

RuBackup

Система резервного копирования и восстановления данных

Резервное копирование и восстановление СУБД PostgreSQL (Universal)



Содержание

Введение.....	3
Установка клиента RuBackup.....	4
Удаление клиента RuBackup.....	6
Мастер-ключ.....	7
Подготовка сервера СУБД PostgreSQL к резервному копированию с помощью RuBackup.....	8
Принцип выполнения базового резервного копирования PostgreSQL...	13
Принцип выполнения инкрементального резервного копирования PostgreSQL.....	15
Принцип восстановления резервной копии PostgreSQL.....	16
Защитное преобразование резервных копий.....	18
Использование оконного менеджера администратора RuBackup.....	20
Использование клиентского менеджера RuBackup.....	26
Утилиты командной строки клиента RuBackup.....	32
Восстановление резервной копии СУБД PostgreSQL.....	34

Введение

Система резервного копирования RuBackup поддерживает резервное копирование СУБД PostgreSQL 9.6, 10, 11, 12.

Принцип резервного копирования СУБД PostgreSQL с использованием RuBackup состоит в периодическом создании базовых резервных копий экземпляра СУБД по определенному расписанию и резервному копированию архивированных файлов WAL по мере их появления.

В репозитории RuBackup базовые резервные копии будут храниться как полные резервные копии (*full*), а файлы WAL, созданные после базовой резервной копии - как инкрементальные резервные копии (*incremental*). Дифференциальное резервное копирование для PostgreSQL не предусмотрено, и в случае попытки создания правила в глобальном расписании RuBackup для выполнения дифференциальной резервной копии будет создано правило для инкрементального резервного копирования.

Архивные файлы WAL после успешного выполнения резервного копирования могут быть автоматически удалены клиентом RuBackup из каталога, в котором они были созданы.

После окончания резервного копирования будут созданы два файла (архивный и снимок состояния) на медиасerverе, которому принадлежит пул, указанный в правиле резервного копирования. Точное месторасположение файлов указано в записи репозитория системы резервного копирования RuBackup. При необходимости архивный файл может быть преобразован при помощи одного из алгоритмов защитного преобразования на клиенте и сжат. Снимок состояния не преобразовывается, так как в нем располагается информация о наличии в резервной копии WAL файлов, время старта и окончания резервного копирования. В снимке состояния отсутствуют значимые данные СУБД.

Для выполнения резервного копирования СУБД PostgreSQL на хосте клиента должно быть достаточно свободного места для создания резервной копии. Локальное местоположение временного каталога для создания резервных копий определено в файле */opt/rubackup/etc/config.file* параметром *use-local-backup-directory*. В том случае, если на хосте клиента недостаточно места для создания резервной копии, ему может быть предоставлена сетевая файловая система NFS с сервера резервного копирования во временное пользование (см. Руководство по установке серверов резервного копирования и Linux клиентов RuBackup).

Установка клиента RuBackup

Для возможности резервного копирования при помощи RuBackup на сервер должен быть установлен клиент RuBackup и модуль резервного копирования для PostgreSQL соответствующей версии (*rb_module_postgresql9_6*, *rb_module_postgresql10*, *rb_module_postgresql11*, *rb_module_postgresql12* или *rb_module_postgresql13*).

Установка пакетов клиента RuBackup производится из-под учетной записи с административными правами при помощи следующих команд (имена пакетов могут отличаться в зависимости от используемой операционной системы):

```
# dpkg -i rubackup-client.deb
# dpkg -i rubackup-postgresql.deb
```

```
u@ubuntu:~/20220419$ sudo dpkg -i rubackup-client.deb
(Чтение базы данных ... на данный момент установлен 166061 файл и каталог.)
Подготовка к распаковке rubackup-client.deb ...
Распаковывается rubackup-client (2022-04-19) на замену (2022-04-19) ...
Настраивается пакет rubackup-client (2022-04-19) ...
```

```
u@ubuntu:~/20220419/modules$ sudo dpkg -i rubackup-postgresql-13.deb
Выбор ранее не выбранного пакета rubackup-postgresql-13.
(Чтение базы данных ... на данный момент установлено 166059 файлов и каталогов.)
Подготовка к распаковке rubackup-postgresql-13.deb ...
Распаковывается rubackup-postgresql-13 (2022-04-19) ...
Настраивается пакет rubackup-postgresql-13 (2022-04-19) ...
```

Подробно процедуры подготовки к установке, инсталляция, настройка и запуск клиента описаны в документе «Руководство по установке серверов резервного копирования и Linux клиентов RuBackup».

В том случае, если на сервере PostgreSQL нет места для создания резервных копий, можно настроить предоставление клиенту RuBackup NFS файловой системы сервером резервного копирования (подробнее см. Руководство по установке серверов резервного копирования и Linux клиентов RuBackup).

В том случае, если планируется тестирование RuBackup, рекомендуется запускать клиента RuBackup в терминальном режиме с помощью следующей команды:

```
# /opt/rubackup/bin/rubackup_client start
```

Остановить RuBackup клиента можно с помощью следующей команды:

```
# /opt/rubackup/bin/rubackup_client stop
```

Для штатной эксплуатации рекомендуется запускать клиента RuBackup как сервис. Для этого необходимо из административной учетной записи:

1. Включить сервис клиента RuBackup:

```
# sudo systemctl enable  
/opt/rubackup/etc/systemd/system/rubackup_client.service
```

```
root@jatoba:~# sudo systemctl enable /opt/rubackup/etc/systemd/system/rubackup_client.service  
Created symlink /etc/systemd/system/multi-user.target.wants/rubackup_client.service → /opt/rubackup/etc/systemd/system/rubackup_client.service.  
Created symlink /etc/systemd/system/rubackup_client.service → /opt/rubackup/etc/systemd/system/rubackup_client.service.
```

2. Перезагрузить systemctl:

```
# sudo systemctl daemon-reload
```

3. Запустить сервис rubackup_client

```
# sudo systemctl start rubackup_client
```

4. Уточнить статус клиента можно при помощи команды

```
# sudo systemctl status rubackup_client
```

Все дальнейшие операции по резервному копированию и восстановлению СУБД PostgreSQL должны выполняться под учетной записью *root*.

