Learning Rsync On Rocky Linux (Italian version)

A book from the Documentation Team

Version : 2025/09/27

Rocky Documentation Team

Table of contents

1. Licence

RockyLinux offers Linux courseware for trainers or people wishing to learn how to administer a Linux system on their own.

RockyLinux materials are published under Creative Commons-BY-SA. This means you are free to share and transform the material, while respecting the author's rights.

BY : **Attribution**. You must cite the name of the original author.

SA : **Share Alike**.

- Creative Commons-BY-SA licence : https://creativecommons.org/licenses/by-sa/4.0/

The documents and their sources are freely downloadable from:

- https://docs.rockylinux.org
- https://github.com/rocky-linux/documentation

Our media sources are hosted at github.com. You'll find the source code repository where the version of this document was created.

From these sources, you can generate your own personalized training material using mkdocs. You will find instructions for generating your document here.

How can I contribute to the documentation project?

You'll find all the information you need to join us on our git project home page.

We wish you all a pleasant reading and hope you enjoy the content.

2. Backup in Sintesi

Che cos'è un backup?

Il backup si riferisce alla duplicazione dei dati nel file system o nel database. In caso di errore o disastro, i dati effettivi del sistema possono essere ripristinati in modo tempestivo per un normale funzionamento.

Quali sono i metodi di backup?

- Backup completo: si riferisce a una copia unica di tutti i file, le cartelle o i dati nel disco rigido o nel database. (Pro: il migliore, può recuperare i dati più velocemente. Svantaggi: occupa più spazio su disco rigido.)
- Backup incrementale: si riferisce al backup dei dati aggiornati dopo l'ultimo backup completo o incrementale. Il processo è come questo: un backup completo il primo giorno; un backup dei dati appena aggiunti il secondo giorno, al contrario di un backup completo; il terzo giorno, un backup dei dati appena aggiunti sulla base del secondo giorno, relativo al giorno successivo, e così via.
- Backup differenziale: Si riferisce al backup dei file modificati dopo il backup completo. Ad esempio, un backup completo il primo giorno; un backup dei nuovi dati il secondo giorno; un backup dei nuovi dati dal secondo giorno al terzo giorno del terzo giorno; e un backup di tutti i nuovi dati dal secondo giorno al quarto giorno il quarto giorno.
- Backup selettivo: Si riferisce al backup di una parte del sistema.
- Backup a freddo: si riferisce al backup quando il sistema è in stato di arresto o manutenzione. I dati di backup sono esattamente gli stessi dei dati nel sistema durante questo periodo.
- Hot backup: si riferisce al backup quando il sistema è in funzionamento normale. Poiché i dati nel sistema vengono aggiornati in qualsiasi momento, i dati di backup hanno un certo ritardo rispetto ai dati reali del sistema.
- Backup remoto: si riferisce al backup dei dati in un'altra posizione geografica per evitare la perdita di dati e l'interruzione del servizio causata da incendi, disastri naturali, furti, ecc.

2.1 rsync in sintesi

Su un server, ho eseguito il backup della prima partizione nella seconda partizione, comunemente nota come "Backup locale." Gli strumenti di backup specifici da poter utilizzare sono `tar` , `dd` , `dump` , `cp`, ecc. Anche se i dati vengono salvati su questo server, se l'hardware non riesce ad avviarsi correttamente, i dati non verranno recuperati. Per risolvere questo problema con il backup locale, abbiamo introdotto un altro tipo di backup --- "backup remoto".

Alcune persone diranno, non posso usare il comando `tar` o `cp` sul primo server e inviarlo al secondo server tramite `scp` o `sftp`?

In un ambiente di produzione, la quantità di dati è relativamente elevata. Prima di tutto, `tar` o `cp` consumano molto tempo e occupano le prestazioni del sistema. La trasmissione tramite `scp` o `sftp` occupa anche un sacco di larghezza di banda di rete, che non è consentito nell'ambiente di produzione reale. In secondo luogo, questi comandi o strumenti devono essere inseriti manualmente dall'amministratore e devono essere combinati con il crontab dell'attività pianificata. Tuttavia, il tempo impostato da crontab non è facile da cogliere, e non è opportuno eseguire il backup dei dati se il tempo è troppo breve o troppo lungo.

Pertanto, è necessario un backup dei dati nell'ambiente di produzione che deve soddisfare i seguenti requisiti:

1. Backup trasmessi attraverso la rete
2. Sincronizzazione dei file dati in tempo reale
3. Meno occupazione delle risorse del sistema e maggiore efficienza

`rsync` sembrava soddisfare le esigenze di cui sopra. Utilizza l'accordo di licenza open source GNU. Si tratta di uno strumento di backup incrementale veloce. L'ultima versione è la 3.2.3 (2020-08-06). Puoi visitare il Sito ufficiale per maggiori informazioni.

