

RuBackup

Система резервного копирования и восстановления данных

Резервное копирование и восстановление отдельных баз данных и таблиц Postgres



RuBackup

Версия 1.9

2022 г.

Содержание

Введение.....	3
Подготовка хоста СУБД Postgres.....	4
Подготовка СУБД Postgres.....	6
Принцип базового резервного копирования при помощи pg_dump.....	10
Восстановление резервной копии.....	11
Мастер-ключ.....	12
Защитное преобразование резервных копий.....	13
Менеджер Администратора RuBackup (RBM).....	15
Менеджер Клиента RuBackup (RBC).....	23
Утилиты командной строки клиента RuBackup.....	28
Восстановление резервной копии отдельной БД или таблицы.....	30

Введение

Система резервного копирования (СРК) RuBackup позволяет осуществлять резервное копирование и восстановление таблиц и баз данных СУБД PostgreSQL. Модули резервного копирования поддерживают СУБД PostgreSQL версий 9.6, 10, 11, 12, 13, 14.

Принцип резервного копирования отдельной БД или таблицы с использованием RuBackup состоит в периодическом создании базовых резервных копий экземпляра СУБД по определённому расписанию.

В репозитории RuBackup базовые резервные копии будут храниться как полные резервные копии (*full*). Файл с расширением «.snap», если производится резервное копирование отдельной БД, содержит в себе наименование скопированной БД и имена всех таблиц, которые были внутри неё. При резервном копировании одной таблицы файл с расширением «.snap» содержит в себе наименование БД, к которой принадлежит эта таблица и имя самой скопированной таблицы. Внутри архива с резервной копией также будет находиться копия snap-файла. Модули поддерживают только полное резервное копирование.

После окончания операции резервного копирования будут созданы два файла - архивный и snap-файл на медиасervere, которому принадлежит пул, указанный в правиле резервного копирования. Точное расположение файлов указано в записи репозитория системы резервного копирования RuBackup.

При необходимости архивный файл может быть преобразован при помощи алгоритма защитного преобразования на клиенте и сжат. Снимок состояния не преобразовывается. В снимке состояния отсутствуют значимые данные СУБД.

Для выполнения резервного копирования на хосте клиента должно быть достаточно свободного места для создания резервной копии. 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 системного администратора RuBackup»).

Для выполнения резервного копирования администратор RuBackup может настраивать правила глобального расписания в оконном Менеджере Администратора RuBackup (RBM).

Клиенты RuBackup могут осуществлять восстановление данных резервных копий и создание срочных резервных копий при помощи оконного Менеджера Клиента RuBackup (RBC), а также при помощи утилит командной строки RuBackup.

Подготовка хоста СУБД Postgres

Для возможности резервного копирования отдельной БД или таблицы СУБД Postgres при помощи СРК RuBackup на сервер с СУБД следует установить следующие пакеты:

- 1) `gubackup-client.deb` – клиент резервного копирования;
- 2) `gubackup-pg-dump.deb` – модуль резервного копирования;
- 3) `python3` – интерпретатор Python (версии 3.6 и выше);
- 4) `python3-psycopg2` – драйвер PostgreSQL, совместимый с DB API 2.0.

Установка клиента RuBackup

Для осуществления резервного копирования и восстановления данных СУБД Postgres при помощи RuBackup на сервер должен быть установлен клиент RuBackup со всеми необходимыми модулями. Клиент RuBackup представляет собой фоновое системное приложение (демон или сервис), обеспечивающее взаимодействие с серверной группировкой RuBackup.

Для выполнения резервного копирования ресурсов СУБД Postgres клиент RuBackup должен работать от имени суперпользователя (`root` в Linux и Unix).

Подробно процедуру установки клиента RuBackup см. «Руководство по установке серверов резервного копирования и Linux клиентов RuBackup».

Установка пакета модулей резервного копирования

Установка пакета модулей резервного копирования RuBackup производится из учётной записи с административными правами на узле СУБД Postgres **после** установки на него клиента RuBackup.

Для установки пакета модулей используйте следующий вызов:

```
$ sudo dpkg -i gubackup-pg-dump.deb
```

```
Выбор ранее не выбранного пакета gubackup-pg-dump.
```

```
(Чтение базы данных ... на данный момент установлено 137334 файла и каталога.)
```

```
Подготовка к распаковке gubackup-pg-dump.deb ...
```

```
Распаковывается gubackup-pg-dump (2021-02-20) ...
```

Настраивается пакет `rubackup-pg-dump` (2020-02-20) ...

Удаление клиента RuBackup

При необходимости вы можете удалить с сервера клиент RuBackup и установленные модули резервного копирования.

Удаление клиента RuBackup возможно из учётной записи с административными правами.

Для удаления сервиса `rubackup-client` используйте команды:

```
# systemctl disable rubackup-client  
# systemctl daemon-reload
```

Для удаления клиента RuBackup и модуля `rubackup-pg-dump` используйте команды:

```
# apt remove rubackup-pg-dump  
# apt remove rubackup-client
```

При необходимости удалить клиент RuBackup из конфигурации системы резервного копирования, это может сделать системный администратор RuBackup с помощью оконного Менеджера Администратора (RBM).

После удаления клиента RuBackup в ОС Astra Linux SE 1.6 с активированным режимом защитной программной среды следует:

1. Выполнить команду:

```
$ sudo update-initramfs -u -k all
```

2. Перезагрузить операционную систему

```
$ init 6
```

Подготовка СУБД Postgres

Для подготовки СУБД Postgres к выполнению резервного копирования при помощи СРК RuBackup необходимо выполнить следующие действия:

1. Изменить конфигурационный файл `pg_hba.conf`. Чтобы найти этот файл используйте вызовы:

```
$ sudo -u postgres psql
$ psql -c 'show hba_file'
```

В строке `local` установить методы `peer` и `md5`:

```
# TYPE  DATABASE  USER  ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local   all       all                    peer
local   all       all                    md5
```

2. Перечитать конфигурацию:

```
# psql -c 'select pg_reload_conf()'
```

3. Проверить на наличие опечаток:

```
# psql -c 'select * from pg_hba_file_rules'
```

4. После внесения изменений перезапустить сервис Postgres командой:

```
$ sudo service postgresql restart
```

Создание пользователя СУБД

Для выполнения резервной копии баз данных и таблиц PostgreSQL при помощи RuBackup в СУБД необходимо создать пользователя.

Пользователь СУБД должен обладать правами на выполнение команд SELECT и CREATE. Для настройки необходимо выполнить следующие действия:

1. Подключить к базе данных от имени пользователя postgres:

```
$ sudo -u postgres psql
```

2. Создать пользователя *rubackup_backuper* (в качестве пароля укажите желаемый пароль вместо '12345'):

```
# CREATE USER rubackup_backuper WITH LOGIN password '12345';
```

Вместо пользователя *rubackup_backuper* вы можете создать пользователя с другим именем.

