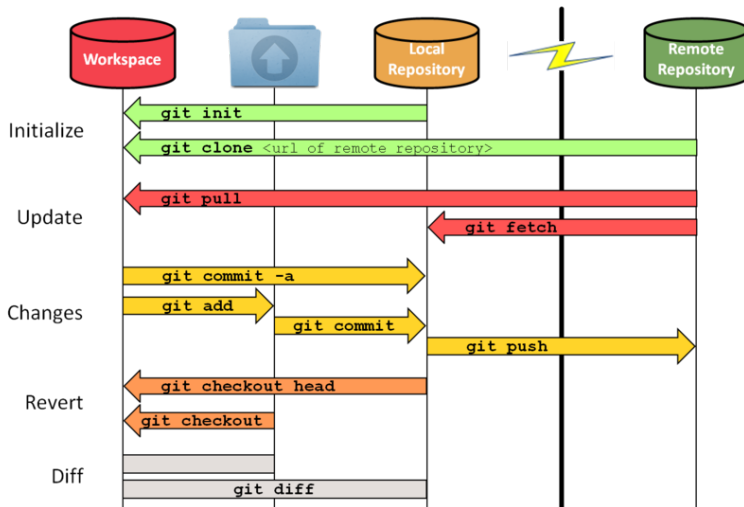


Pulling

- ▶ Suppose you want to merge your changes into the branch/repository.
- ▶ Update the remote repository:
`$ git pull`



Git data transport commands



<https://sselab.de/lab2/public/wiki/sselab/index.php?title=Git>



Argomenti

Sistema di controllo di versione (Version Control System)

- Di cosa si tratta

- Perché sono necessari

- Vantaggi

- Tipologie

Git

- What is Git

- How to install and configure it

- How to clone the class repository and update your clone

- How to create your own Bitbucket repository

- Basic Git commands

Online Storage Service

- Di cosa si tratta

- Esempi di applicazioni gratuite



Di cosa si tratta

- ▶ Online Storage Service (OSS) é un servizio basato su Web.
- ▶ Mette a disposizione di ogni utente registrato uno spazio storage, raggiungibile ovunque via Internet.
- ▶ Permette di:
 - ▶ caricare documenti, video, foto;
 - ▶ sincronizzare documenti da piú computer o altri dispositivi (come tablet o smartphone);
 - ▶ scambiare file con altri utenti iscritti;
 - ▶ accedere ai documenti da ogni altro possibile computer del mondo via Internet.



Esempi di applicazioni gratuite



<https://www.box.com>



<https://www.dropbox.com>



Google Drive

<https://drive.google.com>

I migliori top 10 OSS:

<http://online-storage-service-review.toptenreviews.com/>



- ▶ Box é un esempio di OSS creato per l'ambiente di lavoro.
- ▶ Chiunque si può iscrivere a Box per un accesso gratuito.
- ▶ Ha un elevato numero di funzioni per la privacy e la condivisione dell'informazione.
- ▶ Per accedere ai file su Box si può usare:
 - ▶ il sito Web di Box;
 - ▶ applicazioni per sistemi operativi come Mac e Windows.
 - ▶ applicazioni per piattaforme come iOS, Android e BlackBerry.

- ▶ É possibile salvare qualsiasi tipo di file su Box:
 - ▶ utilizzando il sito Web;
 - ▶ tramite le varie applicazioni disponibili.
- ▶ Il servizio supporta l'operazione:
 - ▶ controllo della privacy su file. Per esempio, stabilire chi può fare cosa;
 - ▶ sincronizzazione dei file sui vari dispositivi utilizzati.

Dropbox

- ▶ Dropbox é un esempio di OSS.
- ▶ Per accedere ai file su Dropbox si può usare:
 - ▶ il sito Web di Dropbox;
 - ▶ applicazioni per sistemi operativi come Mac, Windows e Linux;
 - ▶ applicazioni per piattaforme come iOS, Android, BlackBerry e Kindle Fire.

⇒ Serve il collegamento Internet.



Dropbox

- ▶ É possibile salvare qualsiasi tipo di file su Dropbox:
 - ▶ utilizzando il sito Web;
 - ▶ tramite le varie applicazioni disponibili.
- ▶ Il servizio supporta l'operazione:
 - ▶ drag-and-drop di file e cartelle dal proprio dispositivo allo storage remoto e viceversa;
 - ▶ sincronizzazione dei file sui vari dispositivi utilizzati.
- ▶ É possibile caricare file di diversa dimensione.



- Offre diversi livelli di spazio di archiviazione.³

Gratuito	Spazio
Di Base	2GB
Tramite tutorial iniziale	250MB
Tramite le foto caricate con app. di cellulari	3GB sommando tutti i dispositivi
Amici invitati registrati	500GB per amico fino ad un massimo di 16GB

³Dati aggiornati a metà 2014.