In termini di supporto alle piattaforme, la maggior parte dei sistemi Unix-like sono supportati, sia che si tratti di GNU/Linux o BSD. Inoltre, ci sono i relativi `rsync` sotto la piattaforma Windows, come cwRsync.

L'originale `rsync` è stato mantenuto dal programmatore australiano Andrew Tridgell (mostrato nella figura 1 qui sotto), ed ora è mantenuto da Wayne Davison (mostrato nella figura 2 qui sotto) Per la manutenzione, puoi andare all'indirizzo del progetto github per ottenere le informazioni che vuoi.

Nota

rsync è solo uno strumento di backup incrementale e non ha la funzione di sincronizzazione dei dati in tempo reale (deve essere integrato da altri programmi). Inoltre, la sincronizzazione è unidirezionale. Se vuoi realizzare la sincronizzazione bidirezionale, devi collaborare con altri strumenti.

2.1.1 Principi e funzionalità di base

Come fa `rsync` a realizzare un efficiente backup di sincronizzazione dei dati unidirezionale?

Il core di `rsync` è il suo **algoritmo di Checksum**. Se sei interessato, puoi andare a [Come funziona Rsync](https://rsync.samba.org/how-rsync-works. html) e [L'algoritmo rsync](https://rsync.samba.org/tech_report/) per ulteriori informazioni. Questa sezione è al di là della competenza dell'autore e non sarà trattata troppo.

Le caratteristiche di `rsync` sono:

- L'intera directory può essere aggiornata ricorsivamente;
- Può mantenere selettivamente gli attributi di sincronizzazione dei file, come hard link, soft link, proprietario, gruppo, autorizzazioni corrispondenti, tempo di modifica, ecc. e può conservare alcuni degli attributi;
- Supporta due protocolli per la trasmissione, uno è il protocollo ssh, l'altro è il protocollo rsync

3. Premessa

`rsync` deve eseguire l'autenticazione dell'utente prima della sincronizzazione dei dati. **Ci sono due metodi di protocollo per l'autenticazione: protocollo SSH e protocollo rsync (la porta predefinita del protocollo rsync è la 873)**

- Metodo di verifica del protocollo SSH: usa il protocollo SSH come base per l'autenticazione dell'identità utente (cioè, utilizza l'utente del sistema e la password di GNU/Linux stesso per la verifica), e quindi esegue la sincronizzazione dei dati.
- metodo di verifica del protocollo rsync login: usa il protocollo rsync per l'autenticazione dell'identità utente (utenti di sistema non GNU/Linux, simili agli utenti virtuali vsftpd), e quindi esegue la sincronizzazione dei dati.

Prima della dimostrazione specifica della sincronizzazione rsync, è necessario utilizzare il comando `rsync`. In Rocky Linux 8, il pacchetto rpm rsync è installato per impostazione predefinita, e la versione è la 3.1.3-12, come segue:

```
[root@Rocky ~]# rpm -qa|grep rsync
rsync-3.1.3-12.el8.x86_64
```

```
Basic format: rsync [options] original location target location
Commonly used options:
-a: archive mode, recursive and preserves the attributes of the file object,
which is equivalent to -rlptgoD (without -H, -A, -X)
-v: Display detailed information about the synchronization process
-z: compress when transferring files
-H: Keep hard link files
-A: retain ACL permissions
-X: retain chattr permissions
-r: Recursive mode, including all files in the directory and subdirectories
-l: still reserved for symbolic link files
-p: Permission to retain file attributes
-t: time to retain file attributes
-g: retain the group belonging to the file attribute (only for super users)
-o: retain the owner of the file attributes (only for super users)
-D: Keep device files and other special files
```

Uso personale dell'autore:

`rsync -avz posizione originale posizione di destinazione`

3.1 Descrizione dell'Ambiente

Elemento	Descrizione
Rocky Linux 8(Server)	192.168.100.4/24
Fedora 34(client)	192.168.100.5/24

È possibile utilizzare Fedora 34 per caricare e scaricare

```
graph LR;
RockyLinux8-->|pull/download|Fedora34;
Fedora34-->|push/upload|RockyLinux8;
```

È inoltre possibile utilizzare Rocky Linux 8 per caricare e scaricare

```
graph LR;
RockyLinux8-->|push/upload|Fedora34;
Fedora34-->|pull/download|RockyLinux8;
```

3.2 Dimostrazione basata sul protocollo SSH

Suggerimento

Qui, sia Rocky Linux 8 che Fedora 34 utilizzano l'utente root per accedere. Fedora 34 è il client e Rocky Linux 8 è il server.