Для целей тестирования рекомендуется включить режим *verbose* в конфигурационном файле */opt/rubackup/etc/config.file*:

```
# RuBackup configuration file  
# created by rb_init  
#  
#  
use-local-backup-directory /rubackup-tmp  
node client  
logfile /opt/rubackup/log/RuBackup.log  
who-is-primary-server antares  
local-schedule-file /opt/rubackup/etc/rubackup.lsf  
client-inet-interface eth0  
parallelism 8  
verbose yes  
rbd_algorithm sha  
rbd_block_size 1048576  
rbd_hash_length 512  
digital-signature yes  
digital-sign-hash sha1  
client-shutdown scenario cancel-if-tasks
```

После изменения конфигурационного файла необходимо перезапустить клиента RuBackup.

По окончании данной процедуры клиент RuBackup настроен. Для возможности выполнения резервного копирования потребуются авторизация клиента системным администратором СРК.

Удаление клиента RuBackup

Порядок удаления клиента RuBackup изложен в документе «Руководство по установке серверов резервного копирования и Linux клиентов RuBackup».

Мастер-ключ

В ходе инсталляции будет создан мастер-ключ для защитного преобразования резервных копий и ключи для электронной подписи, если электронную подпись предполагается использовать.

Важно! При утере ключа вы не сможете восстановить данные из резервной копии, если последняя была преобразована с помощью защитных алгоритмов!

Важно! Ключи рекомендуется после создания скопировать на внешний носитель, а так же распечатать бумажную копию и убрать эти копии в надежное место!

Мастер-ключ рекомендуется распечатать при помощи утилиты hexdump, так как он может содержать неотображаемые на экране символы:

```
root@jatoba:~# hexdump /opt/rubackup/keys/master-key
00000000 e973 053d 10a1 c0c1 40e8 d332 9463 a7ee
00000010 8965 f275 d5e4 a04a d07d a625 d4e8 755f
00000020
root@jatoba:~# █
```

Важно! Секретный ключ необходимо хранить в месте, доступном только тем, кто должен иметь возможность произвести обратное преобразование файла. Утеря ключа делает невозможным обратное преобразование файла!

Подготовка сервера СУБД PostgreSQL к резервному копированию с помощью RuBackup

Подготовка имеющегося сервера с СУБД PostgreSQL

Необходимо установить владельцем каталога для хранения архивных WAL файлов пользователя *postgres*:

```
# chown postgres:postgres /opt/rubackup/mnt/postgresql_archives/
```

Для непрерывного архивирования и восстановления СУБД PostgreSQL требуется включить архивирование WAL. В конфигурационном файле */etc/postgresql/10/main/postgresql.conf* необходимо установить параметры (расположение конфигурационного файла может отличаться от вышеуказанного в зависимости от дистрибутива Linux, проконсультируйтесь по этому поводу у вашего администратора) :

```
wal_level = replica  
archive_mode = on  
archive_command = 'test ! -f  
/opt/rubackup/mnt/postgresql_archives/%f && cp %p  
/opt/rubackup/mnt/postgresql_archives/%f'  
archive_timeout = 300
```

Обязательно необходимо указать значение *data_directory* (если оно не определено), в противном случае модуль резервного копирования не сможет определить местоположение файлов СУБД:

```
data_directory = '/var/lib/postgresql/10/main'
```

После внесения изменений необходимо перезапустить PostgreSQL:

```
# service postgresql restart
```

Параметр *archive_command* должен содержать каталог в файловой системе сервера PostgreSQL, в который будут копироваться архивируемые сегменты WAL.

В настройках RuBackup для каждой СУБД PostgreSQL в файле */opt/rubackup/etc/rb_module_postgresql12.conf* имеется параметр *archive_catalog*, содержащий значение каталога, в котором предполагается создание архивных WAL файлов. Значение этого параметра по умолчанию:

```
/opt/rubackup/mnt/postgresql_archives/
```

При планировании инсталляции RuBackup вы можете назначить для хранения архивных WAL файлов выделенное хранилище требуемого размера и сделать на него ссылку на том сервере PostgreSQL, где это требуется.

Объем необходимого пространства под архивные файлы WAL на сервере PostgreSQL можно оценить следующим образом:

1. Один файл WAL по умолчанию имеет размер 16МБайт.

2. Необходимо оценить как часто создается новый WAL файл (максимальный период определяется параметром *archive_timeout* в конфигурационном файле СУБД). Предлагаемое выше значение - 300 секунд или 12 раз в час, но в реальности при высокой нагрузке этот период может оказаться короче и создаваться WAL файл будет чаще.

3. Если настроить правило инкрементального резервного копирования таким образом, что архивный WAL файл будет скопирован сразу же после его появления в каталоге, то потребуется минимум 184МБ (12 раз в час *16МБ). Целесообразно заложить как минимум двукратный запас свободного места для этого каталога, в противном случае, при невозможности переместить архивный WAL файл в каталог из-за недостатка свободного места может привести к деградации производительности СУБД в целом.

Важно! Указанный каталог должен быть доступен для записи и чтения пользователю *postgres*, а так же пользователю, под контролем которого работает клиент RuBackup.

Обеспечить это можно командой:

```
# chown postgres:postgres /opt/rubackup/mnt/postgresql_archives/
```

Исходя из этих же вводных можно оценить требуемый объем хранилища на сервере резервного копирования RuBackup.

Для правильной работы клиента RuBackup значения параметра *archive_catalog* в конфигурации RuBackup и параметра *archive_command* в конфигурационном файле PostgreSQL должны быть идентичны для одной и той же СУБД.

Параметр *archive_timeout* определяет период времени в секундах, по окончании которого сервер PostgreSQL должен переключиться на новый сегмент WAL.

После изменения параметров конфигурационного файла необходимо рестартовать PostgreSQL при помощи команды:

```
# sudo systemctl restart postgresql
```

Создание пользователя СУБД для безопасного выполнения базовой резервной копии PostgreSQL

В случае настройки резервного копирования PostgreSQL в ОС Astra Linux SE 1.6 необходимо в файле */etc/parsec/mswitch.conf* для параметра *zero_if_notfound* установить значение *yes* и после этого перезагрузить сервис PostgreSQL:

```
# sudo service postgresql restart
```

Пользователь для выполнения операции создания базовой резервной копии должен обладать правами на выполнение функций начала и окончания резервного копирования экземпляра PostgreSQL.

