

# Learning Rsync On Rocky Linux (Ukrainian version)

---

A book from the Documentation Team

Version : 2026/06/11

---

*Rocky Documentation Team*

*Copyright © 2023 The Rocky Enterprise Software Foundation*

## Table of contents

---

1. Licence	3
2. Коротко про резервні копії	4
2.1 Коротко про rsync	6
2.1.1 Основні принципи та особливості	9
3. Передмова	11
3.1 Опис середовища	12
3.2 Демонстрація на основі протоколу SSH	12
3.2.1 витягти/завантажити	12
3.2.2 передати/відвантажити	13
4. Демонстрація на основі протоколу rsync	15
4.1 витягти/завантажити	17
4.2 передати/відвантажити	18
5. /etc/rsyncd.conf	19
5.1 Рекомендована конфігурація	20
6. Передмова	21
6.1 Вхід для аутентифікації протоколу SSH без пароля	21
6.2 протокол rsync автентифікація без пароля	22
7. Компіляція та інсталяція	24
7.1 Налаштування параметрів ядра	25
7.2 Пов'язані команди	25
7.3 Демонстрація команди inotifywait	26
7.4 Поєднання inotifywait і rsync	27
8. Коротко	29
8.1 Підготовка середовища	29
8.2 Встановлення Rocky Linux 8 unison	29
8.3 Встановлення Fedora 34 unison	30
8.4 Demo	31
8.4.1 Налаштування Rocky Linux 8	31
8.4.2 Налаштування Fedora 34	31

## 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](https://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. Коротко про резервні копії

---

Що таке резервна копія?

Резервне копіювання належить до дублювання даних у файловій системі або базі даних. У разі помилки або катастрофи ефективні дані системи можуть бути відновлені своєчасно, і робота системи може бути відновлена у нормальний режим.

Які є методи резервного копіювання?

- Повне резервне копіювання: це одноразова копія всіх файлів, директорій або даних на жорсткому диску або в базі даних. (Плюси: найкращий, може швидше відновлювати дані. Недоліки: займають більше місця на жорсткому диску.)
- Інкрементне резервне копіювання: стосується резервного копіювання даних, оновлених після останнього повного або інкрементного резервного копіювання. Процес виглядає так, наприклад, повне резервне копіювання в перший день; резервне копіювання щойно доданих даних на другий день, на відміну від повного резервного копіювання; на третій день резервне копіювання щойно доданих даних на основі другого дня, відносно наступного дня і так далі.
- Диференціальне резервне копіювання: стосується резервного копіювання змінених файлів після повного резервного копіювання. Наприклад, повне резервне копіювання в перший день; резервне копіювання нових даних на другий день; резервне копіювання нових даних з другого дня до третього дня на третій день; і резервне копіювання всіх нових даних з другого дня до четвертого дня на четвертий день і так далі.
- Вибіркове резервне копіювання: належить до резервного копіювання частини системи.
- Холодне резервне копіювання: стосується резервного копіювання, коли система вимкнена або в стані обслуговування. Резервні копії даних точно збігаються з даними в системі протягом цього періоду. Резервні копії даних точно збігаються з даними в системі протягом цього періоду.
- Гаряче резервне копіювання: належить до резервного копіювання, коли система працює в нормальному режимі. Оскільки дані в системі оновлюються в будь-який час, резервні копії даних мають певну затримку відносно реальних даних системи.
- Віддалене резервне копіювання: стосується резервного копіювання даних в іншому географічному місці, щоб уникнути втрати даних і переривання роботи служби через пожежу, стихійне лихо, крадіжку тощо.

## 2.1 Коротко про rsync

---

На сервері я створив резервну копію першого розділу на другий розділ, який широко відомий як «локальне резервне копіювання». Конкретні засоби резервного копіювання: `tar`, `dd`, `dump`, `cp` тощо. Незважаючи на те, що резервна копія даних створюється на цьому сервері, якщо апаратне забезпечення не завантажиться належним чином, дані не будуть відновлені. Щоб розв'язувати цю проблему з локальним резервним копіюванням, ми запровадили інший вид резервного копіювання --- "віддалене резервне копіювання".

Деякі люди скажуть, чи не можу я просто використати команду `tar` або `cp` на першому сервері та надіслати її на другий сервер через `scp` чи `sftp`?