3.2.1 pull/scarica

Dato che si basa sul protocollo SSH, creiamo prima un utente nel server:

```
[root@Rocky ~]# useradd testrsync
[root@Rocky ~]# passwd testrsync
```

Sul lato client, lo tiriamo/scarichiamo e il file sul server è /rsync/aabbcc

```
[root@fedora ~]# rsync -avz testrsync@192.168.100.4:/rsync/aabbcc /root
testrsync@192.168.100.4 ' s password:
receiving incremental file list
aabbcc
sent 43 bytes received 85 bytes 51.20 bytes/sec
total size is 0 speedup is 0.00
[root@fedora ~]# cd
```

```
[root@fedora ~]# ls
aabbcc
```

Trasferimento effettuato con successo.

> **Suggerimento**
>
> Se la porta SSH del server non è quella predefinita la 22, puoi specificare la porta in modo simile ---`rsync -avz -e 'ssh -p [port]'`.

3.2.2 push/carica

```
[root@fedora ~]# touch fedora
[root@fedora ~]# rsync -avz /root/* testrsync@192.168.100.4:/rsync/
testrsync@192.168.100.4 ' s password:
sending incremental file list
anaconda-ks.cfg
fedora
rsync: mkstemp " /rsync/.anaconda-ks.cfg.KWf7JF " failed: Permission denied
(13)
rsync: mkstemp " /rsync/.fedora.fL3zPC " failed: Permission denied (13)
sent 760 bytes received 211 bytes 277.43 bytes/sec
total size is 883 speedup is 0.91
rsync error: some files/attrs were not transferred (see previous errors) (code
23) at main.c(1330) [sender = 3.2.3]
```

Richiesta di autorizzazione negata, come gestirla?

Prima controlla i permessi della directory /rsync/. Ovviamente, non c'è alcun permesso "w". Possiamo utilizzare `setfacl` per dare il permesso:

```
[root@Rocky ~ ] # ls -ld /rsync/
drwxr-xr-x 2 root root 4096 November 2 15:05 /rsync/
```

```
[root@Rocky ~ ] # setfacl -mu:testrsync:rwx /rsync/
[root@Rocky ~ ] # getfacl /rsync/
getfacl: Removing leading ' / ' from absolute path names
# file: rsync/
# owner: root
# group: root
user::rwx
user:testrsync:rwx
group::rx
```

```
mask::rwx
other::rx
```

Prova di nuovo, successo!

```
[root@fedora ~ ] # rsync -avz /root/* testrsync@192.168.100.4:/rsync/
testrsync@192.168.100.4 ' s password:
sending incremental file list
anaconda-ks.cfg
fedora
sent 760 bytes received 54 bytes 180.89 bytes/sec
total size is 883 speedup is 1.08
```

4. Dimostrazione basata sul protocollo rsync

In vsftpd, ci sono utenti virtuali (utenti impersonali personalizzati dall'amministratore) perché non è sicuro usare utenti anonimi e utenti locali. Sappiamo che un server basato sul protocollo SSH deve garantire che ci sia un sistema di utenti. Quando ci sono molti requisiti di sincronizzazione, può essere necessario creare molti utenti. Questo ovviamente non soddisfa gli standard di gestione e manutenzione GNU/Linux (più utenti, più insicuro), in rsync, per motivi di sicurezza, c'è un metodo di autenticazione il protocollo rsync.

Come farlo?

È sufficiente scrivere i parametri e i valori corrispondenti nel file di configurazione. In Rocky Linux 8, è necessario creare manualmente il file /etc/rsyncd.conf.

```
[root@Rocky ~]# touch /etc/rsyncd.conf
[root@Rocky ~]# vim /etc/rsyncd.conf
```

Alcuni parametri e valori di questo file sono i seguenti, qui si trovano altre descrizioni dei parametri:

Elemento	Descrizione
address = 192.168.100.4	L'indirizzo IP dove rsync è in ascolto di default
port = 873	porta rsync di ascolto predefinita
pid file = /var/run/rsyncd.pid	Posizione del file del pid del processo
log file = /var/log/rsyncd.log	Posizione del file del registro
[share]	Nome condiviso
comment = rsync	Osservazioni e informazioni sulla descrizione
path = /rsync/	Posizione del percorso di sistema in cui si trova
read only = yes	sì significa solo leggere, non leggere e scrivere
dont compress = *.gz *.gz2 *.zip	Quali tipi di file non comprimere
auth users = li	Abilita gli utenti virtuali e definisci come viene chiamato un utente virtuale. Devi crearlo da solo
secrets file = /etc/rsyncd_users.db	Usato per specificare la posizione del file password dell'utente virtuale, che deve terminare in .db. Il formato del contenuto del file è "Nome utente: Password", uno per riga

Suggerimento!

L'autorizzazione del file della password deve essere 600.

