

Etckeeper

Teniamo sotto controllo i file di configurazione!



Uno strumento utile per la gestione condivisa di qualsiasi tipo di server e' **etckeeper**. Questa e' una collezione di script in bash che consentono di tener sotto controllo tramite un sistema di revisione distribuito i file di configurazione presenti nella directory /etc di un sistema *nix. Si potrebbe pensare di non utilizzare etckeeper e tener in revisione i file semplicemente utilizzando il sistema a noi piu' congeniale. Prima di tutto e' di fondamentale importanza preservare non solo i permessi, ma anche utenti e gruppi a cui i file appartengono. Oltre a questo alcuni sistemi di revisione non registrano le directory vuote (in tal caso git). Etckeeper evita all'amministratore di sistema di doversi preoccupare di tutto questo e automatizza il processo di gestione.

Prima caratteristica fra' tutte e' quella di non limitare l'utente a dover utilizzare uno specifico sistema di revisione, infatti supporta i piu' diffusi: bzd, mercurial, git e darcs. Da un'installazione di default quello utilizzato e' bzd.

L'integrazione con apt-get rende la revisione automatizzata anche nel gestire i software installati sul sistema.

Per installare etckeeper su una distribuzione debian-like si procede come segue:

aptitude -y install etckeeper

Aptitude provvedera' a installare tutti i pacchetti necessari per far funzionare etckeeper ed effettuera' il primo commit utilizzando bzd. Se si ha intenzione di non

utilizzare quest'ultimo, ma ad esempio git, e' necessario installare il pacchetto specifico (in questo caso git) ed eliminare il primo commit con il seguente comando:

```
# etckeeper uninit
```

```
# aptitude install git
```

Oltre ad installare il sistema di revisione desiderato si deve indicare a etckeeper di utilizzare proprio quello e quindi si procede modificando alcune variabili nel file di configurazione:

```
/etc/etckeeper/etckeeper.conf .
```

Un'altra variabile interessante presente nel file di configurazione e' **AVOID_DAILY_AUTOCOMMITS**.

Da installazione pulita etckeeper effettua commit giornalieri per registrare le modifiche in maniera organizzata. Se si toglie il commento a questa variabile non saranno piu' effettuati.

A questo punto si e' pronti ad effettuare il primo commit:

```
# etckeeper init
```

```
== /etc/etckeeper/etckeeper.conf ==

# The VCS to use.
# VCS="hg"
VCS="git"
# VCS="bzd"
# VCS="darcs"

# Options passed to git commit when run by etckeeper.
#GIT_COMMIT_OPTIONS=""

# Options passed to hg commit when run by etckeeper.
#HG_COMMIT_OPTIONS=""

# Options passed to bzd commit when run by etckeeper.
#BZD_COMMIT_OPTIONS=""

# Options passed to darcs commit when run by etckeeper.
#DARCS_COMMIT_OPTIONS=""

# Uncomment to avoid etckeeper committing existing changes
# to /etc automatically once per day.
#AVOID_DAILY_AUTOCOMMITS=1

# Uncomment to avoid etckeeper committing existing changes to
# /etc before installation. It will cancel the installation,
# so you can commit the changes by hand.
#AVOID_COMMIT_BEFORE_INSTALL=1

# The high-level package manager that's being used.
# (apt, pacman-g2, yum etc)
HIGHLEVEL_PACKAGE_MANAGER=apt

# The low-level package manager that's being used.
# (dpkg, rpm, pacman-g2, etc)
LOWLEVEL_PACKAGE_MANAGER=dpkg

== /etc/etckeeper/etckeeper.conf ==
```

Etckeeper - <http://joey.kitenet.net/code/etckeeper/>

AlcaLabs - <http://labs.alcacoop.it/doku.php?id=docs:etckeeper>



Etckeeper

Teniamo sotto controllo i file di configurazione!

Utilizzando bzd con dei semplici comandi si puo' risalire subito alle varie revisioni e allo stato del repository:

```
# bzd status /etc/
```

```
# bzd log /etc
```

BZR
bzd init
bzd status
bzd ignore PATTERN
bzd [add rm mv] FILENAME [DEST]
bzd commit
bzd log [-r REV_START...REV_STOP]
bzd diff [-r REV_START...REV_STOP]

Ovviamente essendo un repository bzd si puo' tranquillamente usare qualsiasi comando che questo sistema di revisione permette, specificando ovviamente sempre la cartella /etc.

Nel momento in cui si effettuano modifiche manualmente ai file della cartella /etc e si desidera aggiungere un commento non automatico per descrivere il lavoro fatto, si puo' procedere a lanciare un commit in questo modo:

```
# etckeeper commit "Modifica del file resolv.conf"
```

La peculiarita' di etckeeper e' la sua estendibilita' fino ai massimi livelli. Infatti l'installazione di default crea una cartella /etc/etckeeper che contiene vari script gia' ben strutturati che possono

```

/etc/etckeeper/
|-- commit.d
| |-- 10vcs-test
| |-- 30bzd-add
| |-- 30darcs-add
| |-- 30git-add
| |-- 30hg-addremove
| |-- 40git-rm
| |-- 50vcs-commit
| `-- README
|-- etckeeper.conf
|-- init.d
| |-- 10restore-metadata
| |-- 20restore-etckeeper
| |-- 40vcs-init
| |-- 50vcs-ignore
| |-- 50vcs-perm
| |-- 50vcs-pre-commit-hook
| |-- 60darcs-deleted-symlinks
| |-- 70vcs-add
| `-- README
|-- list-installed.d
| `-- 50list-installed
|-- post-install.d
| |-- 50vcs-commit
| `-- README
|-- pre-commit.d
| |-- 20warn-hardlinks
| |-- 20warn-special-file
| |-- 30store-metadata
| `-- README
|-- pre-install.d
| |-- 10packagelist
| |-- 50uncommitted-changes
| `-- README
|-- unclean.d
| |-- 50test
| `-- README
|-- uninit.d
| |-- 01prompt

```

```

| |-- 50remove-metadata
| |-- 50vcs-uninit
| `-- README
`-- update-ignore.d
| |-- 01update-ignore
| `-- README

```

Ogni cartella e' denominata con l'aggiunta del suffisso ".d" e identifica il parametro da passare ad etckeeper per far si che esegua tutti gli script presenti in quella cartella. Ad esempio, digitando il comando "etckeeper commit" stiamo dicendo ad etckeeper di eseguire tutti gli script contenuti nella cartella /etc/etckeeper/commit.d. Da cio' si deduce che e' semplice ampliare le funzionalita' dello stesso solamente creando una nostra cartella denominata "comando.d" e inserire tutte le operazioni di cui abbiamo bisogno tramite appositi script ed utilizzare la chiamata "etckeeper comando" quando sara' necessario eseguire quei particolari task all'interno del sistema.

Prima si e' accennato al fatto che etckeeper instaura un hook (aggancio) all'interno del sistema di pacchettizzazione delle distribuzioni debian-like (apt). Cio' non esclude il fatto che riesca a funzionare anche su altre distribuzioni come redhat, suse, interfacciandosi a yum, pacman e quant'altro in modo da intercettare pacchetti installati, rimossi e file di configurazione che vengono modificati, il tutto in maniera del interamente automatica.

Messi in sicurezza i vostri file di configurazione potrete godervi una birra ghiacciata comodamente in poltrona ;-)