RuBackup

Система резервного копирования и восстановления данных

Резервное копирование и восстановление СУБД PostgreSQL

Версия 2.0 2019

Оглавление

Введение

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

В репозитории RuBackup базовые резервные копии будут хранится 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 выполнение дифференциальной резервной копии будет создано правило для инкрементального резервного копирования.

Архивные файлы WAL после успешного выполнения резерного 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 медиа-сервере, 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 и 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 копия определено в файле /opt/rubackup/etc/config.file параметром *use-local-backup-directory*.

Установка клиента RuBackup на сервер PostgreSQL

Для возможности резервного копирования PostgreSQL при помощи RuBackup на сервер, где располагается PostgreSQL должен быть установлен клиент RuBackup и клиентский менеджер RuBackup PostgreSQL (rbc_postgresql).

Подробно процедура установки клиента описана в «Руководстве по установке RuBackup».

Клиент RuBackup представляет собой фоновое системное приложение (демон или сервис), обеспечивающее взаимодействие с серверной группировкой RuBackup. Клиент RuBackup должен работать от имени суперпользователя (root для Linux и Unix).

Клиентский менеджер RuBackup PostgreSQL представляет собой оконное приложение, позволяющее администратору PostgreSQL организовывать резервное копирование и восстановления кластера баз данных PostgreSQL с возможностью восстановления СУБД до определённого момента во времени из ранее сохранённых резервных копий.

Для функционирования клиента RuBackup и в файле /etc/services должны присутствовать следующие записи (это может быть выполнено утилитой rb_init):

rubackup-cmd 9991/tcp

rubackup-lic 9992/tcp

rubackup-media 9993/tcp

Номера портов могут быть при необходимости переопределены администратором RuBackup.

Для установки клиента и клиентского менеджера RuBackup необходимо выполнить следующие действия:

1. Установить клиентский пакет в каталог /opt/rubackup
2. Настроить клиента RuBackup при помощи утилиты rb_init
3. Включить сервис rubackup_client

Более подробно установка клиента описана в «Руководстве по установке RuBackup».

Настройки переменных окружения (Linux, Unix, MacOS)

Путь доступа к исполняемым файлам RuBackup:

PATH=$PATH:/opt/rubackup/bin

Путь доступа к необходимым библиотекам:

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib

Главный конфигурационный файл

Основные настройки клиента RuBackup можно выполнить, редактируя главный конфигурационный файл, либо внося изменения в конфигурационном диалоге клиентского менеджера RuBackup.

После внесения изменений в конфигурационный файл требуется перезапуск клиента RuBackup, это можно выполнить следующим образом:

 # sudo systemctl stop rubackup_client

 # sudo systemctl start rubackup_client

Главный конфигурационный файл RuBackup располагается в каталоге:

/opt/rubackup/etc/config.file

Для клиента RuBackup значимыми настройками являются:

1. Местонахождение журнального файла RuBacku

logfile /opt/rubackup/log/RuBackup.log

2. Тип узла RuBackup

node client

3. IP адрес основного сервера RuBackup

who-is-primary-server 192.168.0.50

4. IP адрес резервного сервера RuBackup

who-is-secondary-server 192.168.0.52

5. Имя сетевого интерфейса, через который разрешена работа RuBackup на клиенте

client-inet-interface enp0s8

6. Локальный каталог, предназначенный для создания резервных копий и временного хранения файлов во время работы клиента RuBackup

use-local-backup-directory /tmp

7. Требуется ли подписывать резервные копии электронной подписью

digital-signature yes

Ключи защитного преобразования

В файле /opt/rubackup/keys/master-key хранится ключ для выполнения преобразования. Длина ключа преобразования составляет 256 бит.

В каталоге /opt/rubackup/keys/ могут располагаться другие файлы с ключами шифрования, для использования отдельных ключей в локальном расписании резервного копирования (только для файловых систем).

