Spiegare ruolo di “origin”

* “Origin" rappresenta il nome del repository remoto in cui si desidera pubblicare il commit. Per convenzione, il repository remoto predefinito è denominato" origin ", ma è possibile lavorare con più repository remoti (con nomi diversi) allo stesso tempo.

Prendere dimestichezza con i comandi visti

Comandi per creazione repository da riga di comando

* Se si vuole iniziare a tenere traccia con Git di un progetto esistente, bisogna andare nella directory del progetto e digitare:

$ git init

Questo creerà una nuova sottodirectory chiamata .git che conterrà tutti i file necessari per il proprio repository, una struttura del repository Git. A questo punto non è ancora stato tracciato niente del tuo progetto. Per iniziare a tracciare i file esistenti, è necessario iniziare a monitorare questi file con una commit iniziale. La procedura è la seguente:

$ git add \*.c

$ git add README

$ git commit -m 'initial project version'

Se si vuole copiare un repository Git esistente,per esempio, un progetto a cui si vuole contribuire, il comando necessario è git clone. Quando si esegue questo comando vengono scaricate tutte le versioni di ciascun file della cronologia del progetto. Infatti, se si danneggiasse il disco del server, si può usare qualsiasi clone di qualsiasi client per ripristinare il server allo stato in cui era quando è stato clonato

Si può clonare un repository con git clone [url].

Questo comando crea un directory "grit", dentro di questa inizializza una directory.git, scarica tutti i dati del repository e fa il checkout dell'ultima versione per poterci lavorare su. Nella nuova directory grit si trovano tutti i file del progetto, pronti per essere modificati o usati.

Eseguire una git push force e spiegare le conseguenze di tale azione

* Il push force consente di pushare in remoto repository creata ex novo in locale, però può portare dei rischi perche fondamentalmente sovascrive la storia del branch remoto con la perdita dei commit pushate precedentemente.

Ogni commit ha un id univoco. Documentatevi sul come tale id viene assegnato. (non è necessario scendere troppo nel dettaglio).

* Ogni commit e' identificato da un checksum SHA. Questa e' una particolarità di **Git** che lo differenzia da molti altri prodotti simili. Semplificando si può dire che ogni commit ha un ID unico che permette di distinguerlo dagli altri. L'ID è molto lungo, è un codice di 40 cifre esadecimale, ma in genere viene mostrata solo una piccola parte

Spiegare differenza fra una revert e una reset, con relativi esempi

* Può capitare quando si utilizza Git di lanciare commit errate, o di pusciare una commit su un branch errato. Se non è stato effettuato ancora il comando push verso una repository remota, si può usare il comando git reset. In questo modo si rimuovono i file dallo stage; ovvero si copiano i file dell'ultimo commit nello stage. Questo comando annulla un git add file. Si può anche utilizzare git reset per rimuovere tutto dallo stage.

Se al contrario il comando push è stato lanciato, bisogna utilizzare il comando git revert, in tal caso viene creata una nuova commit che annulla tutto ciò che è stato introdotto o modificato dalla commit indesiderata.

Spiegare i due metodi di integrazione branch visti finora, con relativi esempi

* Citiamo un esempio, utilizziamo un ramo vuln-1 per lavorare, ad esempio, alla risoluzione di una vulnerabilità del nostro progetto. Una volta risolte le vulnerabilità non sarà più neccessario il ramo “vuln-1” e potremo ricorrere al merging per fonderlo con il ramo develop; riposizioneremo quindi il puntatore su quest’ultimo e utilizzeremo l’istruzione git merge seguita dal nome del branch che vogliamo unire al develop, in tal caso riportiamo sul ramo develop tutte le modifiche e quindi la risoluzione della vulnerabilità.
* Con il comando git rebase puoi prendere tutti i cambiamenti che sono stati inviati su un ramo ed applicarli su un altro. Se si esamina il log del ramo su cui è stato fatto il rebase, assomiglia ad una storia lineare: appare come se tutto il lavoro fosse stato fatto in serie, invece è stato fatto in parallelo, mantenendo più pulita la storia dei commit sul ramo remoto.