Вызовите *psql* при помощи команды:

```
# sudo -u postgres psql
```

В *psql* создайте пользователя *rubackup_backuper*, в качестве пароля укажите желаемый пароль вместо 12345:

```
# create user rubackup_backuper password '12345';
# alter role rubackup_backuper with login;
# grant execute on function pg_start_backup to rubackup_backuper;
# grant execute on function pg_stop_backup(bool, bool) to
rubackup_backuper;
# grant execute on function pg_switch_wal to rubackup_backuper;
# grant pg_read_all settings to rubackup_backuper.
```

Вместо пользователя *rubackup_backuper* вы можете создать любого другого с соответствующим набором прав. В файле */opt/rubackup/etc/rb_module_postgresql12.conf* необходимо указать правильное значение пользователя и его пароль:

```

root@postgresql:~# cat /opt/rubackup/etc/rb_module_postgresql10.conf
username rubackup_backuper
password 12345
port 5432
archive_catalog /opt/rubackup/mnt/postgresql_archives
pg_ctl /usr/lib/postgresql/10/bin/pg_ctl
exclude_file /opt/rubackup/etc/postgresql.exclude
auto_remove_wal yes
direct_restore yes
postgresql_admin _ postgres

```

Настройки файла */opt/rubackup/etc/rb_module_postgresql12.conf* представлены в таблице 1.

Таблица 1 – Настройки файла */opt/rubackup/etc/rb_module_postgresql12.conf*

Параметр	Назначение	Значение по умолчанию
username	Имя пользователя в СУБД PostgreSQL, обладающего правами выполнять резервное копирование	rubackup_backuper
password	Пароль username	
port	Порт для соединения с СУБД	5432
archive_catalog	Каталог для хранения архивных WAL	/opt/rubackup/mnt/postgresql_archives
pg_ctl	Местонахождение pg_ctl	/usr/lib/postgresql/10/bin/pg_ctl
exclude_file	Файлы или каталоги, которые необходимо исключить из резервной копии	/opt/rubackup/etc/postgresql.exclude
auto_remove_wal	В случае значения yes архивные WAL будут удалены из каталога <i>archive_catalog</i> после выполнения резервного копирования (если они включены в резервную копию)	yes
direct_restore	При значении yes : При восстановлении резервной копии служба postgresql будет остановлена, каталог кластера баз данных будет очищен, файлы цепочки резервных копий будут восстановлены в каталог кластера баз данных и будут выполнены все необходимые настройки для восстановления СУБД при старте службы postgresql. Старт службы postgresql необходимо	yes

Параметр	Назначение	Значение по умолчанию
	<p>выполнить в ручном режиме.</p> <p>При значении no:</p> <p>Файлы цепочки резервных копий будут восстановлены в выбранный пользователем каталог. Восстановление СУБД в данном случае выполняется администратором СУБД в ручном режиме</p>	
postgresql_admin	Login администратора PostgreSQL в операционной системе	postgres

Начиная с версии 12, вместе с модулем rubackup необходимо внести в конфигурационный файл *postgresql.conf* значения параметров *restore_command* и *archive_cleanup_command* в виде:

```
restore_command = 'cp /opt/rubackup/mnt/postgresql_archives/%f %p'
archive_cleanup_command = '/usr/bin/pg_archivecleanup
/opt/rubackup/mnt/postgresql_archives/ %r'
```

Затем необходимо перезагрузить сервис *postgresql*:

```
# sudo systemctl restart postgresql.service
```

После выполнения подготовки сервера СУБД PostgreSQL к выполнению резервного копирования необходимо перезапустить клиента RuBackup:

```
# rubackup_client stop
# rubackup_client start
```

В результате клиент должен сообщить о том, что модуль для резервного копирования PostgreSQL готов к работе:

```
Try to check module: PostgreSQL 10 ...
Execute OS command: /opt/rubackup/modules/rb_module_postgresql10 -t 2>&1
Connected to local PostgreSQL as rubackup_backuper
PostgreSQL version: 10.14 (Ubuntu 10.14-0ubuntu0.18.04.1)
Close connection local PostgreSQL
... module PostgreSQL 10 was checked successfully
```

Принцип выполнения базового

резервного копирования PostgreSQL

В ходе базового резервного копирования выполняются действия (sql запросы от имени пользователя `rubackup_backuper`), аналогичные следующим командам:

1. Старт резервного базового копирования:

```
postgres=# \c postgres rubackup_backuper
postgres=> SELECT pg_start_backup('label', false, false);
```

2. Копирование файлов кластера баз данных:

```
postgres@postgresql:~$ tar cvfp /tmp/pg-backup.tar --
exclude=postmaster.pid --exclude=postmaster.opts --
exclude=pg_replslot/* --exclude=pg_dynshmem/* --
exclude=pg_notify/* --exclude=pg_serial/* --
exclude=pg_snapshots/* --exclude=pg_stat_tmp/ --
exclude=pg_subtrans/* --exclude=pgsql_tmp*
/var/lib/postgresql/10/main
```

В указанной выше команде из копирования по умолчанию исключаются ряд файлов и каталогов, наличие которых в резервной копии не влияет на успешное восстановление данных СУБД. Однако, вы можете изменить этот перечень, переопределив его в файле `/opt/rubackup/etc/postgresql.exclude` (если файл будет пуст, то в резервную копию войдут все файлы, если его не будет, то резервное копирование будет выполнено с исключениями по умолчанию).

3. Стоп резервного копирования:

```
postgres=> SELECT pg_stop_backup(false, true);
```

4. Функция `pg_stop_backup` возвратит одну строку с тремя значениями. Второе из них нужно записать в файл `backup_label` в корневой каталог резервной копии. Третье значение, если оно не пустое, должно быть записано в файл `tablespace_map`. Эти значения крайне важны для восстановления копии и должны записываться без изменений.
5. Копирование WAL файлов, активных в ходе выполнения резервного копирования (потребуется отсечь файлы, созданные до начала операции создания базовой резервной копии, в команде ниже это не учтено):

```
postgres@postgresql:~$ tar cvp /tmp/pg-backup-wal-files.tar  
/opt/rubackup/mnt/postgresql_archives/*
```

Диапазон файлов, которые необходимо скопировать, указан в последнем созданном файле с расширением backup в каталоге */opt/rubackup/mnt/postgresql_archives/*.

Принцип выполнения инкрементального резервного копирования PostgreSQL

Инкрементальное резервное копирование состоит в резервировании новых архивных WAL файлов, которые были созданы в каталоге */opt/rubackup/mnt/postgresql_archives/* после окончания последнего полного или инкрементального резервного копирования.

Принцип восстановления резервной копии PostgreSQL

Данный ручной метод может быть использован при ручном восстановлении служебной базы данных RuBackup, если для ее работы используется СУБД PostgreSQL и выполнялось ее резервное копирование.

