

restic_backup

restic_backup это скрипт для бэкапа файлов и баз данных с помощью утилиты **restic**.

Ссылки:

- [Сайт проекта restic](#)
- [Документация restic](#) (англ.)

Описание работы скрипта

Скрипт выполняет инкрементальные резервные копии в репозиторий для бэкапов. В качестве репозитория может использоваться S3-совместимое хранилище или другой диск. Все варианты описаны [здесь](#).

Инкрементальный бэкап означает, что первый бэкап содержит полную копию всех данных, а все последующие копии содержат только изменённые файлы. Поэтому первая резервная копия будет выполняться значительно дольше последующих.

Бэкап запускается ежедневно по таймеру systemd **restic_backup.timer**, таймер запускает юнит systemd **restic_backup.service** из которого происходит непосредственный запуск скрипта **restic_backup**.

Просмотреть список таймеров можно командой:

```
systemctl list-timers
```

Уведомления на электронную почту при падении бэкапа могут быть настроены по инструкции [systemd Timers](#).

Как альтернатива вместо таймеров systemd можно использовать **cron**. Пример cron-задачи:

```
0 1 * * * /usr/local/bin/restic_backup
```

Общий алгоритм работы скрипта такой:

1. Скрипт вначале выполняет проверку инициализации репозитория и инициализирует его если это не так командой `restic init`.
2. Запускается удаление старых резервных копий командой `restic forget`. В скрипте используется опция `--keep-last`, однако можно переписать `/usr/local/bin/restic_backup` на использование другой опции. Сохраняется то количество резервных копий, которое указано в переменной `RESTIC_KEEP` в файле `/etc/restic_backup/settings.sh`.
3. Запускается дамп баз данных. SQL-дампы сохраняются во временную директорию `/tmp/restic_dbs_temp`.
4. Выполняется запуск `restic backup`. В репозиторий загружаются файлы и дампы баз данных.
5. В конце процесса удаляется временная директория с SQL-дампами.

Подробный отладочный лог (xtrace) работы скрипта сохраняется в файл `/var/log/restic_backup.log`. Если запуск бэкапа производится по таймеру systemd, то логи также будут доступны через утилиту **journalctl**:

```
journalctl -u restic_backup.service
```

Список файлов входящих в скрипт **restic_backup**:

- **/usr/local/bin/restic_backup** — непосредственно скрипт резервного копирования.
- **/etc/restic_backup/settings.sh** — файл с настройками резервного копирования.
- **/etc/systemd/system/restic_backup.timer** — таймер systemd.
- **/etc/systemd/system/restic_backup.service** — юнит systemd запускающий бэкап, запускается через таймер.

Подготовка к резервному копированию

Использование Makefile

Makefile это файл для утилиты **GNU make**, помогающий автоматически выполнять некоторые действия по коротким командам.

В процессе эксплуатации **restic_backup** Makefile не используется, однако он помогает в установке скрипта. Доступные команды:

- **make install** — установка скрипта
- **make uninstall** — удаление скрипта вместе со всеми настройками
- **make html** — конвертировать README.md в формат HTML.
- **make pdf** — конвертировать README.md в формат PDF.

Последние две команды требуют установки утилиты **pandoc**.

Настройка хранилища

Будет рассмотрена настройка S3-бакета в качестве репозитория для бэкапов.

1. Создать бакет
2. Указать реквизиты подключения в файле **/etc/restic_backup/settings.sh**. Файл имеет подробные комментарии с описанием всех параметров.

Конфигурирование

Настройка параметров бэкапа

Фрагмент файла **/etc/restic_backup/settings.sh**:

```
# ----- НАСТРОЙКИ RESTIC ----- #  
  
# Ключ доступа к бакету S3. Совпадает с логином аккаунта  
export AWS_ACCESS_KEY_ID=<MY_ACCESS_KEY>  
  
# Секретный ключ доступа к бакету. Можно скопировать из ПУА.  
export AWS_SECRET_ACCESS_KEY=<MY_SECRET_ACCESS_KEY>
```

```
# Адрес репозитория, куда будут сохраняться резервные копии
export RESTIC_REPOSITORY="s3:https://s3.timeweb.com/restic-demo"
```

```
# Пароль для доступа к бэкапам. Все бэкапы шифруются. В случае утраты этого
# пароля получить доступ к резервным копиям будет невозможно!
```

```
export RESTIC_PASSWORD="tiwfBM4i3jDvMA6ah1AIs2HdoJ9w660S"
```

```
# Количество бэков, которые нужно хранить в репозитории
RESTIC_KEEP=2
```