Dropbox

- Offre diversi livelli di spazio di archiviazione.⁴

A pagamento	Spazio
≈ 10 \$ al mese	50GB
≈ 20 \$ al mese	100GB

Per il corso non serve extra storage rispetto ai 2.25GB.

⁴Dati aggiornati a metà 2014.



Google Drive

- ▶ Google Drive offre molte delle funzionalità di Dropbox.
- ▶ É integrato con l'ecosistema Google, come Google Docs, costruito sul OS Web Chromium.
- ▶ Il servizio include:
 - ▶ elaboratore di testi;
 - ▶ foglio elettronico;
 - ▶ elaboratore di presentazioni.
- ▶ É necessario essere utenti Google per accedere al servizio.
- ▶ Si hanno 15GB di spazio gratuiti condivisi tra i vari applicativi Google.



Google Drive

- ▶ Per accedere ai file su Google Drive si può usare:
 - ▶ il sito Web di Google Drive;
 - ▶ applicazioni per sistemi operativi come Mac e Windows;
 - ▶ applicazioni per piattaforme come iOS e Android.
- ▶ Il servizio supporta l'operazione:
 - ▶ drag-and-drop di file e cartelle dal proprio dispositivo allo storage remoto e viceversa;
 - ▶ sincronizzazione dei file sui vari dispositivi utilizzati.

Ulteriori approfondimenti:

<http://www.gcflearnfree.org/googledriveanddocs>

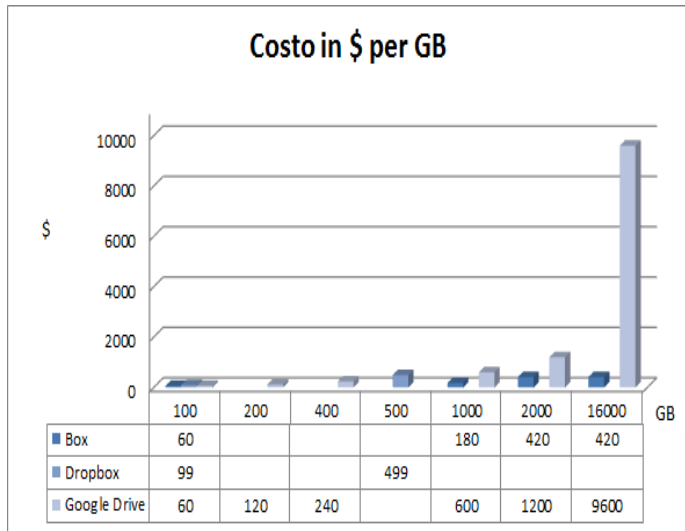


Comparazione tra alcuni OSS⁵

Caratteristica	Box	Dropbox	Google Drive
Massimo storage gratuito	10GB	2GB	15GB
Bonus per extra storage		[500MB, 16GB]	0
Massimo spazio storage comprabile	100GB	500GB	
Massima dimensione del file	250GB	Nessun limite	10GB
Backup continuo	✓	✓	✓
Backup incrementale	✓		
Sincronizzazione dei file	✓	✓	✓
Opzioni per la condivisione	✓	✓	✓
Permessi sulle cartelle	✓	✓	✓
Sincronizzazione con cellulare	✓	✓	✓
Piattaforme supportate	Windows, Mac, Android, BlackBerry, iOS	Windows, Mac, Linux, Android, iOS, BlackBerry, Kindle, Filre	Windows, Mac, iOS, Android

⁵Dati aggiornati a metà 2014.

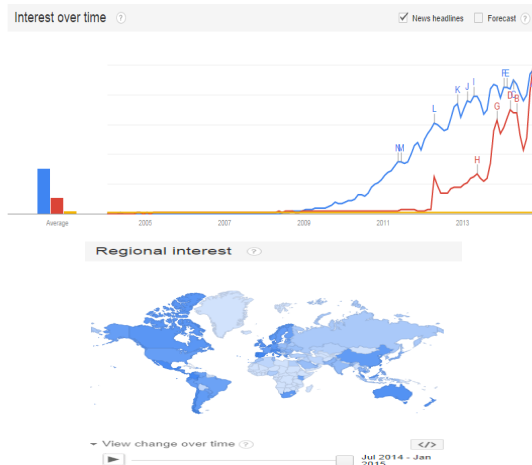
Comparazione tra alcuni OSS rispetto al costo⁶



⁶Dati aggiornati a metà 2014.

Comparazione tra alcuni OSS⁷

- ▶ Tramite Google Trends.
<https://google.com/trends/>



⁷Dati aggiornati a metà 2014.



Per ulteriori letture



Scott. Chacon, *Pro Git*, Apress (2009),
<http://git-scm.com/book>



Jon Loeliger, Matthew McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development Paperback*, O'Reilly Media; Second Edition edition (August 27, 2012)