Перед восстановлением базы данных рекомендуется сделать резервную копию всех имеющихся файлов в каталоге кластера баз данных, а так же запретить доступ пользователей к ней путем внесения соответствующих изменений в файл *pg_hba.conf*.

Для восстановления СУБД PostgreSQL необходимо проделать следующие шаги:

1. Остановить экземпляр PostgreSQL, если он работает:

```
# sudo -iu postgres /usr/lib/postgresql/10/bin/pg_ctl stop -D /var/lib/postgresql/10/main
```

2. Сделать резервную копию файлов каталога кластера баз данных, для возможности отката (в примере использован каталог *~/emergency_copy*, в нем должно быть достаточно места для выполнения данной операции):

```
# sudo -iu postgres (cd /var/lib/postgresql/10/main && tar cfv - *) | (cd ~/emergency_copy && tar xf - )
```

3. Очистить каталог кластера баз данных:

```
# sudo -iu postgres rm -rf /var/lib/postgresql/10/main/*
```

4. Восстановить данные из резервных копий (например, установив значение **no** для параметра **direct_restore** в файле */opt/rubackup/etc/rb_module_postgresql10.conf* и выполнив восстановление резервной копии при помощи *rbc* или *rb_archives* в какой-либо каталог). Важно, чтобы все файлы сохранили изначальные разрешения и владельцев. Архивные WAL файлы из резервных копий необходимо разместить в каталоге */opt/rubackup/mnt/postgresql_archives*

5. Для PostgreSQL версий до 11-й включительно создать файл *recovery.conf* со следующим содержанием:

```
restore_command = 'cp /opt/rubackup/mnt/postgresql_archives/%f
%p'
```

6. Запустить восстановление PostgreSQL:

```
# sudo -iu postgres postgres /usr/lib/postgresql/10/bin/pg_ctl -
D /var/lib/postgresql/10/main
```

Если вы установили параметр `recovery_target_time` в файле `recovery.conf` для восстановления базы данных на определенный момент времени, то после старта PostgreSQL в режиме восстановления необходимо выполнить в psql следующую команду:

```
# select pg_wal_replay_resume();
```

Защитное преобразование резервных

КОПИЙ

При необходимости ваши резервные копии могут быть преобразованы на клиенте сразу после выполнения резервного копирования. Таким образом, критичные данные будут недоступны для администратора RuBackup или для иных лиц, которые могли бы получить доступ к резервной копии (например, во внешнем хранилище картриджей ленточной библиотеки или на площадке провайдера облачного хранилища для ваших резервных копий).

Ключ для преобразования резервных копий располагается на клиенте в файле `/opt/rubackup/keys/master-key`. Пользователь сам должен задать ключ длиной 256 бит (32 байта).

Защитное преобразование осуществляется специальной утилитой `rbcrypt`. Автоматическое защитное преобразование и обратное преобразование резервных копий клиентом RuBackup возможны при помощи ключей длиной 256 бит, однако утилита `rbcrypt` поддерживает ключи длиной 128, 256, 512 и 1024 бита (в зависимости от выбранного алгоритма преобразования). Если необходимо для правила глобального расписания выбрать особый режим преобразования, с длиной ключа, отличной от 256 бит и с ключом, располагающемся в другом месте, то вы можете воспользоваться возможностью сделать это при помощи скрипта, выполняющегося после выполнения резервного копирования (определяется в правиле глобального расписания администратором RuBackup). При этом необходимо, чтобы имя преобразованного файла осталось таким же, как и ранее, иначе задача завершится с ошибкой. Проводить обратное преобразование такого файла после восстановления его из резервной копии следует вручную при помощи утилиты преобразования. При таком режиме работы нет необходимости указывать алгоритм преобразования в правиле резервного копирования, либо архив будет преобразован еще раз автоматически с использованием мастер-ключа.

Для выполнения защитного преобразования доступны алгоритмы, представленные в таблице 2.

Таблица 2 – Алгоритмы защитного преобразования

Наименование алгоритма	Поддерживаемая rbcrypt длина ключа, бит	Примечание
Anubis	128, 256	
Aria	128, 256	
CAST6	128, 256	
Camellia	128, 256	
Kalyna	128, 256, 512	Украинский национальный стандарт ДСТУ 7624:2014
Kuznyechik	256	Российский национальный стандарт ГОСТ Р 34.12-2015
MARS	128, 256	
Rijndael	128, 256	Advanced Encryption Standard (AES)
Serpent	128, 256	
Simon	128	
SM4	128	Chinese national standard for Wireless LAN
Speck	128, 256	
Threefish	256, 512, 1024	
Twofish	128, 256	

Использование оконного менеджера администратора RuBackup

Оконное приложение «Менеджер администратора RuBackup» (RBM) предназначено для общего администрирования серверной группировки RuBackup, управления клиентами резервного копирования, глобальным расписанием резервного копирования, хранилищами резервных копий и пр. RBM может быть запущено администратором на основном сервере резервного копирования RuBackup.

Запуск менеджера администратора RBM:

Вариант 1:

```
# sudo LD_LIBRARY_PATH=/opt/rubackup/lib /opt/rubackup/bin/rbm
```

Вариант 2:

```
# ssh -X root@you_rubackup_server
```

```
# /opt/rubackup/bin/rbm
```

Примечание – Подробно порядок запуска Менеджера администратора RuBackup изложен в документе «Руководство по установке серверов резервного копирования и Linux клиентов RuBackup».

В том случае, если клиент RuBackup был установлен, но не авторизован, в нижней части окна RBM будет сообщение о том, что найдены неавторизованные клиенты (рисунок 1).

Все новые клиенты должны быть авторизованы в системе резервного копирования (рисунок 2).