3. Запустить консольную утилиту *rb_pg_dump_script*, при помощи которой можно выдать права ранее созданному пользователю на выполнение 'SELECT' и 'CREATE' в целевой БД (или в нескольких):

```
$ /opt/rubackup/scripts/rb_pg_dump_script -1 -U admin_name -p  
admin_password -B backup_user -H db_host -P db_port -S schema
```

Параметры команды:

-1 – выдать права ранее созданному пользователю на выполнение команд SELECT и CREATE в целевой БД; параметр -2 снимает эти права;

-U *admin_name* – имя администратора кластера Postgres;

-p *admin_password* – пароль администратора кластера;

-B *backup_user* – имя пользователя RuBackup;

-H *db_host* – хост сервера базы данных или каталог сокетов;

-P *db_port* – порт базы данных;

-S *schema* – схема БД, к которой принадлежат копируемые таблицы.

4. При необходимости лишения ранее выданных прав у пользователя RuBackup, запустить утилиту с опцией -2:

```
# /opt/rubackup/scripts/rb_pg_dump_script -2 -U admin_name -p  
admin_password -B backup_user -H db_host -P db_port -S schema
```

5. Заполнить конфигурационные файлы актуальной информацией.

Для резервного копирования отдельной БД заполните файл */opt/rubackup/etc/rb_module_pg_dump_database.conf* следующим образом:

```
backup_user:rubbackup_backuper  
password:12345  
main_db:postgres  
host:localhost  
port:5432  
pg_dump:/user/bin/pg_dump  
pg_restore:/usr/bin/pg_restore  
direct_restore:yes
```

Параметры конфигурации:

`backup_user` – имя ранее созданного пользователя RuBackup;

`password` – пароль пользователя RuBackup;

`main_db` – база данных администратора Postgres;

`host` – хост сервера базы данных или каталог сокетов;

`port` – порт базы данных;

`pg_dump` – полный путь к утилите `pg_dump`;

`pg_restore` – полный путь к утилите `pg_restore`;

`direct_restore` – выполнить ли восстановление непосредственно в СУБД (yes) или распаковать архив в директорию и завершить операцию восстановления (no).

6. Для резервного копирования отдельной таблицы заполните файл `/opt/rubackup/etc/rb_module_pg_dump_table.conf` следующим образом:

```
backup_user:rubbackup_backuper  
password:12345  
target_db:my_database  
host:localhost  
port:5432  
pg_dump:/user/bin/pg_dump  
pg_restore:/usr/bin/pg_restore  
direct_restore:yes
```

Параметры конфигурации:

`backup_user` – имя ранее созданного пользователя RuBackup;

`password` – пароль пользователя RuBackup;

`target_db` – база данных, содержащая копируемую таблицу;

`host` – хост сервера базы данных или каталог сокетов;

`port` – порт базы данных;

`pg_dump` – полный путь к утилите `pg_dump`;

`pg_restore` – полный путь к утилите `pg_restore`;

`direct_restore` – выполнить ли восстановление непосредственно в СУБД (`yes`) или распаковать архив в директорию и завершить операцию восстановления (`no`).

Внимание! Во избежание раскрытия пароля пользователя RuBackup для конфигурационных файлов рекомендуется определить строгие права только для суперпользователя (`chmod 0600`).

Для ОС Astra Linux в файле `«/etc/parsec/mswitch.conf»` установить значение `zero_if_not_found: yes`.

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

```
# rubackup_client stop
```

```
# rubackup_client start
```

В результате клиент должен сообщить о том, что модуль `pg_dump` готов к работе.

Принцип базового резервного копирования при помощи pg_dump

В ходе базового резервного копирования выполняется взаимодействие с утилитой pg_dump.

Команда на выполнение полного резервного копирования отдельной базы данных:

```
# /usr/bin/pg_dump --username=rubackup_backuper --dbname=my_db --host=localhost --port=5432 -F c > backup.dump
```

Параметры команды:

- username – ранее созданный пользователь RuBackup;
- dbname – имя копируемой базы данных;
- host – хост сервера базы данных или каталог сокетов;
- port – порт базы данных;
- F – формат дампа (используется custom).

Команда для создания резервной копии отдельной таблицы:

```
# /usr/bin/pg_dump --username=rubackup_backuper --dbname=my_db --table=my_table --host=localhost --port=5432 -F c > backup.dump
```

Параметры команды:

- dbname – база данных, которая содержит в себе копируемую таблицу;
- table – имя копируемой таблицы.

Восстановление резервной копии

Данный метод может быть использован при ручном восстановлении отдельной БД или таблицы.

Для восстановления резервной копии отдельной базы данных или таблицы PostgreSQL используется утилита `pg_restore`.

Чтобы получить файл резервной копии в конфигурационном файле модуля (для модуля резервного копирования отдельной БД – `rb_module_pg_dump_database.conf` или для модуля резервного копирования таблицы - `rb_module_pg_dump_table.conf`) установите значение параметра `direct_restore` в `no`. Файл резервной копии будет создан в указанном каталоге.

Выполните команду:

```
# /usr/bin/pg_restore --username=rubackup_backuper --dbname=my_db  
--host=localhost --port=5432 backup.dump
```

Внимание! Если в целевой БД уже есть таблица с тем же именем что и восстанавливаемая, то восстановление завершится ошибкой. Рекомендуется либо удалить таблицу, не позволяющую выполнить восстановление, либо выполнить восстановление в другую базу данных.

Мастер-ключ

В ходе установки клиента RuBackup будет создан мастер-ключ для защитного преобразования резервных копий, а также ключи для электронной подписи, если предполагается использовать электронную подпись.

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

Важно! Ключи рекомендуется после создания скопировать на внешний носитель, а также 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:

```
$ hexdump /opt/rubackup/keys/master-key
00000000 79d1 4749 7335 e387 9f74 c67e 55a7 20ff
00000010 6284 54as 83a3 2053 4818 e183 1528 a343
00000020
```

Защитное преобразование резервных КОПИЙ

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

Защитное преобразование осуществляется входящей в состав RuBackup утилитой `gbscrypt`. Ключ для защитного преобразования резервных копий располагается на хосте клиента в файле `/opt/rubackup/keys/master-key`. Защитное преобразование данных при помощи `gbscrypt` возможно с длиной ключа 256 бит (по умолчанию), а также 128, 512 или 1024 бита в зависимости от выбранного алгоритма преобразования.

Если для правила глобального расписания необходимо выбрать особый режим защитного преобразования с длиной ключа, отличной от 256 бит, и с ключом, расположенным в другом месте, то вы можете 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. Провести обратное преобразование такого файла после восстановления его из архива следует вручную при помощи утилиты `gbscrypt`. При таком режиме работы нет необходимости указывать алгоритм преобразования в правиле резервного копирования, иначе архив будет повторно преобразован с использованием мастер-ключа.

Алгоритмы защитного преобразования

Для выполнения защитного преобразования доступны алгоритмы, представленные в таблице 1.

Таблица 1 – Алгоритмы защитного преобразования, доступные в утилите гбсCRYPT.

Наименование алгоритма	Поддерживаемая гбсCRYPT 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	

Менеджер Администратора RuBackup

(RBM)

Оконное приложение Менеджер Администратора RuBackup (RBM) предназначено для администрирования серверной группировки RuBackup, включая управление клиентами, глобальным расписанием, хранилищами резервных копий и другими параметрами RuBackup. Системный администратор RuBackup может запустить RBM на основном сервере резервного копирования RuBackup.

Для запуска RBM следует выполнить команду:

```
# ssh -X user@rubackup_server  
# /opt/rubackup/bin/rbm&
```

Пользователь, запускающий RBM, должен входить в группу rubackup.