Scrivi il contenuto del file su /etc/rsyncd.confe scrivi il nome utente e la password su /etc/rsyncd_users.db, il permesso è 600

```
[root@Rocky ~]# cat /etc/rsyncd.conf
address = 192.168.100.4
port = 873
pid file = /var/run/rsyncd.pid
log file = /var/log/rsyncd.log
[share]
comment = rsync
path = /rsync/
read only = yes
dont compress = *.gz *.bz2 *.zip
auth users = li
secrets file = /etc/rsyncd_users.db
[root@Rocky ~]# ll /etc/rsyncd_users.db
-rw------- 1 root root 9 November 2 16:16 /etc/rsyncd_users.db
[root@Rocky ~]# cat /etc/rsyncd_users.db
li:13579
```

Potrebbe essere necessario `dnf -y install rsync-daemon` prima di poter avviare il servizio: `systemctl start rsyncd.service`

```
[root@Rocky ~]# systemctl start rsyncd.service
[root@Rocky ~]# netstat -tulnp
Proto Recv-Q Send-Q Local Address           Foreign Address
State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*
LISTEN      691/sshd
tcp        0      0 192.168.100.4:873       0.0.0.0:*
LISTEN      4607/rsync
tcp6       0      0 :::22                   :::*
LISTEN      691/sshd
udp        0      0 127.0.0.1:323           0.
0.0.0:*                           671/chronyd
udp6       0      0 ::
1:323                :::*                                 671/chronyd
```

4.1 pull/scarica

Crea un file nel server per la verifica: `[root@Rocky]# touch /rsync/rsynctest.txt`

Il client fa quanto segue:

```
[root@fedora ~]# rsync -avz li@192.168.100.4::share /root
Password:
receiving incremental file list
./
rsynctest.txt
sent 52 bytes received 195 bytes 7.16 bytes/sec
total size is 883 speedup is 3.57
[root@fedora ~]# ls
aabbcc anaconda-ks.cfg fedora rsynctest.txt
```

successo! Oltre alla scrittura precedente basata sul protocollo rsync, puoi anche scrivere così: `rsync://li@10.1.2.84/share`

4.2 push/carica

```
[root@fedora ~]# touch /root/fedora.txt
[root@fedora ~]# rsync -avz /root/* li@192.168.100.4::share
Password:
sending incremental file list
rsync: [sender] read error: Connection reset by peer (104)
rsync error: error in socket IO (code 10) at io.c(784) [sender = 3.2.3]
```

Vieni informato che l'errore di lettura è relativo al "read only = yes" del server. Cambialo in "no" e riavvia il servizio `[root@Rocky ~]# systemctl restart rsyncd.service`

Prova di nuovo, ti viene negato il permesso:

```
[root@fedora ~]# rsync -avz /root/* li@192.168.100.4::share
Password:
sending incremental file list
fedora.txt
rsync: mkstemp " /.fedora.txt.hxzBIQ " (in share) failed: Permission denied
(13)
sent 206 bytes received 118 bytes 92.57 bytes/sec
total size is 883 speedup is 2.73
```

```
rsync error: some files/attrs were not transferred (see previous errors) (code
23) at main.c(1330) [sender = 3.2.3]
```

Il nostro utente virtuale qui è li, che viene mappato all'utente di sistema nobody per impostazione predefinita. Naturalmente, è possibile cambiarlo in altri utenti del sistema. In altre parole, nobody non ha il permesso di scrittura nella directory /rsync/. Naturalmente, possiamo usare `[root@Rocky ~]# setfacl -mu:nobody:rwx /rsync/` , riprovare, e avere successo.

```
[root@fedora ~]# rsync -avz /root/* li@192.168.100.4::share
Password:
sending incremental file list
fedora.txt
sent 206 bytes received 35 bytes 96.40 bytes/sec
total size is 883 speedup is 3.66
```

5. /etc/rsyncd.conf

Nel precedente articolo rsync demo 02 abbiamo introdotto alcuni parametri di base. Questo articolo è per integrare altri parametri.