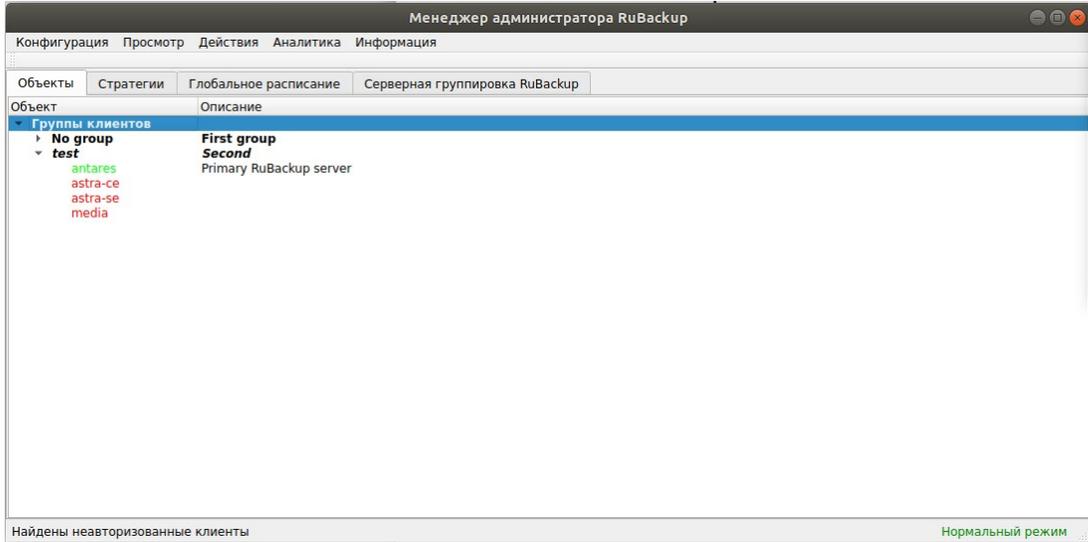


Рисунок 1

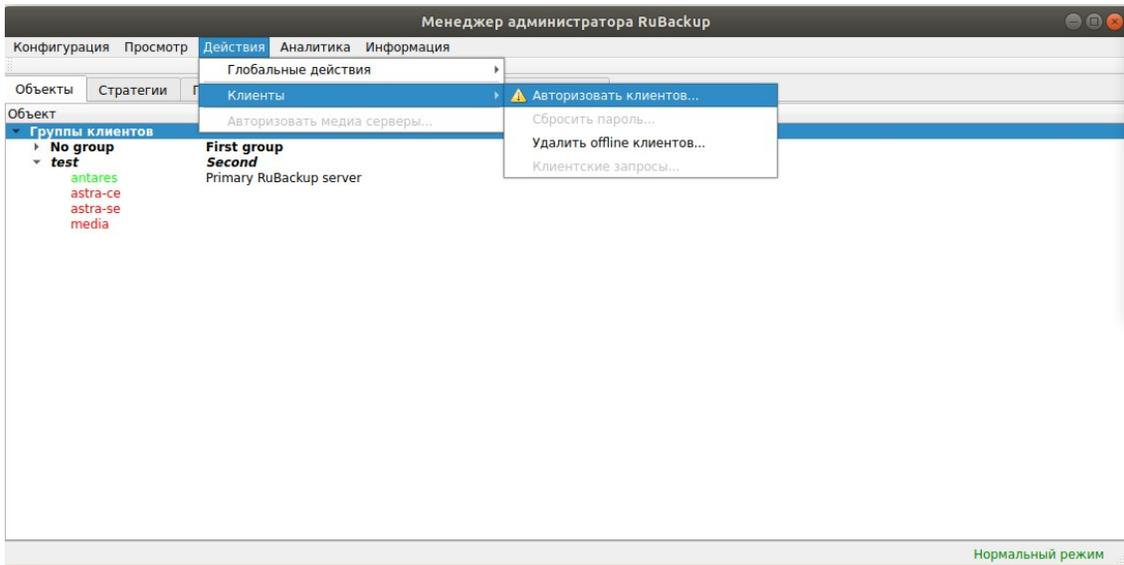


Рисунок 2

Неавторизованные клиенты будут видны в окне (рисунок 3).

Имя хоста	Тип ОС	ОС дистрибутив	MAC	IPv4	IPv6	Последний раз на связи
1 srv.brest.loc	Linux	astra	52:54:00:6c:7c:3b	10.49.1.10		23.04.2020 16:01

Buttons: Закрыть, Авторизовать, Удалить

Рисунок 3

После авторизации новый клиент будет виден в главном окне RBM (рисунок 4).

Менеджер администратора RuBackup

Конфигурация | Просмотр | Действия | Аналитика | Информация

Объекты | Стратегии | Глобальное расписание | Серверная группировка RuBackup

Объект	Описание
Группы клиентов	
No group	First group
redos.rubackup.local	
srv.brest.loc	
test	Second
antares	Primary RuBackup server
astra-ce	
astra-se	
media	

Нормальный режим

Рисунок 4

Клиенты могут быть сгруппированы администратором по какому-либо общему признаку. В случае необходимости восстанавливать резервные копии на другом хосте клиенты должны принадлежать к разделяемой группе (такая группа отмечается шрифтом *italic*).

Для того, чтобы выполнять регулярное резервное копирование СУБД PostgreSQL, необходимо создать правило в глобальном расписании.

Выберите клиентский хост, на котором установлен *front* ПК-Брест и добавьте правило резервного копирования (рисунок 5):

