Comandi base SSH

I comandi principali sono : **ssh** (che sostituisce **telnet** ed **rsh**) e **scp** (che sostitusce l'**ftp**)

Dalla linea di comando :

– **ssh *nomeserver* -l *username***
per collegarsi a *nomeserver* con la username specificata

– **ssh *nomeserver* -l *username comando_da_eseguire***
per eseguire *comando* su *nomeserver*

– **scp *nomefile username@nomeserver:/percorso/dove/mettere/il/file***
per trasferire *nomefile* VERSO *nomeserver*

– **scp *username@nomeserver:/dove/si/trova/il/file* .**
per trasferire *nomefile* DA *nomeserver* nella directory corrente (il punto)

Mini tutorial su SSH e SCP

Se state leggendo quest'articolo significa che avete già avuto a che fare con un server SSH, ma per completezza cerco spiegare meglio il classico scenario:

Io voglio avere una shell di comando su un server remoto, nella quale eseguire dei comandi e controllare quindi il mio server.

Usando windows potrei utilizzare PUTTY per impartire i comandi e WINSCP per gestirne i files in maniera grafica e molto intuitiva, ma se usassi linux?

Beh .. come primo passo mi devo collegare al mio server remoto e ipotizzando che il suo indirizzo IP sia 134.34.54.20 e io sia l'utente pippo in un terminale linux qualsiasi digiterei questo comando:

```
# ssh pippo@134.34.54.20
```

se la porta non fosse quella classica (22) allora dovrei specificarla con l'opzione -p , esempio :

```
# ssh pippo@134.34.54.20 -p 2222
```

dove 2222 è la porta su cui il mio server SSH rimane in ascolto.

dopo di che inserisco la password ed il gioco è fatto.

Ma se volessi copiare o trasferire dei file dal mio pc al server e viceversa?

La soluzione è Secure Copy (copia sicura) detto anche SCP, il quale è un mezzo per trasferire in modo sicuro un file del computer tra un computer locale ed un host remoto o tra due host remoti, usando il protocollo Secure Shell (SSH).

Tipicamente, la sintassi del programma **scp** è la stessa del comando **cp** che serve per

copiare i file in locale.
Per copiare un file dal mio pc verso il server remoto, dovrei utilizzare il seguente formato:

scp FileSorgente nomeUtente@host:directory/FileDestinazione

Al contrario se volessi copiare un file da remoto nel mio computer:

scp nomeUtente@host:directory/FileSorgente FileDestinazione

Quindi nel mio caso se volessi copiare un documento in una directory sul server remoto :

scp miodocumento.txt pippo@134.34.54.20:/home/pippo/miodocumento.txt

Se invece volessi fare il contrario cioè copiare un file dal server remoto nel mio pc :

scp pippo@134.34.54.20:/home/pippo/miodocumento.txt miodocumento.txt

Se volessi trasferire tutti i file cartella "sorgente" del pc remoto alla cartella "destinazione" del pc locale:

scp pippo@134.34.54.20:/home/pippo/sorgente/* /home/pluto/destinazione/

Se la porta del server non fosse la 22 per specificarla dovrei usare l'opzione **-P** (attenzione al P deve essere maiuscola), esempio:

 # scp -P 2222 miodocumento.txt pippo@134.34.54.20:/home/pippo/miodocumento.txt

Per non consumare tutta la banda, dato che in caso di invio dati e con una ADSL velocità di trasmissione è molto limitata, potrei utilizzare l'opzione "**-l**" seguita dal numeri di Kbit da utilizzare, notare che parlo di Kbit non di Kbyte quindi se impongo "-l 240" intendo limitare la banda a 240 Kbit che corrispondono a 240/8=30KByte.

Altre opzioni utili sono "**-r**" con il quale si permette la ricerca ricorsiva anche nelle subdirectory, molto utile se si cerca di copiare una cartella che ne contiene a sua volta delle altre, e l'opzione "**-C**" che abilita la compressione dei dati trasmessi.

Comandi veloci SSH

Non sono molto ferrato in materia ma devo ammettere che quando si lavora in remoto, su macchine virtuali o sui Grid Hosting di MT, risulta molto comodo avere conoscenza basilare di alcuni dei più comuni **comandi unix da eseguire in SSH**.