Parametri	Descrizione
fake super = yes	sì significa che non è necessario che il demone venga eseguito come root per memorizzare gli attributi completi del file.
uid =	user id
gid =	Due parametri sono utilizzati per specificare l'utente e il gruppo utilizzati per trasferire i file quando si esegue il demone rsync come root. Il valore predefinito è nobody
use chroot = yes	Se la directory radice deve essere bloccata prima della trasmissione, sì sì, no no. Al fine di aumentare la sicurezza, rsync yes di default.
max connections = 4	Il numero massimo di connessioni ammesse, il valore predefinito è 0, il che significa che non ci sono restrizioni
lock file = /var/run/rsyncd.lock	Il file di blocco specificato, che è associato al parametro "max connection"
exclude = lost+found/	Escludi directory che non devono essere trasferite
transfer logging = yes	Indica se abilitare il formato di log simile a ftp per registrare i caricamenti e i download di rsync
timeout = 900	Specifica il periodo di timeout. Se nessun dato viene trasmesso entro il tempo specificato, rsync uscirà direttamente. L'unità è secondi, il valore predefinito è 0 significa mai time out
ignore nonreadable = yes	Indica se ignorare i file a cui l'utente non ha diritti di accesso
motd file = /etc/rsyncd/rsyncd.motd	Usato per specificare il percorso del file del messaggio. Per impostazione predefinita, non c'è un file motd. Questo messaggio è il messaggio di benvenuto visualizzato quando l'utente accede.
hosts allow = 10.1.1.1/24	Usato per specificare a quali client IP o segmento di rete è consentito accedere. È possibile compilare l'ip, il segmento di rete, il nome dell'host, l'host sotto il dominio e separare i multipli con gli spazi. Consenti a tutti di accedere per impostazione predefinita
hosts deny = 10.1.1.20	A quali client Ip o segmento di rete specificati dall'utente non è consentito accedere. Se gli host consentono e gli host negano lo stesso risultato corrispondente, il client non può accedervi alla fine. Se l'indirizzo del client non è né nel host allow né nel host deny, il client è autorizzato ad accedere. Per impostazione predefinita, questo parametro non esiste
auth users = li	Abilita utenti virtuali, più utenti sono separati da virgole nello stato inglese
syslog facility = daemon	Definisce il livello del registro di sistema. Questo valore può essere riempito con: auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0, local1, local2 local3, local4, local5, local6 and local7. Il valore predefinito è daemon

5.1 Configurazione raccomandata

/etc/rsyncd.conf

```
uid = nobody
gid = nobody
address = 192.168.100.4
use chroot = yes
max connections = 10
syslog facility = daemon
pid file = /var/run/rsyncd.pid
log file = /var/log/rsyncd.log
lock file = /var/run/rsyncd.lock
[file]
  comment = rsync
  path = /rsync/
  read only = no
  dont compress = *.gz *.bz2 *.zip
  auth users = li
  secrets file = /etc/rsyncd users.db
```

6. Prefazione

Da rsync Breve Descrizione sappiamo che rsync è uno strumento di sincronizzazione incrementale. Ogni volta che viene eseguito il comando `rsync`, i dati possono essere sincronizzati una volta, ma i dati non possono essere sincronizzati in tempo reale. Come farlo?

Con inotify-tools, questo strumento può realizzare la sincronizzazione unidirezionale in tempo reale. Poiché si tratta di sincronizzazione dati in tempo reale, il prerequisito è quello di accedere senza autenticazione con password.

Indipendentemente dal fatto che si tratti di protocollo rsync o protocollo SSH, entrambi possono ottenere l'accesso all'autenticazione senza password.

6.1 Accesso all'autenticazione senza password del protocollo SSH

Innanzitutto, genera una chiave pubblica e una coppia di chiavi private sul client e continua a premere Invio dopo aver digitato il comando. La coppia di chiavi viene salvata nella directory /root/.ssh/.

```
[root@fedora ~]# ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256: TDA3tWeRhQIqzTORLaqy18nKnQOFNDhoAsNqRLo1TMg root@fedora
The key's randomart image is:
+---[RSA 2048]----+
|O+. +o+o. .+. |
|BEo oo*....o. |
|*o+o..*.. ..o |
|.+..o. = o |
|o o S |
|. o |
| o +. |
|....=. |
```

```
| .o.o. |
+----[SHA256]-----+
```

Quindi, utilizza il comando `scp` per caricare il file della chiave pubblica sul server. Per esempio, carica questa chiave pubblica dell'utente **testrsync**

```
[root@fedora ~]# scp -P 22 /root/.ssh/id_rsa.pub root@192.168.100.4:/home/
testrsync/
```

```
[root@Rocky ~]# cat /home/testrsync/id_rsa.pub >> /home/testrsync/.ssh/
authorized_keys
```

Prova ad accedere senza autenticazione segreta, successo!

```
[root@fedora ~]# ssh -p 22 testrsync@192.168.100.4
Last login: Tue Nov 2 21:42:44 2021 from 192.168.100.5
[testrsync@Rocky ~]$
```

Suggerimento!

Il file di configurazione del server **/etc/ssh/sshd_config** dovrebbe essere aperto PubkeyAuthentication yes

6.2 rsync protocollo di autenticazione senza password

Dal lato client, il servizio rsync prepara una variabile di ambiente per il sistema-**RSYNC_PASSWORD**, che è vuoto per impostazione predefinita, come mostrato sotto:

```
[root@fedora ~]# echo "$RSYNC_PASSWORD"

[root@fedora ~]#
```

Se si desidera ottenere l'accesso all'autenticazione senza password, è sufficiente assegnare un valore a questa variabile. Il valore assegnato è la password precedentemente impostata per l'utente virtuale li. Allo stesso tempo, dichiara questa variabile come una variabile globale.

```
[root@Rocky ~]# cat /etc/rsyncd_users.db
li:13579
```

```
[root@fedora ~]# export RSYNC_PASSWORD=13579
```

Provalo, successo! Nessun nuovo file appare qui, quindi l'elenco dei file trasferiti non è visualizzato.

```
[root@fedora ~]# rsync -avz li@192.168.100.4::share /root/
receiving incremental file list
./

sent 30 bytes received 193 bytes 148.67 bytes/sec
total size is 883 speedup is 3.96
```

Suggerimento!