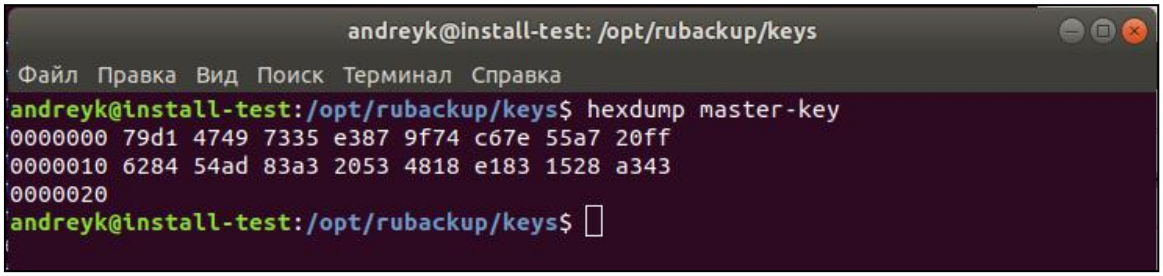
Каталог /opt/rubackup/keys/ невозможно скопировать клиентом RuBackup, так как он принудительно исключается из каждой задачи резервного копирования в целях предупреждения утечки ключей.

В том случае, если локальное или глобальное правило резервного копирования должно выполняться с защитным преобразованием, но необходимого ключа на клиенте не обнаружено, то задача резервного копирования будет выполнена без преобразования.

[Важно !!!] При утере ключа вы не сможете восстановить данные из резервной копии.

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

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



Подготовка сервера СУБД PostgreSQL к резервному копированию с помощью RuBackup

Если у вас нет сервера с СУБД PostgreSQL

В том случае, если PostgreSQL не установлен на вашем сервере, но вы хотите проверить как осуществляется резервное копирование СУБД PostgreSQL при помощи RuBackup, вам потребуется установить эту СУБД. Для проверки рекомендуется использовать ОС Linux Ubuntu. Установка СУБД осуществляется при помощи команды:

 # sudo apt-get install postgresql

После завершения установки кластер баз данных будет расположен в каталоге /var/lib/postgresql/10/main, о чем будет сообщено в конце инсталляции (в других дистрибутивах Linux месторасположение может отличаться).

Проверить работу postgresql можно при помощи команды:

 # sudo -iu postgres /usr/lib/postgresql/10/bin/pg_ctl status -D /etc/postgresql/10/main

Запустить psql можно при помощи следующей команды:

 # sudo -u postgres psql

Далее вы можете создать свою тестовую базу данных (инструкции об этом выходят за рамки данного руководства) и приступить к подготовке к резервному копированию при помощи RuBackup.

При необходимости удалить PostgreSQL вы можете воспользоваться командами:

*# apt-get --purge remove postgresql**

 # rm -r /etc/postgresql/

 # rm -r /etc/postgresql-common/

 # rm -r /var/lib/postgresql/

 # userdel -r postgres

 # groupdel postgres

Подготовка имеющегося сервера с СУБД PostgreSQL

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 быть другим в других дистрибутивах, проконсультируйтесь по этому поводу у вашего администратора PostgreSQL) :

wal_level = archive

archive_mode = on

archive_command = 'test ! -f /opt/rubackup/mnt/postgresql_archives/%f && cp %p /opt/rubackup/mnt/postgresql_archives/%f'

archive_timeout = 300

После внесения изменений необходимо перезапустить PostgreSQL.

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 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. Это значение может быть изменено при сборке PostgreSQL.

2. Необходимо оценить как часто создаётся 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 двухкратный 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 переключится на новый сегмент WAL.

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

 # sudo service postgresql restart

Создание пользователя СУБД для безопасного выполнения базовой резервной копии PostgreSQL

Пользователь PostgreSQL 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 qwerty1234:

 # create user rubackup_backuper password 'qwerty1234';

 # 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;

Вместо пользователя rubackup_backuper вы можете создать любого другого с соответствующим набором прав. В настройках RuBackup для каждой СУБД PostgreSQL можно будет указать правильное значение при

создании правила резервного копирования PostgreSQL. Учитывая что пароль этого пользователя хранится в базе данных RuBackup сервера в незашифрованном виде, нецелесообразно этому пользователю предоставлять какие-либо иные права с точки зрения безопасности. Это не означает, что пароль доступен любому желающему, но администратор серверной группировки RuBackup при желании может получить к нему доступ.