Для резервного копирования отдельной БД или таблицы PostgreSQL на хосте должен быть установлен клиент RuBackup и необходимые модули. Клиент должен быть авторизован администратором RuBackup.

Если клиент RuBackup установлен, но не авторизован, в нижней части окна RBM появится сообщение о том, что найдены неавторизованные клиенты (рисунок 1). Все новые клиенты должны быть авторизованы в системе резервного копирования RuBackup.

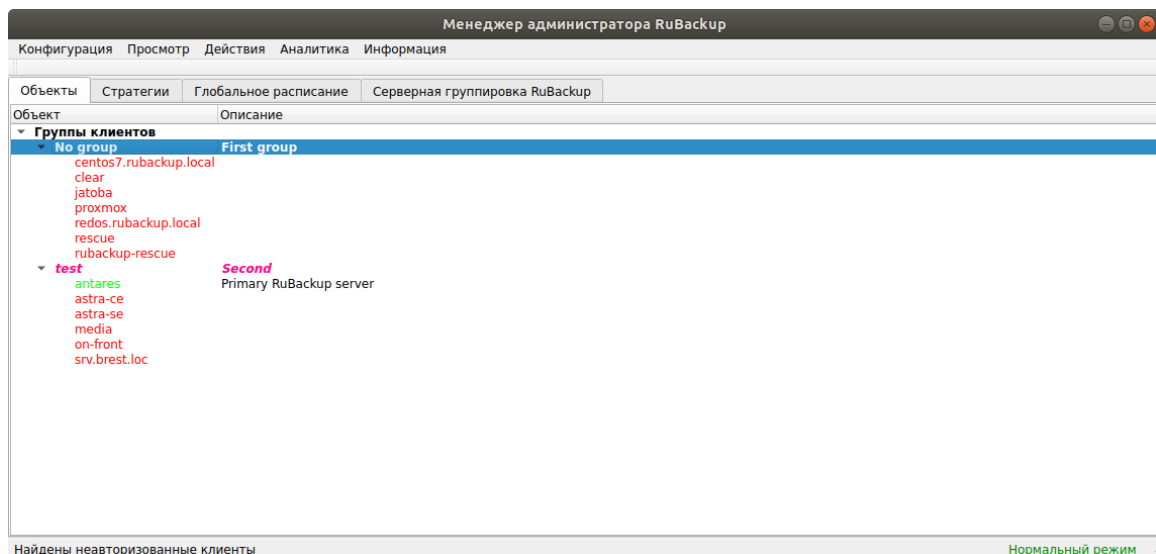


Рисунок 1

Для авторизации неавторизованного клиента в RBM выполните следующие действия:

1. Откройте меню **Действия** → **Клиенты** → **Авторизовать клиентов** (рисунок 2).

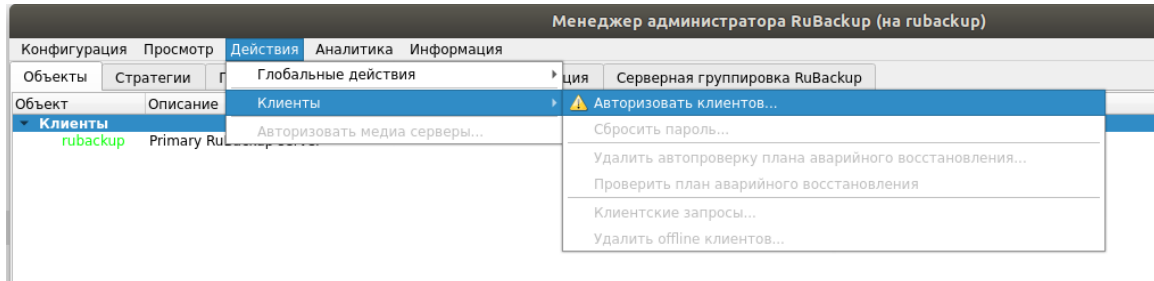


Рисунок 2

2. Выберите нужного неавторизованного клиента и нажмите **Авторизовать** (рисунок 3).

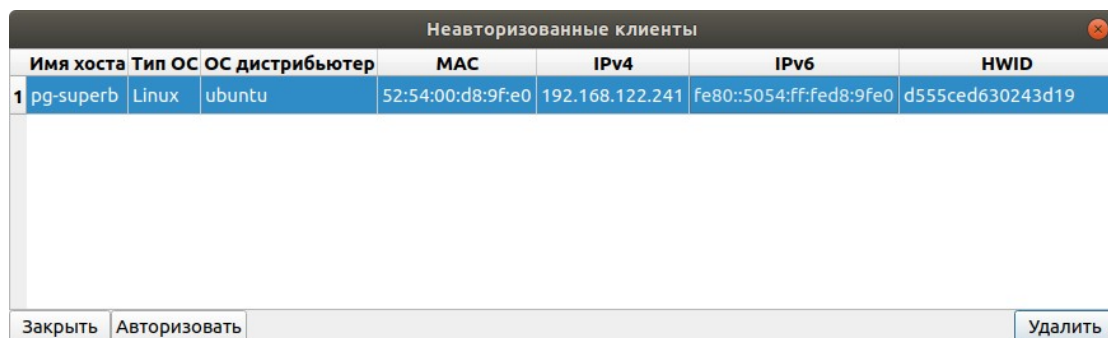


Рисунок 3

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

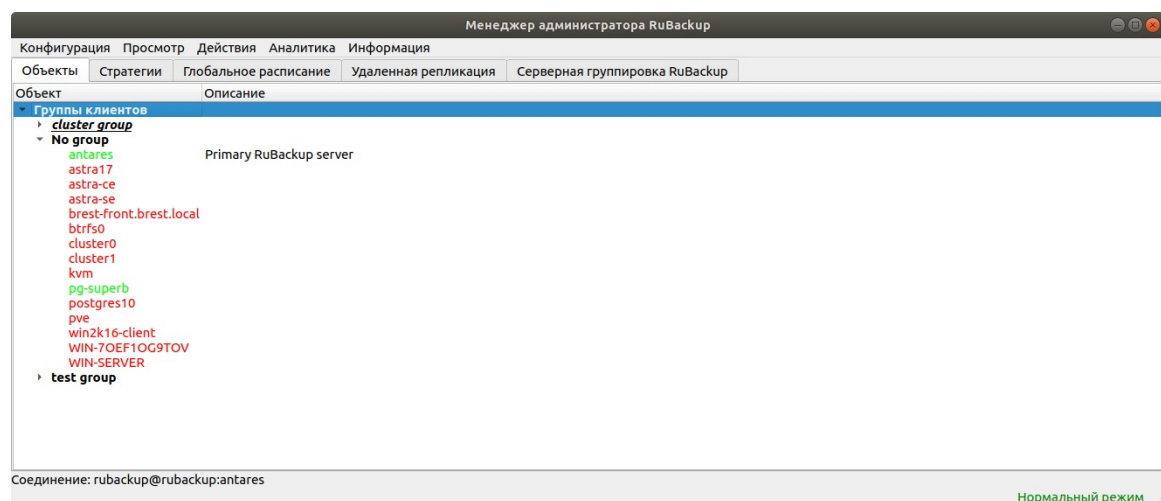


Рисунок 4

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

Чтобы выполнять регулярное резервное копирование отдельной БД или таблицы, необходимо создать правило в глобальном расписании. Для этого выполните следующие действия:

1. Выберите хост клиента, на котором находится СУБД PostgreSQL, и добавьте правило резервного копирования (рисунок 5):

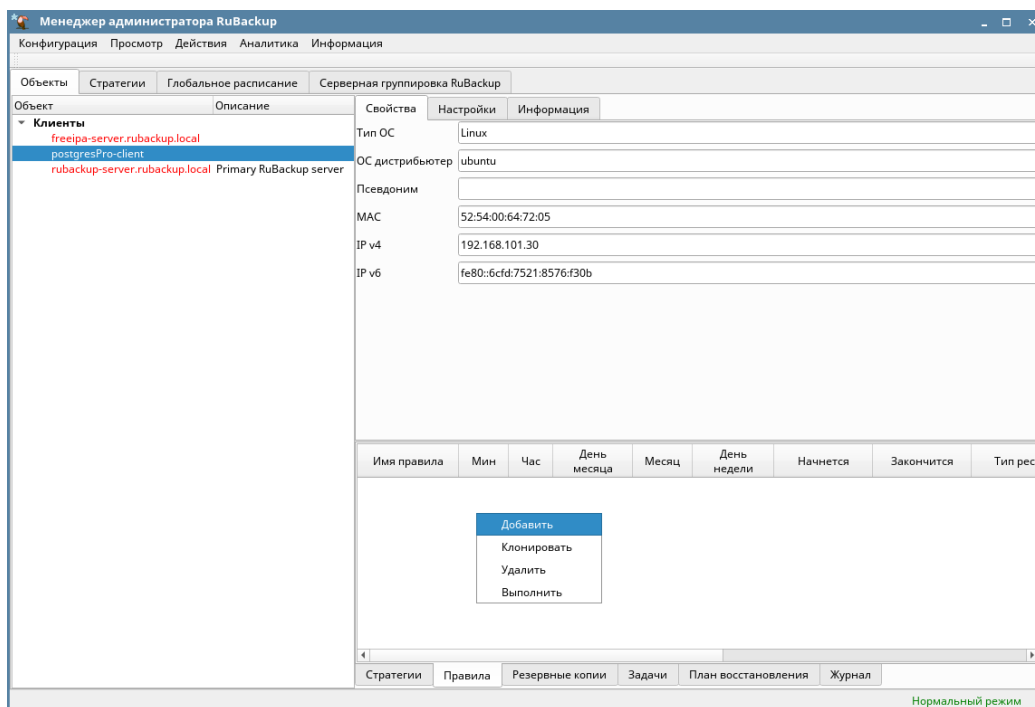


Рисунок 5

2. Выберите тип ресурса: «**PG_dump database**» (для резервного копирования отдельной БД) или «**PG_dump table**» (для резервного копирования таблицы) (рисунок 6):

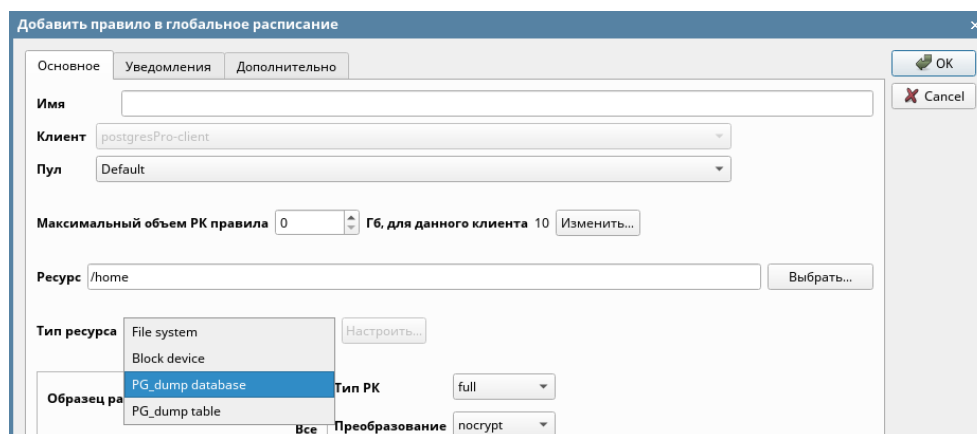


Рисунок 6

3. Выберите ресурс, нажав кнопку **Выбрать**.

Для типа ресурса: «**PG_dump database**» выберите целевую БД (рисунок 7).

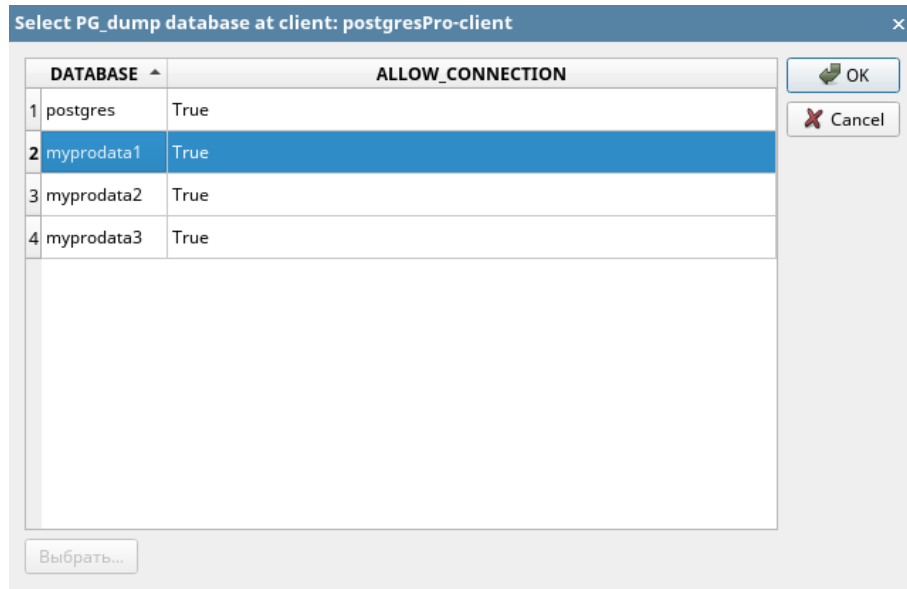


Рисунок 7

Для типа ресурса «**PG_dump table**» выберите целевую таблицу (рисунок 8):