Puoi scrivere questa variabile in **/etc/profile** per renderla efficace in modo permanente. Il contenuto è: `export RSYNC_PASSWORD=13579`

7. Compilare e installare

Eseguire le seguenti operazioni nel server. Nel vostro ambiente, alcuni pacchetti dipendenti potrebbero essere mancanti. Installali utilizzando: `dnf -y install autoconf automake libtool`

```
[root@Rocky ~]# wget -c https://github.com/inotify-tools/inotify-tools/archive/refs/tags/3.21.9.6.tar.gz
[root@Rocky ~]# tar -zvxf 3.21.9.6.tar.gz -C /usr/local/src/
[root@Rocky ~]# cd /usr/local/src/inotify-tools-3.21.9.6/
[root@Rocky /usr/local/src/inotify-tools-3.21.9.6]# ./autogen.sh && \
./configure --prefix=/usr/local/inotify-tools && \
make && \
make install
...
[root@Rocky ~]# ls /usr/local/inotify-tools/bin/
inotifywait inotifywatch
```

Aggiungi la variabile di ambiente PATH, scrivila al file di configurazione e lascia che abbia effetto in modo permanente.

```
[root@Rocky ~]# vim /etc/profile
...
PATH=$PATH:/usr/local/inotify-tools/bin/
[root@Rocky ~]# . /etc/profile
```

Perché non utilizzare il pacchetto RPM inotify-tools dell'archivio EPEL? E il modo per usare il codice sorgente per compilare e installare?

L'autore ritiene personalmente che la trasmissione di dati a distanza sia una questione di efficienza, soprattutto in un ambiente di produzione, dove ci sono un gran numero di file da sincronizzare e un singolo file è particolarmente grande. Inoltre, la nuova versione avrà alcune correzioni di bug e espansioni delle funzioni, e forse l'efficienza di trasmissione della nuova versione sarà maggiore, quindi vi consiglio di installare inotify-tools dal codice sorgente. Naturalmente, questo è il suggerimento personale dell'autore, non tutti gli utenti lo devono seguire.

7.1 Regolazione dei parametri del kernel

È possibile regolare i parametri del kernel in base alle esigenze dell'ambiente di produzione. Per impostazione predefinita, ci sono tre file in **/proc/sys/fs/inotity/**

```
[root@Rocky ~]# cd /proc/sys/fs/inotify/
[root@Rocky /proc/sys/fs/inotify]# cat max_queued_events ;cat
max_user_instances ;cat max_user_watches
16384
128
28014
```

- max_queued_events-dimensione massima della coda del monitor, predefinita 16384
- max_user_instances-il numero massimo di istanze di monitoraggio, il valore predefinito è 128
- max_user_watches-il numero massimo di file monitorati per istanza, il valore predefinito è 8192

Scrivi alcuni parametri e valori in **/etc/sysctl.conf**, gli esempi sono i seguenti. Quindi usa `sysctl -p` per rendere i file effettivi

```
fs.inotify.max_queued_events = 16384
fs.inotify.max_user_instances = 1024
fs.inotify.max_user_watches = 1048576
```

7.2 Comandi correlati

Lo strumento inotify-tools ha due comandi, chiamati:

- **inotifywait**: per il monitoraggio continuo, risultati in tempo reale. È generalmente usato con lo strumento di backup incrementale rsync. Poiché si tratta di un monitoraggio del file system, può essere utilizzato con uno script. Introdurremo lo script specifico in un secondo momento.
- **inotifywatch**: per il monitoraggio a breve termine, risultati in uscita dopo il completamento dell'attività.

`inotifywait` ha principalmente le seguenti opzioni:

```
-m significa monitoraggio continuo
-r Monitoraggio ricorsivo
-q Semplificare le informazioni di output
-e specifica il tipo di evento di dati di monitoraggio, più tipi di eventi sono
separati da virgole nello stato inglese
```

I tipi di eventi sono i seguenti:

Tipo di evento	Descrizione
access	Accesso al contenuto di un file o di una directory
modify	I contenuti del file o della directory vengono scritti
attrib	Gli attributi del file o della directory vengono modificati
close_write	Il file o la directory viene aperto in modalità scrivibile e quindi chiuso
close_nowrite	Il file o la directory è chiuso dopo essere stato aperto in modalità di sola lettura
close	Indipendentemente dalla modalità lettura/scrittura, il file o la directory viene chiuso
open	Il file o la directory è aperto
moved_to	Un file o una directory è spostato nella directory monitorata
moved_from	Un file o una directory viene spostato dalla directory monitorata
move	Ci sono file o cartelle che vengono spostati o rimossi dalla directory di monitoraggio
move_self	Il file o la directory monitorati sono stati spostati
create	Ci sono file o directory create nella directory monitorata
delete	Viene eliminato un file o una directory nella directory monitorata
delete_self	Il file o la directory sono stati eliminati
unmount	File system contenente file o directory non montati