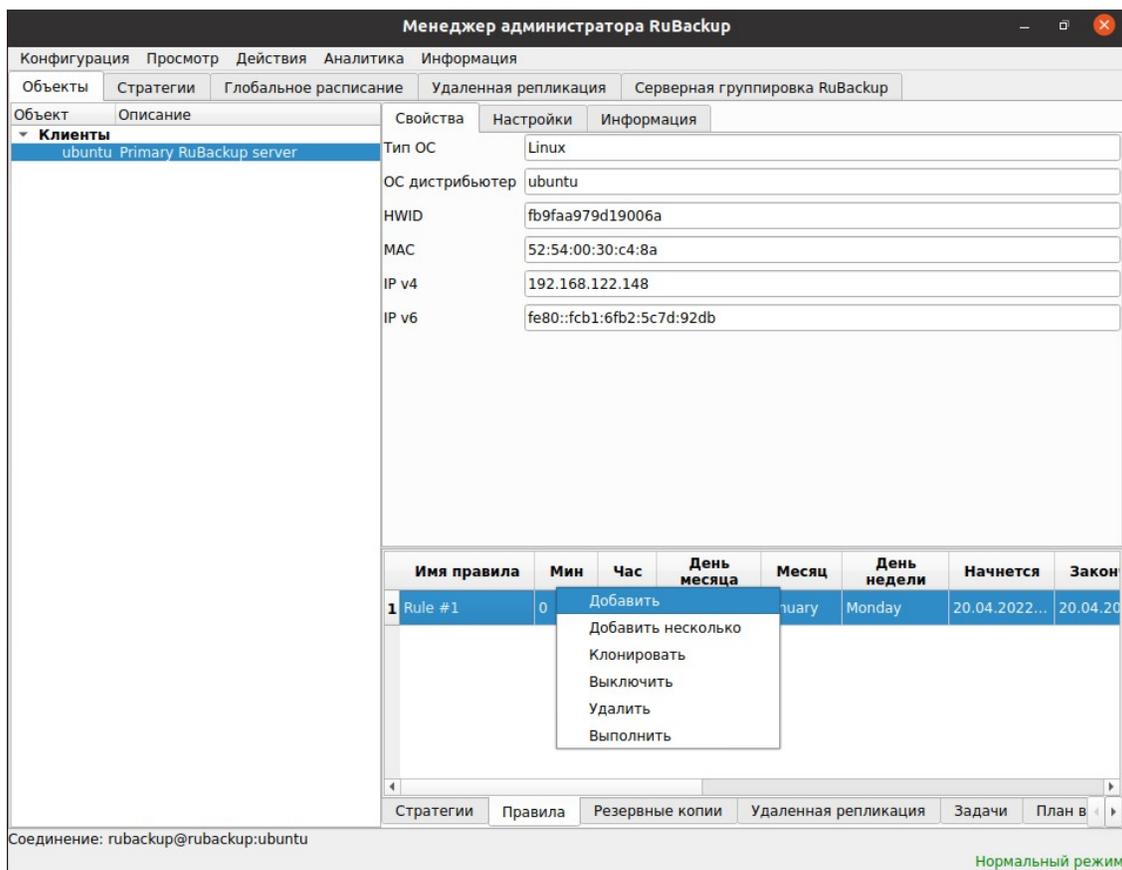


Рисунок 5

Выберите тип ресурса «PostgreSQL» (рисунок 6).

В качестве ресурса автоматически будет подставлено значение главного конфигурационного файла СУБД:

/etc/postgresql/12/main/postgresql.conf.

Установите прочие настройки: расписание резервного копирования, тип резервного копирования, максимальный объем для резервных копий данного правила, срок хранения, через какой промежуток времени требуется выполнить проверку резервной копии.

На вкладке «Дополнительно» можно установить разрешение для клиента удалять резервные копии, установить автоматическое удаление устаревших резервных копий или определить условие их перемещения в другой пул (рисунок 7).

Добавить правило в глобальное расписание (на rubackup-stress.rubackup.local)

Основное | Уведомления | Дополнительно

Имя:

Клиент: HWID:

Пул:

Максимальный объем РК правил: Гб, для данного клиента нет Приоритет:

Ресурс:

Тип ресурса:

Образец расписания Все

Минута:

Час:

День месяца:

Месяц:

День недели:

Тип РК:

Преобразование:

Период действия правила

Начало:

Окончание:

Проверять РК через

Включить после создания

Срок хранения РК:

Рисунок 6

Добавить правило в глобальное расписание

Основное | Уведомления | Дополнительно

Устаревшие резервные копии:

Автоматическое удаление РК Информировать:

Резервные копии:

Переместить в пул: если старше чем

Клиенту разрешено удалять резервные копии этого правила из репозитория

Рисунок 7

Вновь созданное правило будет обладать статусом «wait», это означает что оно не будет порождать задач на выполнение резервного копирования до той поры, пока администратор RuBackup не запустит его и оно изменит свой статус на «run». При необходимости работу правила можно будет приостановить или запустить в любой момент времени по желанию администратора. Так же администратор может инициировать немедленное создание задачи при статусе правила «wait».

Правило глобального расписания имеет срок жизни, определяемый при его создании, а так же предусматривает следующие возможности:

1) выполнить скрипт на клиенте перед началом резервного копирования;

2) выполнить скрипт на клиенте после успешного окончания резервного копирования;

3) выполнить скрипт на клиенте после неудачного завершения резервного копирования;

4) выполнить преобразование резервной копии на клиенте;

5) периодически выполнять проверку целостности резервной копии;

6) хранить резервные копии определенный срок, а после его окончания удалять их из хранилища резервных копий и из записей репозитория, либо просто уведомлять пользователей системы резервного копирования об окончании срока хранения;

7) через определенный срок после создания резервной копии автоматически переместить ее в другой пул хранения резервных копий, например на картридж ленточной библиотеки;

8) уведомлять пользователей системы резервного копирования о результатах выполнения тех или иных операций, связанных с правилом глобального расписания.

При создании задачи RuBackup она появляется в главной очереди задач. Отслеживать исполнение правил может как администратор, с помощью RBM, так клиент при помощи утилиты командной строки *rb_tasks*.

После успешного завершения резервного копирования резервная копия будет размещена в хранилище резервных копий, а информация о ней будет размещена в репозитории RuBackup.

Использование клиентского менеджера RuBackup

В том случае, если клиент RuBackup устанавливается на ОС без графической оболочки, то для возможности использовать RBC потребуется установить следующие пакеты:

```
libgl1-mesa-dev  
libxkbcommon-x11-0  
libfontconfig1
```

Принцип взаимодействия клиентского менеджера с системой резервного копирования состоит в том, что пользователь может сформировать ту или иную команду (желаемое действие) и отправить ее серверу резервного копирования RuBackup. Взаимодействие пользователя с сервером резервного копирования производится через клиента (фоновый процесс) резервного копирования. Клиентский менеджер отправляет команду пользователя клиенту, клиент отправляет ее серверу. В том случае, если действие допустимо, то сервер RuBackup отдаст обратную команду клиенту и/или перенаправит ее медиасерверу RuBackup для дальнейшей обработки. Это означает, что клиентский менеджер обычно не ожидает завершения того или иного действия, но ожидает ответа от клиента, что задание принято. Это позволяет инициировать параллельные запросы клиента к серверу резервного копирования, но требует от пользователя самостоятельно контролировать чтобы не было “встречных” операций, когда происходит восстановление данных, и в этот же момент эти же данные требуются для создания новой резервной копии. После того, как вы отдали ту или иную команду при помощи клиентского менеджера, вы можете просто закрыть приложение, все действия будут выполнены системой резервного копирования (однако стоит дождаться сообщения что задание принято к исполнению и проконтролировать это в закладке “Задачи”).

Графический интерфейс клиентского менеджера поддерживает русский и английский языки.

Запуск клиентского менеджера (для примера использован хост postgresql):

```
# ssh -X root@postgresql
```

```
andreyk@antares:~$ ssh -X root@postgresql
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-76-generic x86_64)
Last login: Mon Oct 26 10:26:23 2020 from 192.168.0.6
root@postgresql:~# rbc&
```

В том случае, если клиентская операция выполняется впервые, потребуется ввести пароль клиента (рисунок 8). Без ввода пароля получить резервную копию для клиента из хранилища невозможно.