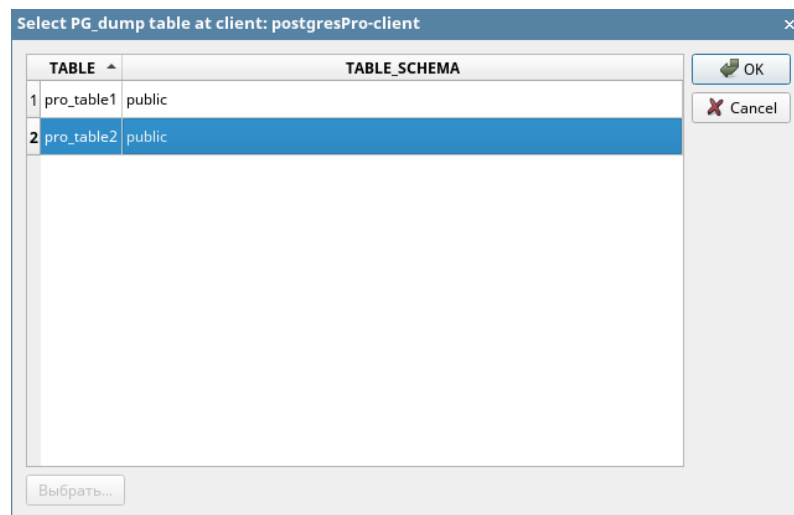
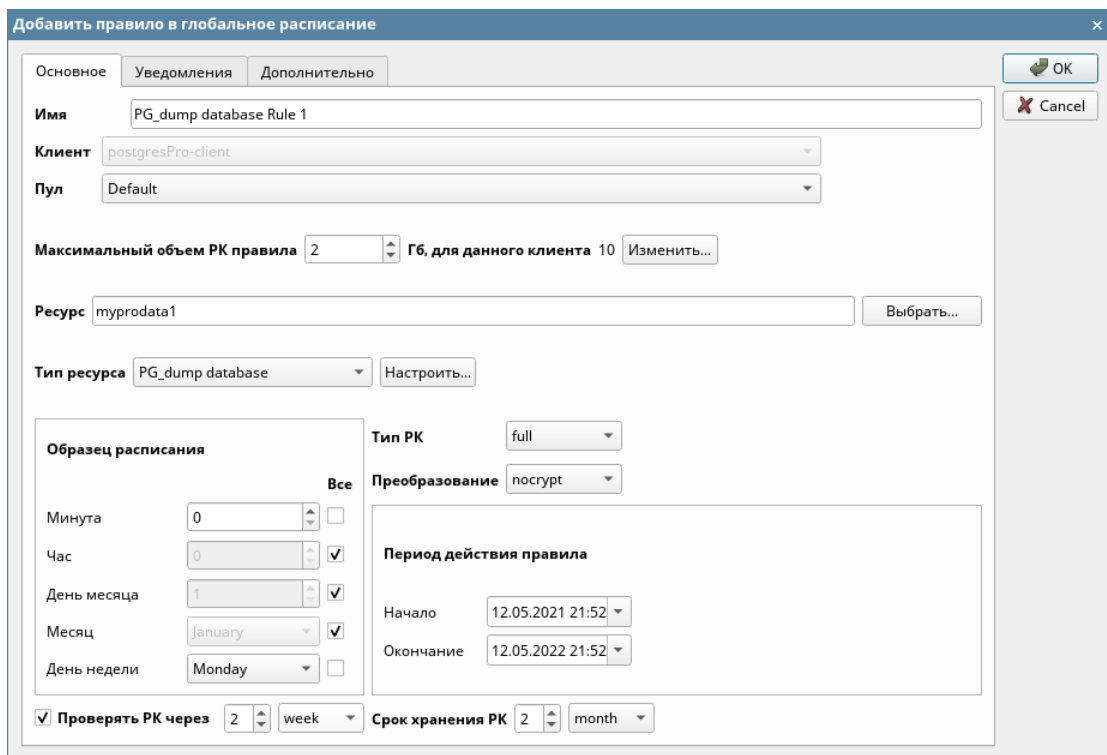


Рисунок 8

Внимание! Для типа ресурса «**PG_dump table**» выбор таблицы происходит из таблиц в базе данных, указанной в конфигурационном файле «**rb_module_pg_dump_table.conf**» в поле «**target_db**».

4. Установите настройки правила: название правила, пул хранения данных, максимальный объем для резервных копий правила (в ГБ), тип резервного копирования, расписание резервного копирования, срок хранения и

необязательный временной промежуток проверки резервной копии (рисунок 9).



Добавить правило в глобальное расписание

Основное | Уведомления | Дополнительно

Имя: PG_dump database Rule 1

Клиент: postgresPro-client

Пул: Default

Максимальный объем РК правила: 2 Гб, для данного клиента 10 Изменить...

Ресурс: myprodata1

Тип ресурса: PG_dump database

Образец расписания:

- Минута: 0
- Час: 0
- День месяца: 1
- Месяц: January
- День недели: Monday

Тип РК: full

Преобразование: nocrypt

Период действия правила:

- Начало: 12.05.2021 21:52
- Окончание: 12.05.2022 21:52

Проверять РК через 2 week

Срок хранения РК: 2 month

Рисунок 9

5. В системе предусмотрена возможность тонкой настройки правила резервного копирования, для их редактирования нажмите кнопку **Настроить** (рисунок 10):

	Параметр	Тип данных	Значение
1	threads	integer	1
2	dump_format	integer	2
3	serializable_deferrable	boolean	true
4	no_owner	boolean	false
5	no_sync	boolean	false
6	data_only	boolean	false
7	schema_only	boolean	false
8	blobs	boolean	false
9	no_blobs	boolean	false
10	no_publications	boolean	false
11	no_security_labels	boolean	false
12	no_subscriptions	boolean	false
13	no_synchronized_snapshots	boolean	false
14	no_tablespace	boolean	false
15	no_unlogged_table_data	boolean	false

Рисунок 10

Дополнительные параметры ресурса представлены в таблице 2.

Таблица 2 – Дополнительные параметры ресурса

Параметр	Описание	Значение по умолчанию	Допустимые значения
threads	Количество потоков выполнения резервной копии	1	Положительные целые числа
serializable_deferrable	Дождаться момента для выгрузки данных без аномалий	true	true, false
no_owner	Не восстанавливать владение объектами	false	true, false
no_sync	Не ждать надёжного сохранения изменений на диске	false	true, false
data_only	Выгрузить только данные, без схемы	false	true, false
schema_only	Выгрузить только схему, без данных	false	true, false
blobs	Выгрузить также большие объекты	false	true, false
no_blobs	Исключить из выгрузки большие объекты	false	true, false
no_publications	Не выгружать публикации	false	true, false
no_security_labels	Не выгружать назначения меток безопасности	false	true, false
no_subscriptions	Не выгружать подписки	false	true, false
no_synchronized_snapshots	Не использовать синхронизированные снимки в параллельных заданиях	false	true, false
no_tablespaces	Не выгружать назначения табличных пространств	false	true, false
no_unlogged_table_data	Не выгружать данные нежурналируемых таблиц	false	true, false
-F, --format=c d t p	Формат выводимых данных	текстовый	пользовательский каталог tar текстовый

6. На вкладке «Дополнительно» можно настроить автоматическое удаление устаревших резервных копий, определить условие их перемещения в другой пул и установить разрешение для клиента удалять резервные копии (рисунок 11).