Esempio: `[root@Rocky ~]# inotifywait -mrq -e create,delete /rsync/`

7.3 Dimostrazione del comando `inotifywait`

Digitare il comando nel primo terminale pts/0, e la finestra viene bloccata dopo aver premuto Invio, indicando che sta monitorando

```
[root@Rocky ~]# inotifywait -mrq -e create,delete /rsync/
```

Nel secondo terminale pts/1, vai nella directory /rsync/ e crea un file.

```
[root@Rocky ~]# cd /rsync/
[root@Rocky /rsync]# touch inotify
```

Torna al primo terminale pts/0, le informazioni di output sono le seguenti:

```
[root@Rocky ~]# inotifywait -mrq -e create,delete /rsync/
/rsync/ CREATE inotify
```

7.4 Combinazione di `inotifywait` and `rsync`

Suggerimento!

Stiamo operando nel server Rocky Linux 8, utilizzando il protocollo SSH per la dimostrazione.

Per il login di autenticazione senza password del protocollo SSH, si prega di fare riferimento a rsync accesso di autenticazione senza password, che non è descritto qui. Un esempio del contenuto di uno script bash è il seguente. È possibile aggiungere diverse opzioni dopo il comando in base alle esigenze per soddisfare le vostre esigenze. Ad esempio, puoi anche aggiungere `--delete` dopo il comando `rsync`.

```
#!/bin/bash
a="/usr/local/inotify-tools/bin/inotifywait -mrq -e modify,move,create,delete /
rsync/"
b="/usr/bin/rsync -avz /rsync/* testfedora@192.168.100.5:/home/testfedora/"
$a | while read directory event file
    do
        $b &>> /tmp/rsync.log
    done
```

```
[root@Rocky ~]# chmod +x rsync_inotify.sh
[root@Rocky ~]# bash /root/rsync_inotify.sh &
```

Suggerimento!

Quando si utilizza il protocollo SSH per la trasmissione della sincronizzazione dei dati, se la porta di servizio SSH della macchina di destinazione non è la 22, puoi usare un metodo simile a questo—— `b="/usr/bin/rsync -avz -e 'ssh -p [port-number]' /rsync/* testfedora@192. 68.100.5:/home/testfedora/"`

Suggerimento!

Se vuoi avviare questa script all'avvio `[root@Rocky ~]# echo "bash /root/rsync_inotify. h &" >> /etc/rc.local [root@Rocky ~]# chmod +x /etc/rc.local`

Se si sta utilizzando il protocollo rsync per la sincronizzazione, è necessario configurare il servizio rsync della macchina di destinazione, fare riferimento a rsync demo 02, rsync file di configurazione, accesso senza autenticazione segreta rsync

8. Breve

Come abbiamo accennato in precedenza, la sincronizzazione unidirezionale utilizza rsync + inotify-tools. In alcuni scenari di utilizzo speciali, può essere richiesta la sincronizzazione a due vie, che richiede inotify-tools + unison.

8.1 Preparazione dell'ambiente

- Sia Rocky Linux 8 che Fedora 34 richiedono la compilazione del codice sorgente e l'installazione **inotify-tools**, che non è specificamente trattato qui.
- Entrambe le macchine devono essere autenticate senza password, qui usiamo il protocollo SSH
- ocaml utilizza v4.12.0, unison utilizza v2.51.4.

Dopo che l'ambiente è pronto, si può verificare:

```
[root@Rocky ~]# inotifywa
inotifywait inotifywatch
[root@Rocky ~]# ssh -p 22 testrsync@192.168.100.5
Last login: Thu Nov 4 13:13:42 2021 from 192.168.100.4
[testrsync@fedora ~]$
```

```
[root@fedora ~]# inotifywa
inotifywait inotifywatch
[root@fedora ~]# ssh -p 22 testrsync@192.168.100.4
Last login: Wed Nov 3 22:07:18 2021 from 192.168.100.5
[testrsync@Rocky ~]$
```

Suggerimento

I file di configurazione delle due macchine **/etc/ssh/sshd_config** dovrebbero essere aperti PubkeyAuthentication yes