Iniziamo dalla connessione alla macchina remota:

MacPro:~ user$ ssh utente@server

Mentre per scaricare un file, sempre dal terminale della macchina locale:

MacPro:~ user$ scp utente@server:~/domains/dominio.it/nomedelfile.txt

Visualizzare i file di una cartella con tutte le informazioni:

MacPro:~ user$ ls -alh

Una volta collegati in ssh alla macchina remota potete aprire un file in lettura in modo che eventuali modifiche al file vengano immediatamente visualizzate a schermo (utile ad esempio per visualizzare log in tempo reale):

utente@server:~$ tail -f -n 10 /var/adm/syslog

Fare il backup veloce di una cartella:

utente@server:~$ utente@server:~$ tar -cvpzf ~/domains/dominio.it/backup.tar.gz html/

Per decomprimere il backup invece:

tar -xzvf ~/domains/dominio.it/backup.tar.gz

Fare il backup di una database MySql:

utente@server:~$ mysqldump -uUtente -pPassword -h Server NomeDatabase > sqlbackup.sql

Effettuare una query al db MySql:

utente@server:~$ mysql -uUtente -pPassword -h Server

> show databases;

> use NomeDatabase;

> show tables;

> select * from nome_tabella where id = 1;

Visualizzare la versione di PHP:

utente@server:~$ php -V

Visualizzare altre informazioni su php:

utente@server:~$ php -i | grep -e "memory_limit"

Mostrare tutte le informazioni su Apache:

utente@server:~$ httpd -V

oppure:

utente@server:~$ apachectl -v

Visualizzare i moduli installati su Apache:

utente@server:~$ httpd -M

oppure:

utente@server:~$ apachectl -M

Modificare un file (se protetto aggiungere *sudo* all'inizio del comando):

utente@server:~$ nano /etc/hosts

Visualizzare i processi attualmente attivi:

utente@server:~$ ps aux

Visualizzare il consumo di risorse dei processi attivi in tempo reale:

utente@server:~$ top

Chiudere un processo attivo (sostituire pid con il numero di processo):

utente@server:~$ kill pid

Quanta memoria ram libera ha il sistema?

utente@server:~$ free

Visualizzare le connessioni attive in tempo reale:

utente@server:~$ netstat -t -u -c | grep ESTABLISHED

FPT (via SSH)

questo è anche semplice se ci pensiamo, ma a volte non c'è il tempo di pensarci su, quindi

`utente@server:ftp utente@serverremoto.ext`
immettere la password al prompt poi:

`lcd`
per vedere la cartella corrente in locale (se sei collegato in ssh, è il server ssh ovviamente)

`lcd nome_cartella_locale`
per spostarti nella cartella locale

`prompt no`
per evitare di confermare ogni file da trasferire

`hash`
se vuoi vedere il progresso di caricamento del file con tanti ####

per vedere le cartelle del server ftp (quello a cui ti sei collegato)

`cd nome_cartella_remota`
per aprire una cartella remota

`mget *`
per prelevare dal server remoto tutti i file. Oppure anche

`mget *.jpg`
per prelevare solo i .jpg (ma anche *_big.jpg, per esempio)

```
mput *
```
per trasferire dal locale al remoto, secondo le regole di cui sopra.

Rinominare in batch tanti file

per esempio, se togliere "_big" da tutti i file .jpg che ho sul server (es: immagine1_big.jpg e rinominarla in immagine1.jpg)

```
find . -name "*_big.jpg" -exec sh -c 'mv "$1" "${1%_big.jpg}.jpg"' _ {} \;
```

Convertire da MAIUSCOLO a minuscolo tutti i file di una cartella

Dai ammettiamolo: 'sta storia che in linux "FILE.jpg" è diverso da "file.jpg" è una gran rottura, ma se c'è, a qualcuno servirà. Io per intanto preferisco uniformare tutti i file in minuscolo, e per farlo a volte devo riscrivere tutti i nomi file. Ma c'è la possibilità di farlo fare al terminale/shell

```
for a_file in *;do mv -v $a_file `echo $a_file | tr [:upper:] [:lower:]` ;done;
```
(sia in linux che in OSX, questo funziona.)

Come funziona ssh (mediante esempi)

Al fine di eseguire un comando su un host server remoto la sintassi basilare è la seguente:

```
ssh [USERNAME]@[HOST-REMOTO] [comando o script]
```

dove il significato dei vari termini è il seguente:

1. **ssh**: il client *ssh* non è altro che un programma per fare login all'interno di una macchina remota ed eseguire comandi di vario genere.
2. **[USERNAME]**: la username dell'utente sull'host remoto del server in questione
3. **[HOST-REMOTO]**: indirizzo IP o nome dell'host coinvolto, ad esempio nome.abc.est.
4. **[comando o script]**: il nome del comando (con eventuali parametri) e dello script che sarà eseguito sulla macchina remota.

Vediamo di seguito una serie di esempi pratici che potete adattare alle vostre necessità.