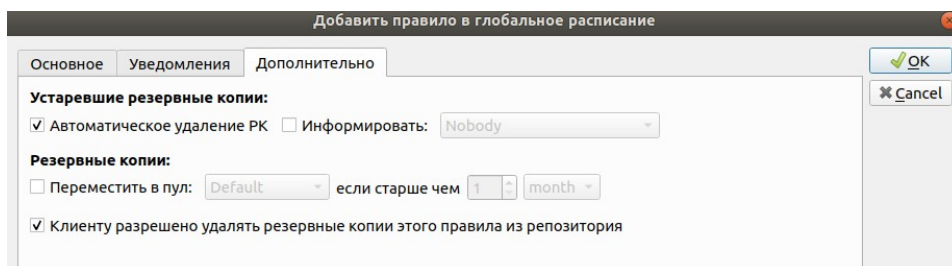


Рисунок 11

Вновь созданное правило будет иметь статус `wait`. Это означает, что оно не будет порождать задач на выполнение резервного копирования, пока администратор RuBackup не запустит его (тогда его статус сменится на `run`). При необходимости, администратор может приостановить работу правила или немедленно запустить его (т. е. инициировать немедленное создание задачи при статусе правила `wait`).

Правила глобального расписания имеют срок жизни, определяемый при их создании, а также предоставляют следующие возможности:

- выполнить скрипт на клиенте перед началом резервного копирования;
- выполнить скрипт на клиенте после успешного окончания резервного копирования;
- выполнить скрипт на клиенте после неудачного завершения резервного копирования;
- выполнить защитное преобразование резервной копии на клиенте;
- периодически выполнять проверку целостности резервной копии;
- хранить резервные копии определённый срок, по окончании которого удалять их из хранилища резервных копий и из записей репозитория, либо уведомлять клиента об окончании срока хранения;
- через определённый срок после создания резервной копии автоматически переместить её в другой пул хранения резервных копий, например, на картридж ленточной библиотеки.

– 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 выполнение правил может как администратор (при помощи RBM или утилит командной строки), так и клиент (при помощи RBC или утилиты командной строки `gb_tasks`).

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

Менеджер Клиента RuBackup (RBC)

Принцип взаимодействия Менеджера Клиента RuBackup (RBC) с системой резервного копирования состоит в том, что клиент может сформировать ту или иную задачу (желаемое действие) и отправить её серверу резервного копирования RuBackup. Взаимодействие клиента с сервером резервного копирования производится через клиента RuBackup (фоновый процесс). RBC отправляет команду клиенту RuBackup, который отправляет её серверу. Если действие допустимо, то сервер RuBackup отдаст команду клиенту RuBackup и, при необходимости, перенаправит её медиасерверу RuBackup для дальнейшей обработки. Это означает, что, как правило, RBC не ожидает завершения того или иного действия, но ожидает ответа от клиента RuBackup, что задание принято. Это позволяет инициировать параллельные запросы процесса клиента RuBackup к серверу, но требует от клиента самостоятельно контролировать отсутствие «встречных» операций, при которых происходит восстановление данных, и в этот же момент эти же данные требуются для создания новой резервной копии. После того, как клиент отдал какую-либо команду при помощи RBC, он может просто закрыть приложение, все действия будут выполнены системой резервного копирования (тем не менее, стоит дождаться сообщения о том, что задание принято к исполнению, и проконтролировать это на вкладке «Задачи»).

Графический интерфейс RBC поддерживает русский и английский языки.

Для запуска RBC следует выполнить команды:

```
# ssh -X user@postgresql-host  
# /opt/rubackup/bin/rbc&
```

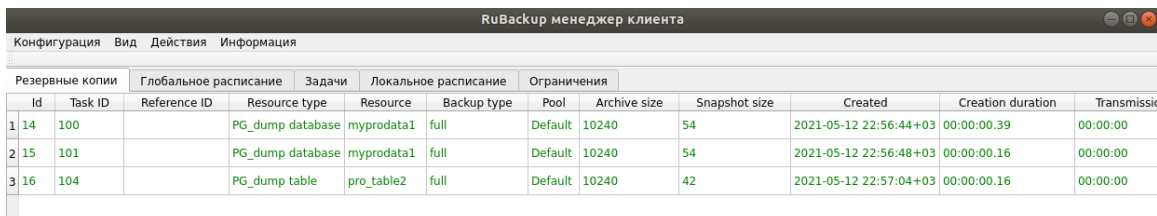
Пользователь, запускающий RBC, должен входить в группу rubackup.

При первом запуске RBC необходимо задать пароль, при помощи которого впоследствии можно будет запросить восстановление резервной копии. Без ввода пароля получить резервную копию для клиента из хранилища невозможно. Хеш пароля восстановления хранится в базе данных сервера RuBackup. При необходимости клиент может изменить пароль при помощи RBC (меню **Конфигурация** → **Изменить пароль**).

Главная страница RBC содержит вкладки, которые позволяют управлять резервными копиями и расписанием резервного копирования, а также просматривать текущие задачи клиента, локальное расписание и ограничения.

Вкладка «Резервные копии»

Вкладка «Резервные копии» содержит таблицу с информацией обо всех резервных копиях клиента, которые хранятся в репозитории RuBackup (рисунок 12):



RuBackup менеджер клиента											
Конфигурация Вид Действия Информация											
Резервные копии		Глобальное расписание		Задачи		Локальное расписание		Ограничения			
Id	Task ID	Reference ID	Resource type	Resource	Backup type	Pool	Archive size	Snapshot size	Created	Creation duration	Transmissi
1	14	100	PG_dump database	myprodata1	full	Default	10240	54	2021-05-12 22:56:44+03	00:00:00.39	00:00:00
2	15	101	PG_dump database	myprodata1	full	Default	10240	54	2021-05-12 22:56:48+03	00:00:00.16	00:00:00
3	16	104	PG_dump table	pro_table2	full	Default	10240	42	2021-05-12 22:57:04+03	00:00:00.16	00:00:00

Рисунок 12

На этой вкладке клиенту доступны следующие действия:

- **Удалить выбранную резервную копию.** Это действие возможно в том случае, если в правиле глобального расписания есть соответствующее разрешение. При удалении резервной копии потребуются вести пароль клиента.

- **Восстановить резервную копию.** Это действие запускает процесс восстановления резервной копии на локальной файловой системе клиента. При восстановлении резервной копии клиент должен выбрать место для восстановления файлов резервной копии. Рекомендуется использовать временный каталог для операций с резервными копиями (например, /rubackup-tmp).

Если при восстановлении РК базы данных в файле /opt/rubackup/etc/rb_module_pg_dump_database.conf параметр direct_restore имеет значение yes, то произойдёт восстановление таблиц, которые были в БД на момент создания РК. Восстановление происходит в базу данных с таким же названием, как и у копируемой БД. Если при восстановлении РК отдельной таблицы в файле /opt/rubackup/etc/rb_module_pg_dump_table.conf параметр direct_restore имеет значение yes, то произойдёт восстановление в базу данных с таким же названием как и у той, к которой принадлежала таблица на момент выполнения резервного копирования.

Внимание! Если при восстановлении обнаружится, что хотя бы одна таблица, с таким же именем как и у восстанавливаемой, присутствует в целевой БД, то модуль создаст новую базу данных с суффиксом «_rbcору_[index]» и восстановит таблицы в неё. Это действие выполняется и при восстановлении РК отдельной таблицы.