8.2 Rocky Linux 8 installare unison

Ocaml è un linguaggio di programmazione, e lo strato inferiore di unison dipende da esso.

```
[root@Rocky ~]# wget -c https://github.com/ocaml/ocaml/archive/refs/tags/
4.12.0.tar.gz
[root@Rocky ~]# tar -zvxf 4.12.0.tar.gz -C /usr/local/src/
[root@Rocky ~]# cd /usr/local/src/ocaml-4.12.0
[root@Rocky /usr/local/src/
ocaml-4.12.0]# ./configure --prefix=/usr/local/ocaml && make world opt && make
install
...
[root@Rocky ~]# ls /usr/local/ocaml/
bin lib man
[root@Rocky ~]# echo PATH=$PATH:/usr/local/ocaml/bin >> /etc/profile
[root@Rocky ~]# . /etc/profile
```

```
[root@Rocky ~]# wget -c https://github.com/bcpierce00/unison/archive/refs/tags/
v2.51.4.tar.gz
[root@Rocky ~]# tar -zvxf v2.51.4.tar.gz -C /usr/local/src/
[root@Rocky ~]# cd /usr/local/src/unison-2.51.4/
[root@Rocky /usr/local/src/unison-2.51.4]# make UISTYLE=txt
...
[root@Rocky /usr/local/src/unison-2.51.4]# ls src/unison
src/unison
[root@Rocky /usr/local/src/unison-2.51.4] cp -p src/unison /usr/local/bin
```

8.3 Fedora 34 installare unison

La stessa operazione.

```
[root@fedora ~]# wget -c https://github.com/ocaml/ocaml/archive/refs/tags/
4.12.0.tar.gz
[root@feodora ~]# tar -zvxf 4.12.0.tar.gz -C /usr/local/src/
[root@fedora ~]# cd /usr/local/src/ocaml-4.12.0
[root@fedora /usr/local/src/ocaml-4.12.0]# ./configure --prefix=/usr/local/
ocaml && make world opt && make install
...
[root@fedora ~]# ls /usr/local/ocaml/
bin lib man
[root@fedora ~]# echo PATH=$PATH:/usr/local/ocaml/bin >> /etc/profile
[root@fedora ~]#. /etc/profile
```

```
[root@fedora ~]# wget -c https://github.com/bcpierce00/unison/archive/refs/
tags/v2.51.4.tar.gz
[root@fedora ~]# tar -zvxf v2.51.4.tar.gz -C /usr/local/src/
[root@fedora ~]# cd /usr/local/src/unison-2.51.4/
[root@fedora /usr/local/src/unison-2.51.4]# make UISTYLE=txt
```

```
...
[root@fedora /usr/local/src/unison-2.51.4]# ls src/unison
src/unison
[root@fedora /usr/local/src/unison-2.51.4]# cp -p src/unison /usr/local/bin
```

8.4 Dimostrazione

Il nostro requisito è la directory /dir1/ di Rocky Linux 8 che viene automaticamente sincronizzata nella directory /dir2/ di Fedora 34; allo stesso tempo, la directory /dir2/ di Fedora 34 viene automaticamente sincronizzata con la directory /dir1/ di Rocky Linux 8

8.4.1 Configure Rocky Linux 8

```
[root@Rocky ~]# mkdir /dir1
[root@Rocky ~]# setfacl -m u:testrsync:rwx /dir1/
[root@Rocky ~]# vim /root/unison1.sh
#!/bin/bash
a="/usr/local/inotify-tools/bin/inotifywait -mrq -e create,delete,modify,move /
dir1/"
b="/usr/local/bin/unison -batch /dir1/ ssh://testrsync@192.168.100.5//dir2"
$a | while read directory event file
do
    $b &>> /tmp/unison1.log
done
[root@Rocky ~]# chmod +x /root/unison1.sh
[root@Rocky ~]# bash /root/unison1.sh &
[root@Rocky ~]# jobs -l
```

8.4.2 Configurare Fedora 34

```
[root@fedora ~]# mkdir /dir2
[root@fedora ~]# setfacl -m u:testrsync:rwx /dir2/
[root@fedora ~]# vim /root/unison2.sh
#!/bin/bash
a="/usr/local/inotify-tools/bin/inotifywait -mrq -e create,delete,modify,move /
dir2/"
b="/usr/local/bin/unison -batch /dir2/ ssh://testrsync@192.168.100.4//dir1"
$a | while read directory event file
do
    $b &>> /tmp/unison2.log
done
```

```
[root@fedora ~]# chmod +x /root/unison2.sh
[root@fedora ~]# bash /root/unison2.sh &
[root@fedora ~]# jobs -l
```

Suggerimento!

Per la sincronizzazione bidirezionale, gli script di entrambe le macchine devono essere avviati, altrimenti verrà segnalato un errore.

Suggerimento!

Se vuoi avviare questo script all'avvio `[root@Rocky ~]# echo "bash /root/unison1. h &" >> /etc/rc.local`
`[root@Rocky ~]# chmod +x /etc/rc.local`

Suggerimento!

Se vuoi interrompere il processo corrispondente di questo script, puoi trovarlo con il comando `htop` e quindi **kill**