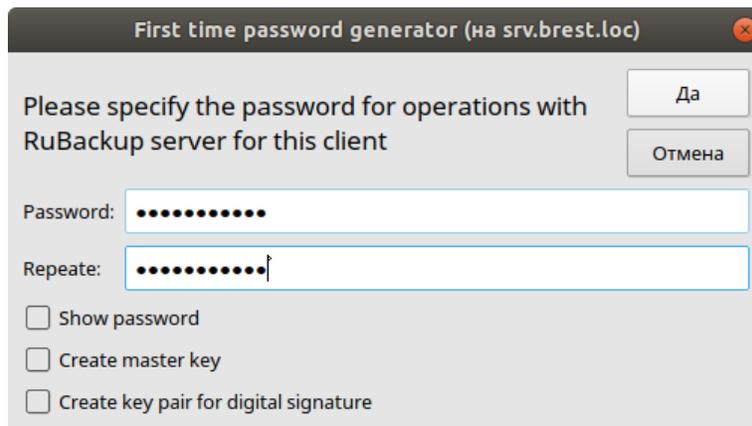


Рисунок 8

В случае успешного выполнения появится окно (рисунок 9):

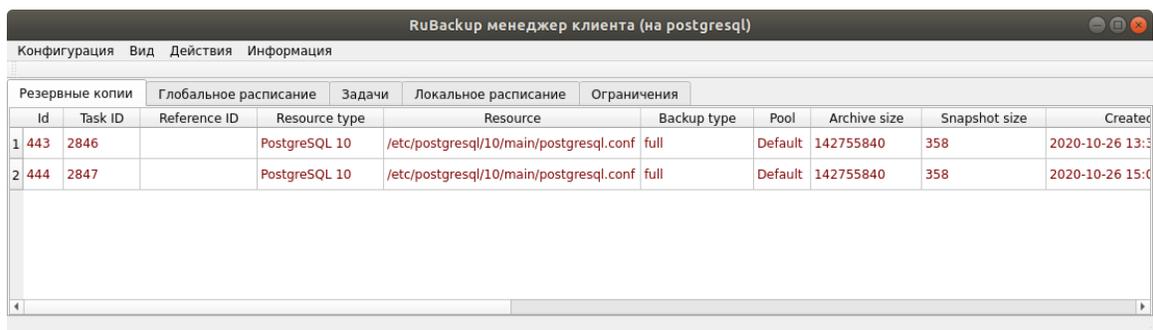


Рисунок 9

Хэш пароля восстановления хранится в базе данных RuBackup сервера. При необходимости можно изменить пароль при помощи клиентского менеджера (Меню «Конфигурация» → «Изменить пароль»).

На главной странице клиентского менеджера расположены переключающиеся вкладки, позволяющие управлять резервными копиями, расписанием резервного копирования и просматривать текущие задачи клиента.

Вкладка «Резервные копии»



Id	Task ID	Reference ID	Resource type	Resource	Backup type	Pool	Archive size	Snapshot size	Created
1	443	2846	PostgreSQL 10	/etc/postgresql/10/main/postgresql.conf	full	Default	142755840	358	2020-10-26 13:3
2	444	2847	PostgreSQL 10	/etc/postgresql/10/main/postgresql.conf	full	Default	142755840	358	2020-10-26 15:0

Рисунок 10

В таблице вкладки «Резервные копии» содержится информация обо всех резервных копиях клиента, которые хранятся в репозитории RuBackup. Инкрементальные резервные копии ссылаются на полные резервные копии или предыдущие инкрементальные, так что при необходимости восстановить данные можно одной командой инициировать восстановление всей цепочки резервных копий.

Во вкладке «Резервные копии» пользователю доступны следующие действия:

1. Удалить выбранную резервную копию.

Это действие возможно в том случае, если в правиле глобального расписания есть соответствующее разрешение. Кроме того, при необходимости выполнить удаление резервной копии потребуется ввести пароль клиента.

2. Восстановить цепочку резервных копий.

Это действие запускает процесс восстановления цепочки резервных копий на локальной файловой системе клиента. При восстановлении резервной копии или цепочки резервных копий пользователь должен выбрать место для восстановления файлов резервной копии. Рекомендуется использовать временный каталог для операций с резервными копиями (например, `/rubackup-tmp`). В том случае, если в файле `/opt/rubackup/etc/rb_module_postgresql10.conf` параметр `direct_restore` имеет значение `yes`, то произойдет остановка сервиса PostgreSQL, очистка каталога кластера баз данных, перемещение восстановленной полной резервной копии в каталога кластера баз данных (а инкрементальных копий в каталог с архивными WAL) и будут выполнены все необходимые настройки для восстановления СУБД при старте службы postgresql. Старт службы postgresql необходимо выполнить в ручном режиме. В том случае, если в файле `/opt/rubackup/etc/rb_module_postgresql10.conf` параметр `direct_restore` имеет значение `no`, то восстановленные резервные копии можно будет обнаружить в

выбранном для восстановления каталоге и далее провести восстановление СУБД в ручном режиме.

Клиентский менеджер не ожидает окончания восстановления всех резервных копий, пользователь должен проконтролировать во вкладке «Задачи» что все созданные задачи на восстановление данных завершились успешно (статус задач «Done»). Для успешного выполнения этого действия требуется наличие достаточного свободного места в каталоге, предназначенном для создания и временного хранения резервных копий (см.опцию *use-local-backup-directory*).

3. Проверить резервную копию.

Это действие инициирует создание задачи проверки резервной копии. В том случае, если резервная копия была подписана цифровой подписью, то будут проверены размер файлов резервной копии, md5 сумма и проверена сама резервная копия. Если резервная копия не была подписана цифровой подписью, то будут проверены размер файлов резервной копии и md5 сумма.

Вкладка «Глобальное расписание»

RuBackup менеджер клиента (на postgresql)											
Конфигурация Вид Действия Информация											
Резервные копии		Глобальное расписание			Задачи		Локальное расписание		Ограничения		
Id	Rule name	Storage capacity, GB	Min	Hour	Day of month	Month	Day of week	Validity start period	Validity end period	Resource type	
1	85 PostgreSQL full backup	100	0	0	*	*	Sunday	2020-10-26 13:17:00+03	2021-10-26 13:17:00+03	PostgreSQL 10	/etc/pos

Рисунок 11

В таблице вкладки «Глобальное расписание» содержится информация обо всех правилах в глобальном расписании RuBackup для этого клиента.