Esempi pratici

Di seguito riportiamo alcune delle istruzioni che si possono eseguire mediante ssh, ovvero terminale che usualmente utilizziamo su Linux/Unix mediante accesso ad una macchina remota. Si noti come le operazioni siano tipicamente vincolate all'inserimento della password di accesso, che viene richiesta di solito dopo aver inviato il comando. In certi casi, inoltre, la procedura di autenticazione può funzionare con meccanismo a chiave pubblica / privata (crittografia asimmetrica).

Preleva informazioni da un disco remoto sull'host www1.host123.com come utente *pippo*, e le mostra in output

```
ssh pippo@www1.host123.com df -h
```

Mostra le porte aperte sull'host remoto come utente *pippo*

```
ssh pippo@www1.host123.com netstat -vatn
```

Esegui un riavvio della macchina remota come utente *root*

```
ssh root@www1.host123.com reboot
```

Riavvia il server mysql sulla macchina remota (utile per molti servizi di hosting)

```
ssh root@www1.host123.com '/etc/init.d/mysql restart'
```

Memorizza nel file *status.txt* nella cartella *tmp* informazioni sulla memoria libera sulla macchina

```
ssh pippo@www1.host123.com 'free -m' > /tmp/status.txt
```

Utilizza una *pipeline* per concatenare ed eseguire istruzioni più complesse

```
ssh pippo@www1.host123.com free -m | grep "Mem:" | awk '{ print "Memoria
disponibile (utilizzata + libera): " $3 " + " $4 " = " $2 }'
```

Prova la connessione.

Prima di buttarti sulla creazione di chiavi sicure e di spostare i file, ti consigliamo di verificare che SSH sia configurato correttamente sul computer, così come il sistema a cui ti connetti. Immetti il seguente comando, sostituendo <username> con il tuo nome utente sul computer remoto e <remote> con l'indirizzo per il computer remoto o server:

- `$ ssh <username>@<remote>`
- Ti verrà chiesta la password una volta stabilita la connessione. Quando digiti la password, non vedrai muoversi il cursore né leggerai i caratteri immessi.
- Se questo passaggio non ti riesce, i casi sono due: o SSH non è configurato correttamente sul tuo computer o il computer remoto non accetta connessioni SSH.

Parte 2

Imparare i Comandi di Base

1

Naviga nella shell SSH. Durante la prima connessione al computer remoto, dovresti trovarti nella tua directory HOME. Per spostarti nella struttura delle directory, utilizza il comando `cd`:

- `cd ..` ti sposterà in su di una directory.
- `cd <nomedirectory>` ti sposterà nella sottodirectory specificata.
- `cd/home/directory/percorso/` ti sposterà nella directory specificata dalla radice (home).
- `cd ~` ti riporterà alla tua directory HOME.

2

Verifica il contenuto della directory corrente. Per vedere quali file e cartelle si trovano nella posizione corrente, puoi utilizzare il comando `ls`:

- `ls` elencherà tutti i file e le cartelle presenti nella directory corrente.
- `ls –l` visualizzerà il contenuto della directory con delle informazioni aggiuntive quali dimensione, autorizzazioni e data.
- `ls-a` elencherà tutti i contenuti, inclusi cartelle e file nascosti.

3

Copia i file dal percorso al computer remoto. Se devi copiare i file dal computer locale a quello remoto, puoi utilizzare il comando `scp`:

- `/directorylocale/esempio1.txt <username>@<remote>:<percorso>` copierà esempio1.txt sul <percorso> nel computer remoto specificato. Puoi lasciare vuoto il <percorso> per copiare sulla cartella radice del computer remoto .
- `scp <username>@<remote>:/home/esempio1.txt ./` sposterà esempio1.txt dalla directory home del computer remoto alla directory corrente di quello locale.

4

Copia i file attraverso la shell. Puoi utilizzare il comando `cp` per fare copie di file nella stessa directory o in una directory di tua scelta:

- `cp esempio1.txt esempio2.txt` creerà una copia di esempio1.txt chiamandolo esempio2.txt nella stessa posizione.
- `cp esempio1.txt <directory>/` creerà una copia di esempio1.txt nella posizione specificata da <directory>.

5

Sposta e rinomina i file. Se desideri cambiare il nome di un file o vuoi spostarlo senza copiarlo, puoi utilizzare il comando `mv`:

- `mv esempio1.txt esempio2.txt` rinominerà esempio1.txt in esempio2.txt. Il file rimarrà nella stessa posizione.
- `mv directory1 directory2` rinominerà directory1 in directory2. Il contenuto della directory rimarrà invariato.
- `mv esempio1.txt directory1/` sposterà esempio1.txt nella directory1.
- `mv esempio1.txt directory1/esempio2.txt` sposterà esempio1.txt nella directory1 e lo rinominerà come esempio2.txt

6

Elimina file e directory. Se devi rimuovere qualcosa dal computer a cui sei connesso, puoi utilizzare il comando `rm`:

- `rm esempio1.txt` cancellarà il file esempio1.txt.
- `rm –I esempio1.txt eliminerà il file esempio1.txt dopo la`

`richiesta della tua conferma.`
- `rm directory1 /` eliminerà la `directory1` e tutti i suoi contenuti.