Принцип выполнения базового резервного копирования PostgreSQL

В ходе базового резервного копирования выполняются действия (sql запросы от имени пользователя rubackup_backuper, действия в операционной системе от имени пользователя, под которым работает клиентский процесс RuBackup), аналогичные следующим командам:

1. Старт резервного базового копирования PostgreSQL:

   postgres=# \c postgres rubackup_backuper

   postgres=> SELECT pg_start_backup('label', false, false);

2. Копирование файлов кластера баз данных:

   postgres@pg-server:~$ 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 PostgreSQL:

   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@pg-server:~$ 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

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 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 /etc/postgresql/10/main

2. Сделать резервную копию файлов в каталоге кластера баз данных, для возможности отката (в примере использован каталог *~/emergency_copy)*:

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

3. Очистить каталог кластера баз данных:

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

4. Восстановить данные из резервных копий. 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. Создать файл recovery.conf со следующим содержимым:

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

6. Запустить восстановление PostgreSQL:

 # sudo -iu postgres /usr/lib/postgresql/10/bin/pg_ctl start -D /etc/postgresql/10/main

Если вы установили параметр recovery_target_time в файле recovery.conf для восстановления базы данных на определённый 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 мошут 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 rbcryt поддерживает ключи длиной 128, 256, 512 и 1024 бита (в зависимости от выбранного алгоритма преобразования). Если необходимо для правила глобального расписания выбрать особый режим преобразование, 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 ещё раз автоматически с использованием мастер-ключа.

Для выполнения защитного преобразования доступны следующие алгоритмы:

Наименовние 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 менеджера администратора RuBackup для резервного копирования PostgreSQL

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.

Для резервного копирования серверов PostgreSQL на каждом из них должен быть установлен клиент RuBackup, и этот клиент должен быть авторизован администратором RuBackup (см.раздел “Клиенты” менеджера администратора RuBackup).

Архитектура клиента RuBackup позволяет проводить одновременно несколько операций резервного копирования и/или восстановления данных. Эта особенность может быть полезна, если на сервере работает более одного экземпляра PostgreSQL и для каждого из них должны быть предусмотрены свои особенности политики резервного копирования. Одновременные операции резервного копирования и восстановления не блокируют друг друга, однако необходимо быть осторожным в том плане, чтобы не выполнялось резервное копирование данных, восстановление которых происходит в тот же самый момент и подобные ситуации.

При помощи менеджера администратора RuBackup можно создать в глобальном расписании одно или несколько правил резервного копирования PostgreSQL для клиента. 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. Выполнить сжатие резервной копии на клиенте или на сервере после передачи ему резервной копии

6. Периодически выполнять проверку целостности резервной копии

7. Хранить резервные копии определяенный 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.

8. Через определённый срок после создания резервной копии автоматически переместить её на другой пул хранения резервных копий, например на картридж ленточной библиотеки.

9. 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 rbc_postgres.

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

Настройки правил глобального расписания RuBackup для резервного копирования PostgreSQL

Для выполнения резервного копирования PostgreSQL необходимо при помощи менеджера администратора Rubackup создать правило в глобальном расписании, в котором указать тип ресурса PostgreSQL, а сам ресурс должен указывать на каталог кластера баз данных экземпляра PostgreSQL (по умолчанию */var/lib/postgresql/10/main/).*

При выборе типа резервируемого ресурса PostgreSQL поле ресурса будет автоматически заполнено значением по умолчанию /var/lib/postgresql/10/main. В том случае, если у вас другая версия PostgreSQL, то значение 10 нужно поменять на требуемую версию; возможно потребуется изменить значение ресурса на тот каталог, где располагается кластер баз данных на сервере. Проконсультируйтесь по этому поводу у вашего администратора PostgreSQL.