В закладке «Глобальное расписание» пользователю доступны следующие действия:

1. Запросить новое правило.

Это действие вызывает диалог подготовки нового правила в глобальном расписании RuBackup для данного клиента. Запрос на добавление правила требует одобрения администратора RuBackup, одобрение может быть сделано в оконном менеджере администратора RuBackup.

2. Запросить удалить правило из глобального расписания.

Это действие формирует запрос к администратору RuBackup об удалении выбранного пользователем правила из глобального расписания RuBackup. Запрос на удаление правила требует одобрения администратора RuBackup, одобрение может быть сделано в оконном менеджере администратора RuBackup.

Вкладка «Задачи»

Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created
1	Backup global	PostgreSQL 10	/etc/postgresql/10/main/postgresql.conf	full			0	Default	Transmission	2020-10-26 ...

Рисунок 12

В таблице вкладки «Задачи» содержится информация обо всех задачах в главной очереди заданий RuBackup для этого клиента. В зависимости от настроек резервного сервера RuBackup выполненные задачи и задачи, завершившиеся неудачно, через какое-то время могут быть автоматически удалены из главной очереди задач. Информация о выполнении заданий фиксируется в специальном журнале задач сервера RuBackup, при необходимости статус любой задачи, даже удаленной из очереди, можно уточнить у администратора RuBackup. Так же информация о выполнении задач клиента заносится в локальный журнальный файл на клиенте. В клиентском менеджере можно открыть окно отслеживания журнального файла (меню «Информация» → «Журнальный файл»).

Вкладка «Локальное расписание»

Во вкладке «Локальное расписание» можно определить правила, задаваемые клиентом для тех или иных локальных ресурсов. Для работы локального расписания эта возможность должна быть включена администратором RuBackup для клиента.

Вкладка «Ограничения»

Во вкладке «Ограничения» могут быть определены локальные ресурсы, резервное копирование которых нежелательно. Для работы локальных ограничений эта возможность должна быть включена администратором RuBackup для клиента.

Утилиты командной строки клиента

RuBackup

Для управления RuBackup со стороны клиента, помимо клиентского оконного менеджера, можно воспользоваться утилитами командной строки:

rb_archive

Утилита предназначена для просмотра списка резервных копий клиента в системе резервного копирования, создания срочных резервных копий, их удаления, проверки и восстановления.

```
root@postgresql:~#
root@postgresql:~# rb_archive
Id | Ref ID | Resource | Resource type | Backup type | Created | Crypto | Signed | Status
-----+-----+-----+-----+-----+-----+-----+-----+-----
443 | | /etc/postgresql/10/main/postgresql.conf | PostgreSQL 10 | full | 2020-10-26 13:36:04+03 | nocrypt | True | Not Verified
444 | | /etc/postgresql/10/main/postgresql.conf | PostgreSQL 10 | full | 2020-10-26 15:02:05+03 | nocrypt | True | Not Verified
445 | | /etc/postgresql/10/main/postgresql.conf | PostgreSQL 10 | full | 2020-10-26 15:08:19+03 | nocrypt | True | Not Verified
root@postgresql:~#
```

rb_schedule

Утилита предназначена для просмотра имеющихся правил клиента в глобальном расписании резервного копирования.

```
root@postgresql:~# rb_schedule
Id | Name | Resource type | Resource | Backup type | Status
-----+-----+-----+-----+-----+-----
85 | PostgreSQL full backup | PostgreSQL 10 | /etc/postgresql/10/main/postgresql.conf | full | wait
root@postgresql:~#
```

rb_tasks

Утилита предназначена для просмотра задач клиента, которые присутствуют в главной очереди задач системы резервного копирования.

```
root@postgresql:~# rb_tasks
Id | Task type | Resource | Backup type | Status | Created
-----+-----+-----+-----+-----+-----
2848 | Backup global | /etc/postgresql/10/main/postgresql.conf | full | Done | 2020-10-26 15:08:13+03
root@postgresql:~#
```

rbcrypt

Утилита клиента RuBackup для защитного преобразования файлов на стороне клиента RuBackup.

Более подробно ознакомиться с функциями утилит командной строки можно при помощи команды `man` или в руководстве «Утилиты командной строки RuBackup».

Восстановление резервной копии

СУБД PostgreSQL

В том случае, если в файле `/opt/rubackup/etc/rb_module_postgresql10.conf` параметр `direct_restore` имеет значение `yes`, то произойдет остановка сервиса `postgresql`, очистка каталога кластера баз данных, перемещение восстановленной полной резервной копии в каталога кластера баз данных (а инкрементальных копий в каталог с архивными WAL) и будут выполнены все необходимые настройки для восстановления СУБД при старте службы `postgresql`. Старт службы `postgresql` необходимо выполнить в ручном режиме.

В том случае, если в файле `/opt/rubackup/etc/rb_module_postgresql10.conf` параметр `direct_restore` имеет значение `no`, то восстановленные резервные копии можно будет обнаружить в выбранном для восстановления каталоге и далее провести восстановление СУБД в ручном режиме.

Для восстановления резервной копии СУБД PostgreSQL необходимо определить идентификатор резервной копии, которую необходимо восстановить, например, при помощи команды `rb_archives`:

```
root@postgresql:~# rb_archives
```

Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
443		/etc/postgresql/10/main/postgresql.conf	PostgreSQL 10	full	2020-10-26 13:36:04+03	nocrypt	True	Not Verified
444		/etc/postgresql/10/main/postgresql.conf	PostgreSQL 10	full	2020-10-26 15:02:05+03	nocrypt	True	Not Verified
445		/etc/postgresql/10/main/postgresql.conf	PostgreSQL 10	full	2020-10-26 15:08:19+03	nocrypt	True	Trusted

В приведенном примере в системе резервного копирования присутствуют три резервные копии с идентификаторами 443, 444 и 445. Для восстановления резервной копии 445 необходимо выполнить команду

```
# rb_archives -x 445
```

В случае успешно принятой задачи команда вернет список созданных задач, а восстановление будет происходить в фоновом режиме.

```
root@postgresql:~# rb_archives -x 445
Password:
----> Restore archive chain: 445 < ----
Record ID: 445 has status: Trusted
[RBC] Request to restore next archive(s) ID from repository: 445 to: /root
TASK WAS ADDED TO QUEUE:2850
```