7

Modifica le autorizzazioni per i tuoi file. Puoi modificare i privilegi di lettura e scrittura dei tuoi file utilizzando il comando `chmod`:

- `chmod u+w esempio1.txt` aggiungerà al file l'autorizzazione di scrittura (modifica) per l'utente (u). Inoltre, puoi utilizzare il modificatore `g` per le autorizzazioni di gruppo oppure `o` per le autorizzazioni globali.
- `chmod g+r esempio1.txt` aggiungerà al file l'autorizzazione di lettura (accesso) per il gruppo.
- C'è un ampio elenco di autorizzazioni da poter utilizzare per fissare o aprire i vari aspetti del tuo sistema.

8

Imparare gli altri diversi comandi di base. Ci sono alcuni comandi più importanti che utilizzerai nell'interfaccia della shell. Essi comprendono:

- `mkdir nuovadirectory` creerà una nuova sottodirectory denominata "nuovadirectory".
- `pwd` visualizzerà il percorso della directory corrente.
- `who` mostra chi viene registrato nel sistema.
- `pico nuovofile.txt` o `vi nuovofile.txt` creerà un nuovo file e aprirà l'editor del file. Ogni diverso sistema avrà installato un differente editor di file. I più comuni sono pico e vi. Potresti dover utilizzare comandi diversi se hai installato un altro editor di file.

9

Cerca di ottenere le informazioni dettagliate su qualsiasi comando. Se non sei sicuro su che cosa farà un comando, puoi sempre utilizzare il comando `man` per conoscere tutti i possibili usi e parametri:

- `man <comando>` visualizzerà informazioni su quel comando.
- `man –k <keyword>` cercherà tutte le pagine di aiuto per la parola chiave specificata.

Parte 3

Creazione di Chiavi Crittografate

1. **Crea le chiavi SSH.** Queste chiavi ti permetteranno di collegarti alla posizione remota senza dover immettere ogni volta la password. Questo è un modo molto più sicuro per connettersi al computer remoto, perché la password non dovrà essere trasmessa in rete.
    - Crea la cartella delle chiavi sul tuo computer inserendo il comando `$ mkdir .ssh`
    - Crea le chiavi pubbliche e private utilizzando il comando `$ ssh-keygen – t rsa`
    - Ti verrà chiesto se desideri creare una passphrase per le chiavi; è opzionale. Se non vuoi

creare una passphrase, premi INVIO e verranno create due chiavi nella directory ssh: id_rsa e id_rsa.pub

- Modifica le autorizzazioni della tua chiave privata. Al fine di garantire che la chiave privata sia leggibile solo dall'utente, dovrai immettere il comando `$ chmod 600 .ssh/id_rsa`

2

Inserisci la chiave pubblica sul computer remoto. Una volta create le chiavi, sei pronto ad inserire la chiave pubblica sul computer remoto in modo da poterti connettere senza una password. Immetti il seguente comando, sostituendo le parti appropriate, come spiegato in precedenza:

- `$ scp .ssh/id_rsa.pub <username>@<remote>:`
- Assicurati di includere i due punti (:) alla fine del comando.
- Ti verrà chiesto di inserire la password prima che inizi il trasferimento del file.

3

Installa la chiave pubblica sul computer remoto. Una volta che hai inserito la chiave nel computer remoto, dovrai installarla in modo che funzioni correttamente. In primo luogo, accedi al computer remoto come hai fatto nel passaggio 3.

- Crea una cartella SSH sul computer remoto, se non esiste già: `$ mkdir .ssh`
- Accoda la chiave al file delle chiavi autorizzate. Se il file non esiste ancora, verrà creato: `$ cat id_rsa.pub >> .ssh/authorized_keys`
- Modifica le autorizzazioni per la cartella SSH per consentire l'accesso:
- `$ chmod 700 .ssh`

4

Verifica che la connessione funzioni. Una volta che la chiave è stata installata sul computer remoto, dovresti essere in grado di avviare una connessione senza che ti si chieda di inserire la password. Immetti il seguente comando per verificare la connessione: `$ ssh <username>@<remote>`

- Se ti connetti senza che ti sia richiesta la password, allora le chiavi sono configurate correttamente.