В особых настройках необходимо указать:

Backuper Username - имя пользователя СУБД, у которого есть права на исполнение *pg_start_backup и pg_stop_backup (по умолчанию rubackup_backuper)*

Backuper Password - пароль пользователя

Archive WAL catalog - месторасположение каталога архивных WAL файлов для резервного копирования их при помощи RuBackup. Значение по умолчанию: /opt/rubackup/mnt/postgresql_archives

Auto remove WAL - удалять ли архивные WAL файлы, после удачной процедуры резервного копирования. Удаляются только те WAL файлы, которые были скопированы в резервную копию. Может оказаться так что, при неправильной настройке резервного копирования, в архивном каталоге

остаются ненужные WAL файлы и он постепенно заполняется. Такое, к примеру, может случиться, если выполнять только базовое резервное копирование PostgreSQL.

Правило резервного копирования может быть полным (full) и инкрементальным (incremental) для PostgreSQL. При полном резервном копировании выполняется базовое резервное копирование файлов базы данных и WAL файлов, созданных за время процедуры их резервного копирования. Инкрементальное копирование резервирует архивные WAL файлы, созданные после завершения последней базовой резервной копии PoastgreSQL. В том случае, если инкрементальный бэкап начался слишком быстро после окончания последней операции резервного копирования и в архивном каталоге не появилось ещё ни одного нового WAL файла, то задача будет завершена с ошибкой.

Настройки правила RuBackup для резервного копирования конфигурационных файлов PostgreSQL

Базовые настройки PostgreSQL располагаются в нескольких файлах файловой системы хоста. Изменение этих файлов производится системным администратором при помощи того или иного текстового редактора.

Для выполнения резервного копирования этих файлов необходимо создать правило в глобальном расписании RuBackup для ресурса типа “File system”. В качестве самого ресурса требуется узакать директорию, в которой располагаются конфигурационные файлы PostgreSQL. Обычно это каталог */etc/postgresql/10/main/*, но в зависимости от вашего дистрибутива конфигурационные файлы могут располагаться в другом месте. Проконсультируйтесь по этому поводу у вашего администратора PostgreSQL.

Для резервного копирования конфигурационных файлов со стороны клиента необходимо использовать клиентский менеджер RuBackup для файловых систем (rbc_filesystem).

Использование клиентского менеджера RuBackup

Принцип взаимодействия клиентского менеджера с системой резервного копирования состоит в том, что пользователь может сформировать ту или иную команду (желаемое действие) и отправить его 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 её серверу. В том случае, если действие допустимо, то сервер RuBackup отдаст обратную команду клиенту и/или перенаправит её медиа-серверу 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.

При первом запуске клиентского менеджера необходимо задать пароль, при помощи которого впоследствии можно будет запросить восстановление резервной копии. Без ввода пароля получить резервную копию для клиента из хранилища невозможно. 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 закладки, позволяющие управлять резервными копиями, расписанием резервного копирования и просматривать текущие задачи клиента (для типа ресурса PostgreSQL). Внизу страницы располагаются кнопки, позволяющие выполнять те или иные действия резервного копирования или восстановления.

Закладка “Резервные копии”

В таблице закладки “Резервные копии” содержится информация обо всех резервных копиях клиента (тип ресурса PostgreSQL), 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.

В закладке “Резервные копии” пользователю доступны следующие действия:

◆ Срочное резервное копирование

Выполнение базовой (полной) резервной копии. Это действие (в случае успешного завершения резервного копирования) прервет предыдущею цепочку инкрементальных резервных копий. Новая инкрементальная резервная копия будет ссылаться на вновь созданную полную резервную копию. Из клиентского менеджера возможно выполнение только базовой резервной копии. Для инкрементального резервного копирования требуется создать правило в глобальном расписании.