У виробничому середовищі обсяг даних відносно великий. По-перше, `tar` або `cp` забирає багато часу та споживає продуктивність системи. Передача через `scp` або `sftp` також займає значну пропускну здатність мережі, що не допускається у фактичному виробничому середовищі. По-друге, ці команди або інструменти мають бути введені адміністратором вручну та поєднані з `crontab` запланованого завдання. Однак час, встановлений `crontab`, непросто зрозуміти, і резервне копіювання даних не підходить, якщо час занадто короткий або занадто довгий.

Тому у виробничому середовищі має бути резервне копіювання даних, яке має відповідати таким вимогам:

1. Резервні копії, що передаються мережею
2. Синхронізація файлів даних у реальному часі
3. Менше завантаження системних ресурсів і більш висока ефективність


Схоже, що `rsync` відповідає зазначеним вище потребам. Він використовує ліцензійну угоду GNU з відкритим кодом. Це швидкий інструмент інкрементного резервного копіювання. Остання версія 3.2.3 (2020-08-06). Ви можете відвідати [офіційний веб-сайт](#) для отримання додаткової інформації.

Що стосується підтримки платформи, то підтримується більшість Unix подібних систем, будь то GNU/Linux чи BSD. Крім того, існує відповідний `rsync` під платформою Windows, такий як `swRsync`.

Оригінальний `rsync` підтримувався австралійським програмістом Ендрю Тріджел (показано на малюнку 1 нижче), а тепер його підтримує Вейн Девісон (показано на малюнку 2 нижче). Для обслуговування ви можете перейти на [адресу проекту github](#), щоб отримати потрібну інформацію.





 Примітка

сам по собі `rsync` є лише інкрементним інструментом резервного копіювання та не має функції синхронізації даних у реальному часі (його потрібно доповнити іншими програмами). Крім того, синхронізація одностороння. Якщо ви хочете реалізувати двосторонню синхронізацію, вам потрібно співпрацювати з іншими інструментами.

## 2.1.1 Основні принципи та особливості

Як `rsync` забезпечує ефективне резервне копіювання односторонньої синхронізації даних?

Основою `rsync` є його **алгоритм контрольної суми**. Якщо вам цікаво, ви можете перейти до розділу [Як працює Rsync](#) і [Алгоритм rsync](#). Цей розділ виходить за межі компетенції автора і не буде детально розглядатися. Цей розділ виходить за межі компетенції автора і не буде детально розглядатися.

Чим характеризується `rsync` :

- Весь каталог можна оновлювати рекурсивно;
- Може вибірково зберігати атрибути синхронізації файлів, такі як жорстке посилання, м'яке посилання, власник, група, відповідні дозволи, час модифікації тощо, і може зберігати деякі атрибути;
- Підтримка двох протоколів для передачі: один протокол `ssh`, інший протокол `rsync`

## 3. Передмова

---

`rsync` має виконати автентифікацію користувача перед синхронізацією даних. **Існує два протоколи для автентифікації: протокол SSH і протокол rsync (порт протоколу rsync за замовчуванням — 873)**

- Метод перевірки протоколу SSH: використовуйте протокол SSH як основу для автентифікації користувача (тобто використовуйте для перевірки системного користувача та пароль самого GNU/Linux), а потім виконайте синхронізацію даних.
- Метод входу перевірки протоколу rsync: використовуйте протокол rsync для автентифікації користувача (користувачі системи не GNU/Linux, подібні до віртуальних користувачів vsftpd), а потім виконайте синхронізацію даних.

Перед конкретною демонстрацією синхронізації rsync вам потрібно використати команду `rsync`. У Rocky Linux 8 пакет rsync rpm встановлено за замовчуванням, а його версія — 3.1.3-12, як показано нижче:

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

Основний формат: `rsync [параметри] оригінальне розташування цільове розташування`

Часто використовувані варіанти:

- a: режим архіву, рекурсивний і зберігає атрибути об'єкта файлу, що еквівалентно `-rlptgoD` (без `-H`, `-A`, `-X`)
- v: відображає детальну інформацію про процес синхронізації
- z: стискає під час передачі файлів
- H: Зберігає файли жорстких посилань
- A: зберігає дозволи ACL
- X: зберігає дозволи `chattr`
- r: рекурсивний режим, включаючи всі файли в каталозі та підкаталогах
- l: усе ще зарезервовано для файлів символічних посилань
- p: дозвіл на збереження атрибутів файлів
- t: час для збереження атрибутів файлу
- g: зберігає групу, що належить атрибуту файлу (лише для суперкористувачів)
- o: зберігає власника атрибутів файлу (лише для суперкористувачів)
- D: зберігає файли пристрою та інші спеціальні файли