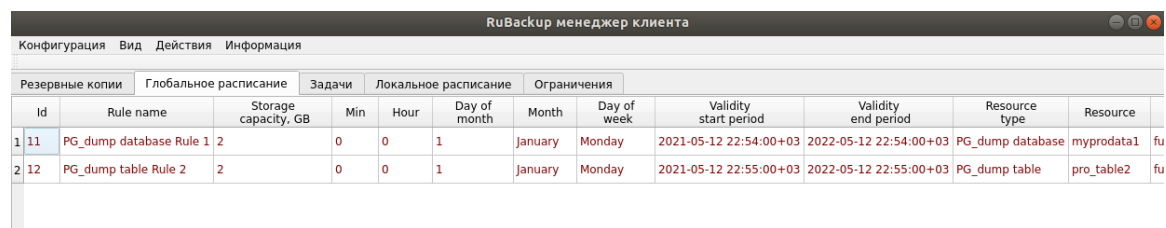
Если в файле opt/rubackup/etc/rb_module_postgres_pro_13.conf параметр direct_restore имеет значение no, то восстановленные резервные копии будут расположены в выбранном для восстановления каталоге и далее

вы сможете провести восстановление в ручном режиме. RBC не ожидает окончания восстановления всех резервных копий. Клиент должен проконтролировать на вкладке «Задачи» успешное завершение созданных задач 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. параметр `use-local-backup-directory`).

- **Проверить резервную копию.** Это действие инициирует создание задачи проверки резервной копии. 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»

Вкладка «Глобальное расписание» содержит таблицу с информацией обо всех правилах в глобальном расписании RuBackup для этого клиента (рисунок 13):



RuBackup менеджер клиента												
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	Resource	
1 11	PG_dump database Rule 1	2	0	0	1	January	Monday	2021-05-12 22:54:00+03	2022-05-12 22:54:00+03	PG_dump database	myprodata1	fu
2 12	PG_dump table Rule 2	2	0	0	1	January	Monday	2021-05-12 22:55:00+03	2022-05-12 22:55:00+03	PG_dump table	pro_table2	fu

Рисунок 13

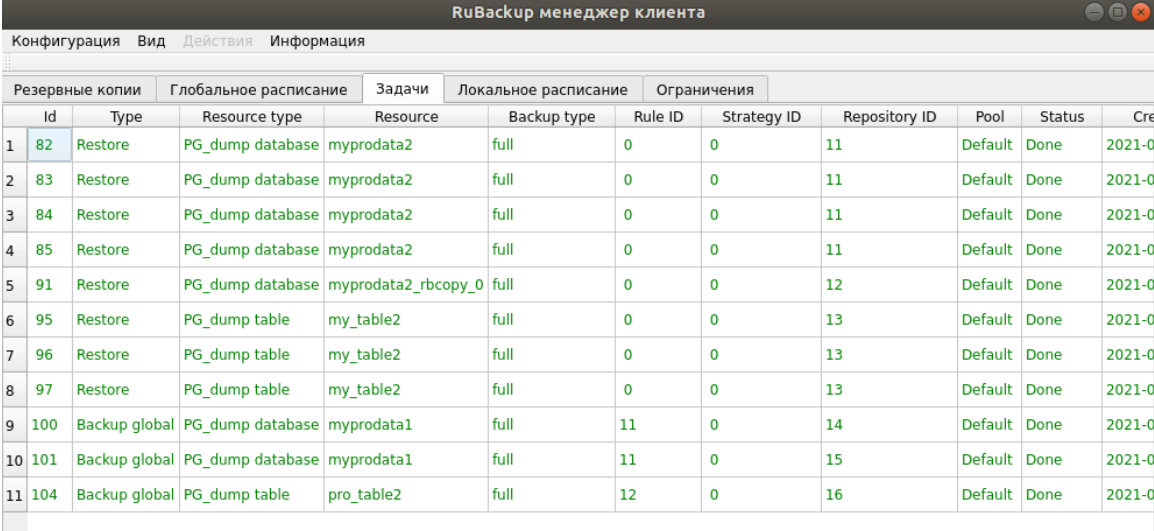
На этой вкладке клиенту доступны следующие действия:

- **Запросить новое правило.** Это действие вызывает диалог подготовки нового правила в глобальном расписании RuBackup для клиента. Запрос на добавление правила требует одобрения администратора RuBackup, одобрение может быть сделано в RBM.

- **Запросить удаление правила 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 RBM.

Вкладка «Задачи»

Вкладка «Задачи» содержит таблицу информацией обо всех задачах в главной очереди заданий RuBackup для этого клиента (рисунок 14):



Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created	
1	82	Restore	PG_dump database	myprodata2	full	0	0	11	Default	Done	2021-0
2	83	Restore	PG_dump database	myprodata2	full	0	0	11	Default	Done	2021-0
3	84	Restore	PG_dump database	myprodata2	full	0	0	11	Default	Done	2021-0
4	85	Restore	PG_dump database	myprodata2	full	0	0	11	Default	Done	2021-0
5	91	Restore	PG_dump database	myprodata2_rbcopy_0	full	0	0	12	Default	Done	2021-0
6	95	Restore	PG_dump table	my_table2	full	0	0	13	Default	Done	2021-0
7	96	Restore	PG_dump table	my_table2	full	0	0	13	Default	Done	2021-0
8	97	Restore	PG_dump table	my_table2	full	0	0	13	Default	Done	2021-0
9	100	Backup global	PG_dump database	myprodata1	full	11	0	14	Default	Done	2021-0
10	101	Backup global	PG_dump database	myprodata1	full	11	0	15	Default	Done	2021-0
11	104	Backup global	PG_dump table	pro_table2	full	12	0	16	Default	Done	2021-0

Рисунок 14

В зависимости от настроек 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. Также информация о выполнении задач клиента заносится в локальный файл журнала на хосте клиента. В RBC можно открыть окно отслеживания журнального файла (меню «**Информация**» → «**Журнальный файл**»).

Примечание – Информация о выполнении служебных задач в данной вкладке не отображается. Служебными являются задачи проверки, удаления, перемещения резервных копий, а также их копирования в другой пул.

Вкладка «Локальное расписание»

На вкладке «Локальное расписание» можно определить правила, задаваемые клиентом для каких-либо локальных ресурсов. Для работы локального расписания эта возможность должна быть включена для клиента администратором RuBackup.

Вкладка «Ограничения»

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

Утилиты командной строки клиента

RuBackup

Для управления RuBackup со стороны клиента, помимо RBC, можно использовать утилиты командной строки. Пользователь, запускающий утилиты командной строки, должен входить в группу `rubackup`.

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

`rb_archives`

Эта утилита предназначена для просмотра списка резервных копий клиента в системе резервного копирования, создания срочных резервных копий, их удаления, проверки и восстановления. Ниже представлен пример.

`# rb_archives`

```
root@postgresPro-client:~# rb_archives
```

Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
14		myprodata1	PG_dump database	full	2021-05-12 22:56:44+03	nocrypt	True	Trusted
15		myprodata1	PG_dump database	full	2021-05-12 22:56:48+03	nocrypt	True	Trusted
16		pro_table2	PG_dump table	full	2021-05-12 22:57:04+03	nocrypt	True	Trusted

`rb_schedule`

Эта утилита предназначена для просмотра имеющихся правил клиента в глобальном расписании резервного копирования. Ниже представлен пример.