◆ Удалить архив

Это действие отправляет запрос серверу резервного копирования RuBackup запрос на удаление выбранной резервной копии. Запрос не

проходит модерации администратором RuBackup, а отправляется на исполнение тому медиа-серверу, на хранилищах которого располагается резервная копия.

◆ Восстановить цепочку архивов

Это действие запускает процесс восстановления цепочки резервных копий на локальной файловой системе клиента. От пользователя потребуется задать каталог, в который требуется восстановить резервные копии. Клиентские 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.

◆ Восстановить СУБД PostgreSQL

Это действие запускает процесс восстановления СУБД PostgreSQL из ранее восстановленных резервных копий на файловой системе клиента.

Восставноление цепочки архивов специально отделено от восстановления СУБД, так как последний процесс необходимо проводить под контролем администратора восстанавливаемой СУБД.

Собственно процедура восстановления СУБД представлена ниже в разделе “Восстановление СУБД PostgreSQL с помощью клиентского менеджера RuBackup”.

◆ Проверить резервную копию

Это действие инициирует создание задачи проверки резервной копии. В том случае, если резервная копия была подписана цифровой подписью, то будет 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.

Закладка “Расписание”

В таблице закладки “Расписание” содержится информация обо всех правилах в глобальном расписании RuBackup для этого клиента (тип ресурса PostgreSQL).

В закладке “Расписание” пользователю доступны следующие действия:

◆ 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.

◆ Запросить удалить правило

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.

Закладка “Задачи”

В таблице закладки “Задачи” содержится информация обо всех задачах в главной очереди заданий RuBackup для этого клиента (тип ресурса PostgreSQL). 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 удалённой 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”).

Восстановление СУБД PostgreSQL с помощью клиентского менеджера RuBackup

Диалог восстановления СУБД PostgreSQL можно запустить только после восстановления цепочки архивов резервных копий.

Восстановление СУБД PostgreSQL требует выполнения следующих последовательных действий:

1. Останов сервиса PostgreSQL.
2. Резервное копирование существующих файлов баз данных в локальный каталог для возможности отката.
3. Очистка каталога кластера баз данных и каталогов табличных пространств.
4. Копирование файлов, восстановленных из резервных копий, в каталог кластера баз данных и в каталоги табличных пространств.
5. Запуск восстановления PostgreSQL,
6. Удаление файлов, скопированных в п.2.

Утилиты командной строки клиента RuBackup

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

rb_archive_pg

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

rb_schedule_pg

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

rb_tasks_pg

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

Ознакомиться с функциями утилит командной строки можно при помощи команды man или в руководстве “Утилиты командной строки RuBackup”.

Лицензирование

Система резервного копирования RuBackup имеет следующие типы лицензий:

1. Простая бесплатная лицензия

Эта лицензия включает в себя возможность выполнять резервное копирование одного клиента, на котором функционирует PostgreSQL. При необходимости выполнять резервное копирования для большего числа клиентов, необходимо приобретение коммерческой лицензии. Для простой лицензии нет возможности использовать резервный сервер RuBackup и добавлять в конфигурацию дополнительные медиа-серверы.

2. Коммерческая лицензия

Эта лицензия включает в себя возможность выполнять резервное копирование трёх клиентов, на которых функционирует PostgreSQL. При необходимости выполнять резервное копирование для большего числа клиентов необходимо расширение коммерческой лицензии. Возможно расширение серверной группировки RuBackup с помощью медиа серверов, построение отказоустойчивой конфигурации путём добавления резервного сервера. Все серверы серверной группировки должны иметь собственные лицензии, одинаковые с точки зрения количества клиентов.

3. Пробная лицензия

Эта лицензия включает в себя возможность проверить функционал системы резервного копирования, выполнять ограниченное время резервное копирование одного клиента, на котором функционирует PostgreSQL. По окончании действия пробной лицензии функционирование системы приостанавливается.

Более подробно о лицензировании RuBackup читайте в соответствующем руководстве.