Особисте використання автором: `rsync -avz` оригінальне розташування цільове розташування

### 3.1 Опис середовища

Елемент	Опис
Rocky Linux 8(Server)	192.168.100.4/24
Fedora 34(client)	192.168.100.4/24

Ви можете використовувати Fedora 34 для завантаження та завантаження

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

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

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

### 3.2 Демонстрація на основі протоколу SSH

#### Підказка

Тут і Rocky Linux 8, і Fedora 34 використовують користувача `root` для входу. Fedora 34 — це клієнт, а Rocky Linux 8 — сервер.

#### 3.2.1 витягти/завантажити

Оскільки він заснований на протоколі SSH, ми спочатку створюємо користувача на сервері:

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

На стороні клієнта ми витягуємо/завантажуємо його, а файл на сервері має назву `/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
```

Передача пройшла успішно.

#### Підказка

Якщо порт SSH сервера не є стандартним 22, ви можете вказати порт подібним чином `---rsync -avz -e 'ssh -p [порт]'`.

## 3.2.2 передати/відвантажити

```
[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]
```

### Відмовлено в дозволі підказки, як із цим впоратися?

Спочатку перевірте права доступу до каталогу `/rsync/`. Очевидно, що дозволу "w" немає. Ми можемо використувати `setfacl`, щоб надати дозвіл:

```
[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
```

Спробуйте ще раз, успіх!

```
[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. Демонстрація на основі протоколу rsync

---

У vsftpd є віртуальні користувачі (імітовані користувачі, налаштовані адміністратором), тому що використовувати анонімних і локальних користувачів небезпечно. Ми знаємо, що сервер, заснований на протоколі SSH, повинен забезпечувати наявність системи користувачів. Коли існує багато вимог до синхронізації, може знадобитися створити багато користувачів. Це, очевидно, не відповідає стандартам роботи та обслуговування GNU/Linux (чим більше користувачів, тим небезпечніше), у rsync з міркувань безпеки існує метод автентифікації протоколу rsync для входу в систему.

### Як це зробити?

Просто запишіть відповідні параметри та значення у файл конфігурації. У Rocky Linux 8 вам потрібно вручну створити файл `/etc/rsyncd.conf`.

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

Деякі параметри та значення цього файлу наведені нижче, ось [тут](#) є інші описи параметрів:

Елемент	Опис
address = 192.168.100.4	IP-адреса, яку rsync прослуховує за замовчуванням
port = 873	порт прослуховування за замовчуванням rsync
pid file = /var/run/ rsyncd.pid	Розташування файлу процесу pid
log file = /var/log/ rsyncd.log	Розташування файлу журналу
[share]	Назва спільного доступу
comment = rsync	Примітки або опис
path = /rsync/	Розташування системного шляху, де він розташований
read only = yes	yes означає лише читання, no означає читання та запис
dont compress = *.gz *.gz2 *.zip	Які типи файлів не стискають його
auth users = li	Вмикає віртуальних користувачів і визначає назву віртуального користувача. Його потрібно створити самостійно
secrets file = /etc/ rsyncd_users.db	Використовується для вказівки розташування файлу пароля віртуального користувача, який має закінчуватися на .db. Формат вмісту файлу: «Ім'я користувача: пароль», по одному на рядок

#### Підказка

Дозвіл доступу до файлу паролів має бути 600.

Запишіть деякий вміст файлу в /etc/rsyncd.conf, а ім'я користувача та пароль – у /etc/rsyncd\_users.db, дозвіл 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
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
```

Можливо, вам знадобиться `dnf -y встановити rsync-daemon`, перш ніж ви зможете запуснути службу: `systemctl start rsyncd.service`

```
[root@Rocky ~]# systemctl start rsyncd.service
[root@Rocky ~]# netstat -tulnp
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.100.4:873      0.0.0.0:*               LISTEN
tcp6       0      0 :::22                  :::*                     LISTEN
udp        0      0 127.0.0.1:323          0.0.0.0:*               LISTEN
udp6       0      0 :::1:323               :::*                     LISTEN
```

## 4.1 витягти/завантажити

---

Створіть файл на сервері для перевірки:

```
[root@Rocky]# touch /rsync/rsyncctest.txt
```

Клієнт робить наступне:

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

успіх! На додаток до написання вище на основі протоколу rsync, ви також можете написати так: `rsync://li@10.1.2.84/share`

## 4.2 передати/відвантажити

```
[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]
```

Вам буде запропоновано, що помилка читання пов'язана з параметром «read only = yes» сервера. Змініть його на "no" та перезапустіть службу [root@Rocky

```
~]# systemctl перезапустіть rsyncd.service
```

Спробуйте ще раз, запитуючи дозвіл заборонено:

```
[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]
```

Наш віртуальний користувач тут li, який за замовчуванням зіставлено з системним користувачем nobody. Звичайно, ви можете змінити його на інших користувачів системи. In other words, nobody does not have write permission to the /rsync/ directory. Звичайно, ми можемо використати

```
[root@Rocky ~]# setfacl -mu:nobody:rwx /rsync/ , спробувати ще раз і вдасться.
```

```
[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

У попередній статті [rsync demo 02](#) ми представили деякі основні параметри. Ця стаття доповнює інші параметри.

Параметри	Опис
fake super = yes	yes означає, що вам не потрібно, щоб демон запускався від імені користувача root для збереження всіх атрибутів файлу.
uid =	ідентифікатор користувача
gid =	Два параметри використовуються для визначення користувача та групи, які використовуються для передачі файлів під час запуску демона rsync від імені користувача root. За замовчуванням ніхто
use chroot = yes	Чи потрібно заблокувати кореневий каталог перед передачею, так, так, ні, ні. З метою підвищення безпеки rsync за замовчуванням має значення «так».
max connections = 4	Максимально дозволена кількість підключень, значення за замовчуванням 0, що означає відсутність обмежень
lock file = /var/run/rsyncd.lock	Зазначений файл блокування, пов'язаний із параметром «max connections»
exclude = lost+found/	Виключити каталоги, які не потрібно переносити
transfer logging = yes	Чи ввімкнути ftp-подібний формат журналу для запису завантажень і завантажень rsync
timeout = 900	Визначте період очікування. Якщо дані не передано протягом зазначеного часу, rsync завершить роботу безпосередньо. Одиницею вимірювання є секунди, значення за замовчуванням 0 означає, що тайм-аут ніколи не буде
ignore nonreadable = yes	Чи ігнорувати файли, до яких користувач не має прав доступу
motd file = /etc/rsyncd/motd	Використовується для вказівки шляху до файлу повідомлення. По замовчуванню, тут відсутній motd файл. Це вітальне повідомлення, яке відображається під час входу користувача.
hosts allow = 10.1.1.1/24	Використовується для вказівки, до якої IP-адреси або сегмента мережі клієнтам дозволено доступ. Ви можете заповнити IP-адресу, сегмент мережі, ім'я хоста, хост у домені та розділити їх пробілами. Дозволити всім доступ за замовчуванням
hosts deny = 10.1.1.20	Доступ до якого IP-адреси чи сегмента мережі, указаним користувачем, заборонено. Якщо хости дозволять і хости заборонять мають однаковий відповідний результат, клієнт зрештою не зможе отримати доступ. Якщо адресу клієнта немає ні в дозволі, ні в забороні хостів, клієнту дозволено доступ. За замовчуванням такого параметра немає
auth users = li	Увімкнути віртуальних користувачів, кілька користувачів відокремлюються комами в англійському стані
syslog facility = daemon	Визначте рівень системного журналу. Ці значення можна заповнити: auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0, local1, local2, local3, local4, local5, local6 і local7. Типовим по замовчуванню значенням є daemon

## 5.1 Рекомендована конфігурація

---

### **/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. Передмова

---

З короткого [опису rsync](#) ми знаємо, що rsync — це інструмент інкрементної синхронізації. Кожного разу, коли виконується команда `rsync`, дані можуть бути синхронізовані один раз, але дані не можуть бути синхронізовані в реальному часі. Як це зробити?

За допомогою `inotify-tools` цей програмний інструмент може реалізувати односторонню синхронізацію в реальному часі. Оскільки це синхронізація даних у реальному часі, необхідною умовою є вхід без автентифікації за паролем.