Добавление файлов в бэкап

```
# ----- СПИСОК ФАЙЛОВ И ДИРЕКТОРИЙ ДЛЯ БЭКАПА ----- #
```

```
# Пути к файлам и директориям обязательно должны быть абсолютными.
# Абсолютный путь содержит полный путь до файла/директории начиная от
# корневого каталога системы.
```

```
#
```

```
# Переменная `files` ниже содержит перечень путей. Обратите внимание, что
# пути, которые содержат пробелы должны быть обязательно включены в кавычки.
# Пример: '/home/user/some dir'
```

```
files=(
    /etc
    /home
)
```

Добавление баз данных в бэкап

```
# ----- СПИСОК БАЗ ДАННЫХ ДЛЯ БЭКАПА ----- #
```

```
# Ниже необходимо указать реквизиты подключения к базам данных в формате URI
# (DSN). Запись должна иметь следующий вид:
```

```
#
```

```
# mysql://пользователь:пароль@хост:порт/имя_базы_данных
```

```
#
```

```
# Обратите внимание на разделители между частями URI: ":", "@", "/". Порт
# можно опустить, если используется стандартный 3306.
```

```
#
```

```
# Ниже дан записи для БД bitrix_db и именем пользователя bitrix_usr и
# паролем boRpBnhGn7ue.
```

```
#
```

```
# mysql://bitrix_usr:boRpBnhGn7ue@localhost/bitrix_db
```

```
#
```

```
# Если пароль пользователя базы данных содержит специальные символы, то такой
```

```
# пароль обязательно надо закодировать в percent code. Это можно сделать
# следующей командой в терминале:
#
# echo 'ваш_пароль' | perl -MURI::Escape -wlnе 'print uri_escape $_'
#
# или через онлайн-сервис https://meyerweb.com/eric/tools/dencoder/
#
# Ниже введите URI с реквизитами баз данных так как это показано. Обратите
# внимание, что знак равно не должен ють отделён пробелами. Формат записи
# строго такой как есть.

databases=(
    mysql://пользователь:пароль@хост/база_данных
    mysql://пользователь:пароль@хост/другая_база_данных
)
```

Запуск резервного копирования

Запустить резервное копирование прямо сейчас можно командой:

```
/usr/local/bin/restic_backup
```

Выполнение бэкапа заблокирует ввод в темнал. Желательно запускать команду в сессии `tmux` или `screen`. Пример запуска сессии `tmux`:

```
tmux new -s restic_backup
```

Весь вывод скрипта отправляется в отладочный лог `/var/log/restic_backup.log`.

Отслеживать проесс можно командой:

```
tail -f /var/log/restic_backup.log
```

Восстановление из резервной копии

Для удобства работы перед вызовом команд `restic` следует импортировать файл с настройками.

Это делается командой:

```
source /etc/restic_backup/settings.sh
```

После чего можно выполнять команды `restic` без надобности вручную указывать адрес репозитория и пароль.

Восстановление файлов

Воспользуйтесь инструкцией из официальной документации утилиты: <https://restic.readthedocs.io/en/sta>

Сперва вам потребуется узнать ID снимота (так в restic называются бэкапы). Для этого запустите команду:

```
restic snapshots
```

В терминал будет напечатана таблица со списком всех имеющихся снимотов.

Вы можете восстанавливать снимоты целиком командой вида:

```
restic restore 79766175 --target /tmp/restore-work
```

где 79766175 это ID снимота, а /tmp/restore-work директория, куда будут восстанавливаться файлы.

Другой вариант восстановления — через монтирование репозитория в качестве сетевого диска. Документация: https://restic.readthedocs.io/en/stable/050_restore.html#restore-using-mount

С примонтированного диска вы можете скопировать как все файлы, так и отдельные.

Команды для монтирования:

```
mkdir -p /mnt/restic
restic mount /mnt/restic
```

restic при помощи FUSE создаст виртуальную файловую систему. При этом терминал будет заблокирован. Можно запустить команду на монтирование в сессии tmux или screen и работать с примонтированным диском в другом окне. Либо открыть новую сессию SSH и работать в ней. После можно прервать restic сочетанием клавиш `Ctrl+C` или выполнить команду:

```
fusermount -u /mnt/restic
```

чтобы отмонтировать бэкапы.

Восстановление баз данных

Восстановление баз данных почти ничем не отличается от восстановления обычных файлов.

1. Восстановите дампы в формате **.sql.gz** из репозитория restic
2. Запустите команду для импорта дампа.

```
zcat имя_дампа.sql.gz | mysql -u'пользователь' имя_базы_данных -p'пароль'
```

Удаление старых бэкапов

Удаление производится командой `restic forget`. Смотрите подробное описание в документации https://restic.readthedocs.io/en/stable/060_forget.html