`#rb_schedule`

```
root@postgresPro-client:~# rb_schedule
```

Id	Name	Resource type	Resource	Backup type	Status
11	PG_dump database Rule 1	PG_dump database	myprodata1	full	wait
12	PG_dump table Rule 2	PG_dump table	pro_table2	full	wait

`rb_tasks`

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

```
root@postgresPro-client:~# rb_tasks
```

Id	Task type	Resource	Backup type	Status
82	Restore	myprodata2	full	Done
83	Restore	myprodata2	full	Done
84	Restore	myprodata2	full	Done
85	Restore	myprodata2	full	Done
91	Restore	myprodata2_rbcopy_0	full	Done
95	Restore	my_table2	full	Done
96	Restore	my_table2	full	Done
97	Restore	my_table2	full	Done
100	Backup global	myprodata1	full	Done
101	Backup global	myprodata1	full	Done
104	Backup global	pro_table2	full	Done

Восстановление резервной копии

отдельной БД или таблицы

Ход восстановления резервной копии отдельной БД или таблицы зависит от значения параметра `direct_restore` в файле конфигурации модуля резервного копирования `/opt/rubackup/etc/rb_module_pg_dump_database.conf` и `/opt/rubackup/etc/rb_module_pg_dump_table.conf`, соответственно.

Если при восстановлении РК отдельной базы данных параметр `direct_restore` имеет значение `yes`, то произойдёт восстановление таблиц, которые были в БД на момент создания РК. Восстановление происходит в базу данных с таким же названием, как и у копируемой БД.

Если при восстановлении РК отдельной таблицы параметр `direct_restore` имеет значение `yes`, то произойдёт восстановление в базу данных с таким же названием как и у той, к которой принадлежала таблица на момент выполнения резервного копирования.

Если параметр `direct_restore` имеет значение `no`, то восстановленные резервные копии будут расположены в выбранном для восстановления каталоге, и восстановление БД или таблицы можно будет провести вручную.

Клиент может осуществить восстановление данных резервной копии в оконном Менеджере Клиента RuBackup (RBC), либо при помощи утилиты командной строки `rb_archives`.

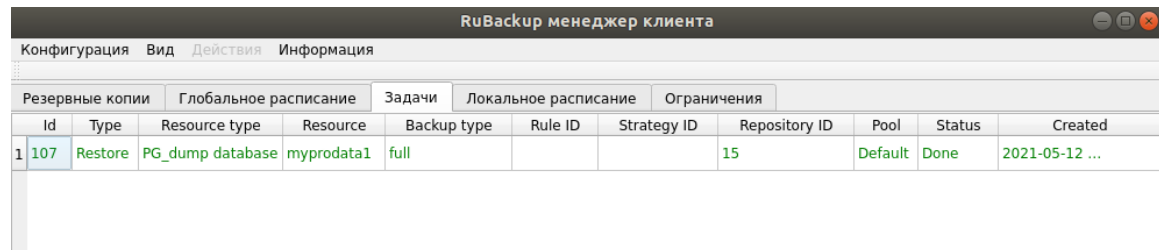
Восстановление резервной копии в RBC

Для восстановления данных резервной копии в оконном Менеджере Клиента RuBackup (RBC) выполните следующие действия:

1. Выделите нужную резервную копию и в контекстном меню выберите **Восстановить**.

2. Для восстановления потребуется ввести пароль клиента. Затем RBC выведет информационное сообщение о дальнейших действиях.
3. Укажите в качестве временного места восстановления резервных копий любой каталог (например, /ruba-backup-tmp).
4. RBC выведет информационное сообщение о создании задачи на восстановление.

Для контроля процесса восстановления RBC автоматически переключится на вкладку «Задачи», в которой можно проконтролировать результат (рисунок 15):



The screenshot shows the RuBackup client manager interface. The title bar reads "RuBackup менеджер клиента". Below the title bar are tabs: "Конфигурация", "Вид", "Действия", and "Информация". Under "Действия", there are sub-tabs: "Резервные копии", "Глобальное расписание", "Задачи", "Локальное расписание", and "Ограничения". The "Задачи" tab is active, displaying a table with the following data:

	Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created
1	107	Restore	PG_dump database	myprodata1	full			15	Default	Done	2021-05-12 ...

Рисунок 15

Восстановление при помощи утилиты rb_archives

Для восстановления резервных копий клиент может использовать утилиту командной строки rb_archives. Вызов следующий:

```
# rb_archives
```

```
root@postgresPro-client:~# rb_archives
```

Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
14		myprodata1	PG_dump database	full	2021-05-12 22:56:44+03	nocrypt	True	Trusted
15		myprodata1	PG_dump database	full	2021-05-12 22:56:48+03	nocrypt	True	Trusted
16		pro_table2	PG_dump table	full	2021-05-12 22:57:04+03	nocrypt	True	Trusted

В приведённом примере в системе резервного копирования присутствуют три резервные копии с идентификаторами 14, 15 и 16. Для восстановления резервной копии 15 необходимо выполнить команду:

```
# rb_archives -x 15
```

```
root@postgresPro-client:~# rb_archives -x 15
Password:
----> Restore archive chain: 15 < ----
Record ID: 15 has status: Trusted
TASK WAS ADDED TO QUEUE:108
```

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

Проконтролировать процесс восстановления можно при помощи утилиты rb_tasks:

```
#rb_tasks
```

```
root@postgresPro-client:~# rb_tasks
```

Id	Task type	Resource	Backup type	Status	Created
107	Restore	myprodata1	full	Done	2021-05-12 23:30:26+03
108	Restore	myprodata1	full	Done	2021-05-12 23:32:06+03

Вы можете проконтролировать процесс восстановления в файле журнала при помощи вызова:

```
# tail -f /opt/rubackup/log/RuBackup.log
```

```
root@postgresPro-client:~# tail -f /opt/rubackup/log/RuBackup.log
Fri May 14 17:52:32 2021: [RBC] Request to restore next archive(s) ID from repository: 15 to: /root
Fri May 14 17:52:33 2021: RuBackup server commands: Run task ID: 108 Resource type: 31 Module: PG_dump database Resource: myprodata1 Media server: ruba
ckup-server.rubackup.local
Fri May 14 17:52:33 2021: Set unlimited bandwidth for task ID: 108
Fri May 14 17:52:33 2021: Create a file: /root/postgresPro-client_TaskID_101_RuleID_11_D2021_5_14H17_17_14_BackupType_1_ResourceType_31.tar
Fri May 14 17:52:34 2021: md5sum of transferred file is ok: a73b6123a93a6155bdaaf7309a675690
Fri May 14 17:52:34 2021: Transfer file is succeeded: /root/postgresPro-client_TaskID_101_RuleID_11_D2021_5_14H17_17_14_BackupType_1_ResourceType_31.tar
Fri May 14 17:52:34 2021: Execute OS command: /opt/rubackup/modules/rb_module_pg_dump_database -r /root/postgresPro-client_TaskID_101_RuleID_11_D2021_5_14H17_17_14_BackupType_1_ResourceType_31.dump
Fri May 14 17:52:34 2021: postgresPro-client_TaskID_101_RuleID_11_D2021_5_14H17_17_14_BackupType_1_ResourceType_31.dump
Fri May 14 17:52:34 2021: postgresPro-client_TaskID_101_RuleID_11_D2021_5_14H17_17_14_BackupType_1_ResourceType_31.snap
Fri May 14 17:52:34 2021: Task was done. ID: 108
```