Rsnapshot — создание резервных копий

Rsnapshot — базирующаяся на rsync утилита на Perl для создания резервных копий. Особенностью является ориентация на создание полных снапшотов файловой системы через заданный интервал времени, для экономии места на диске, не изменившиеся с момента первого бэкапа файлы ссылаются на оригинальный файл с помощью жестких ссылок.

```
$ sudo yum install rsnapshot
```

```
$ sudo ssh-keygen -t rsa -b 4096 -C "Rsnapshot Backup Server Key"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
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:AFba6Jf3PFqN/RW5dWihhOCIRCRT6ewMUZjI+BbftQ8  Rsnapshot Backup Server Key
The key's randomart image is:
+---[RSA 4096]----+
|o.oBOo. .        |
|o.==.* o.. .     |
| . *+.+.... . .  |
| +.+ .oE . . o.  |
| . +. o So . ooo |
| o. . o.+ . =    |
| * o ..          |
| o . . .         |
| . .             |
+----[SHA256]-----+
```

```
$ sudo ssh-copy-id -i /root/.ssh/id_rsa.pub suser@server.com
```

После окончания настройки необходимо проверить полученный

конфиг-файл на отсуствие синтаксических ошибок, для чего используется команда:

```
$ sudo rsnapshot configtest
```

Если синтаксически ошибок не обнаружено, можно запустить Rsnapshot в тестовом режиме, при котором на экран будут выводится исполняемые команды, но без их выполнения:

```
$ sudo rsnapshot -t alpha
```

После этого можно запустить первый часовой бекап:

```
$ sudo rsnapshot alpha
```

Первый бекап займет наибольшее место и будет выполняться дольше всего.
Посмотреть занятое место можно следующей командой:

```
$ rsnapshot du
```

Когда бекапов образуется несколько, их можно сравнивать командой:

```
$ sudo rsnapshot diff hourly.0 hourly.1
```

После проверки в ручном режиме – автоматизируем через крон:

```
$ sudo vim /etc/cron.d/rsnapshot
```

```
00 */4  * * * rsnapshot hourly     # ежечасный бекап
45 13   * * * rsnapshot daily      # ежедневный бекап
15 14   * * 1 rsnapshot weekly     # еженедельный бекап
30 14   1 * * rsnapshot monthly    # ежемесячный бекап
```

Прим. Если какие-то интервалы не определены в кроне, для избежания ошибок при работе Rsnapshot нужно нужно удалить из него оответсующие строки.

Например, если не задан **interval hourly**, то нужно удалить строку с **/usr/bin/rsnapshot hourly**.

Чтобы не бекапировать различный незначащий файловый мусор,

нужно внести в конфиг его исключения, например:

```
## Задание файлов исключений
exclude         lost+found
exclude         .Trash/
exclude /home/*/.mozilla/firefox/*/Cache/
exclude /home/*/.mozilla/firefox/*/SDThumbs/
exclude         *.tmp
exclude         ~*
exclude         .git/
exclude /home/*/.thumbnails
exclude /home/*/.cache
exclude /home/*/.gvfs
```

https://te-st.ru/entries/rsnapshot-and-borgbckup/

https://www.cyberciti.biz/faq/redhat-cetos-linux-remote-backup-snapshot-server/

https://habr.com/post/45912/

https://poplinux.ru/node/1087

Настройка резервного копирования с помощью Rsnapshot