**Незалежно від того, чи це протокол rsync чи протокол SSH, обидва можуть досягти входу без пароля.**

### 6.1 Вхід для автентифікації протоколу SSH без пароля

---

Спершу згенеруйте пару відкритого та закритого ключів на клієнті та продовжуйте натискати Enter після введення команди. Пара ключів зберігається в каталозі `/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]-----+
```

Потім скористайтеся командою `scp`, щоб завантажити файл відкритого ключа на сервер. Наприклад, я завантажую цей відкритий ключ користувачу **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
```

Спробуйте увійти без секретної автентифікації, успіх!

```
[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 ~]$
```

#### Підказка

Файл конфігурації сервера `/etc/ssh/sshd_config` має бути відкритий `PubkeyAuthentication yes`

## 6.2 протокол rsync автентифікація без пароля

---

На стороні клієнта служба rsync готує змінну середовища для системного **RSYNC\_PASSWORD**, яка за замовчуванням порожня, як показано нижче:

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

[root@fedora ~]#
```

Якщо ви бажаєте отримати вхід із автентифікацією без пароля, вам потрібно лише призначити значення цій змінній. Призначене значення — це пароль, попередньо встановлений для віртуального користувача `li`. Одночасно оголосите цю змінну як глобальну.

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

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

Спробуйте, успіх! Тут не з'являються нові файли, тому список переданих файлів не відображається.

```
[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
```

#### Підказка

Ви можете записати цю змінну в **/etc/profile**, щоб вона стала чинною. Вміст: `export RSYNC_PASSWORD=13579`

## 7. Компіляція та інсталяція

---

Виконайте наступні операції на сервері. У вашому середовищі можуть бути відсутні деякі залежні пакунки. Встановіть їх за допомогою: `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 -zxvf 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
```

Додайте змінну середовища PATH, запишіть її у файл конфігурації та залиште її діяти постійно.

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

### **Чому б не використати пакет RPM inotify-tools зі сховища EPEL? А як використовувати вихідний код для компіляції та встановлення?**

Автор особисто вважає, що віддалена передача даних є питанням ефективності, особливо у виробничому середовищі, де існує велика кількість файлів, які потрібно синхронізувати, і один файл є особливо великим. Крім того, у новій версії буде виправлено деякі помилки та розширено функції, і, можливо, ефективність передачі нової версії буде вищою, тому я рекомендую встановлювати inotify-tools за вихідним кодом. Звичайно, це особиста пропозиція автора, не кожен користувач повинен її дотримуватися.

## 7.1 Налаштування параметрів ядра

---

Ви можете налаштувати параметри ядра відповідно до потреб виробничого середовища. За замовчуванням у **/proc/sys/fs/inotify/** є три файли

```
[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**-maximum розмір черги монітора, за замовчуванням 16384
- **max\_user\_instances**-the - максимальна кількість екземплярів моніторингу, за замовчуванням 128
- **max\_user\_watches**-the максимальна кількість файлів, що відстежуються на екземпляр, за замовчуванням 8192

Запишіть деякі параметри та значення в **/etc/sysctl.conf**, приклади наведені нижче. Потім скористайтеся `sysctl -p`, щоб файли набули чинності

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

## 7.2 Пов'язані команди

---

Інструмент `inotify-tools` має дві команди, а саме:

- **inotifywait**: для безперервного моніторингу, результати виведення в реальному часі. Зазвичай він використовується з інструментом інкрементального резервного копіювання `rsync`. Оскільки це моніторинг файлової системи, його можна використовувати зі сценарієм. Конкретний сценарій ми представимо пізніше.
- **inotifywatch**: виведення результатів після виконання завдання для короткочасного моніторингу.

`inotifywait` переважно має наступні параметри:

```
-m означає постійний моніторинг
-r Рекурсивний моніторинг
-q Спростити вихідну інформацію
-e вказує тип події даних моніторингу, кілька типів подій розділяються комами в статусі англійською мовою
```

Типи подій:

Тип події	Опис
access	Доступ до вмісту файлу або каталогу
modify	Записується вміст файлу або каталогу
attrib	Атрибути файлу або каталогу змінено
close_write	Файл або каталог відкривається в режимі запису, а потім закривається
close_nowrite	Файл або каталог закриваються після відкриття в режимі лише для читання
close	Незалежно від режиму читання/запису, файл або каталог закриваються
open	Файл або каталог відкрито
moved_to	Файл або каталог переміщується до контрольованого каталогу
moved_from	Файл або каталог переміщено з контрольованого каталогу
move	Є файли або каталоги, які переміщуються до або видаляються з каталогу моніторингу
move_self	Контрольований файл або каталог було переміщено
create	У контрольованому каталозі створено файли або каталоги
delete	Файл або каталог у контрольованому каталозі видалено
delete_self	Файл або каталог видалено
unmount	Файлова система, що містить немонтовані файли або каталоги

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

## 7.3 Демонстрація команди `inotifywait`

Введіть команду в перший термінал `pts/0`, і після натискання `Enter` вікно заблокується, вказуючи на те, що здійснюється моніторинг

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

У другому терміналі `pts/1` перейдіть до каталогу `/rsync/` і створіть файл.

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

Повернемося до першого терміналу pts/0, вихідна інформація така:

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

## 7.4 Поєднання inotifywait і rsync

### Підказка

Ми працюємо на сервері Rocky Linux 8, використовуючи для демонстрації протокол SSH.


Для входу без пароля для автентифікації протоколу SSH зверніться до [rsync для входу без пароля](#), який тут не описано. Нижче наведено приклад вмісту сценарію bash. Ви можете додати різні параметри після команди відповідно до ваших потреб. Наприклад, ви також можете додати `--delete` після команди `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 &
```

### Підказка

Під час використання протоколу SSH для передачі даних синхронізації, якщо порт служби SSH цільової машини не 22, ви можете використовувати метод, подібний до цього—— `b="/usr/bin/rsync -avz -e 'ssh -p [port-number]' /rsync/* testfedora@192.168.100.5:/home/testfedora/"`

 Підказка

```
Якщо ви хочете запустити цей сценарій під час завантаження [root@Rocky ~]# echo "bash /root/rsync_inotify.sh &" >> /etc/rc.local [root@Rocky ~]# chmod +x /etc/rc.local
```

Якщо ви використовуєте протокол rsync для синхронізації, вам потрібно налаштувати службу rsync цільової машини, будь ласка, зверніться до [demo rsync 02](#), файл конфігурації rsync, rsync free Secret authentication login

## 8. Коротко

---

Як ми вже згадували раніше, одностороння синхронізація використовує `rsync` + `inotify-tools`. У деяких особливих сценаріях використання може знадобитися двостороння синхронізація, яка потребує `inotify-tools` + `unison`.

### 8.1 Підготовка середовища

---

- І для Rocky Linux 8, і для Fedora 34 потрібна компіляція вихідного коду та інсталяція **inotify-tools**, яка тут спеціально не розкривається.
- Обидві машини повинні мати автентифікацію без пароля, тут ми використовуємо протокол SSH
- `ocaml` використовує v4.12.0, `unison` використовує v2.51.4.

Після того, як середовище готове, його можна перевірити:

```
[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 ~]$
```

#### Підказка

Конфігураційні файли двох машин `/etc/ssh/sshd_config` мають бути відкриті `PubkeyAuthentication yes`

### 8.2 Встановлення Rocky Linux 8 unison

---

`Ocaml` — це мова програмування, і нижній рівень `unison` залежить від неї.

```
[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 Встановлення Fedora 34 unison

---

Така сама операція.

```
[root@fedora ~]# wget -c https://github.com/ocaml/ocaml/archive/refs/tags/
4.12.0.tar.gz
[root@fedora ~]# 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 Demo

---

**Наша вимога: каталог /dir1/ Rocky Linux 8 автоматично синхронізується з каталогом /dir2/ Fedora 34; водночас каталог /dir2/ Fedora 34 автоматично синхронізується з каталогом /dir1/ Rocky Linux 8**

### 8.4.1 Налаштування 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 Налаштування 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
```

#### Підказка

Для двосторонньої синхронізації повинні бути запущені сценарії обох машин, інакше буде повідомлено про помилку.

#### Підказка

Якщо ви хочете запустити цей сценарій під час завантаження [root@Rocky ~]# echo "bash /root/unison1.sh &" >> /etc/rc.local  
[root@Rocky ~]# chmod +x /etc/rc.local

#### Підказка

Якщо ви хочете зупинити відповідний процес цього сценарію, ви можете знайти його в команді `htop`, а потім **kill**

<https://docs.rockylinux.org/>