

**RuBackup**

Система резервного копирования и восстановления данных

# **Резервное копирование и восстановление СУБД Jatoba**



**RuBackup**

Версия 1.9

2022 г.

# Содержание

Введение.....	3
Подготовка хоста СУБД Jatoba.....	5
Установка клиента RuBackup.....	5
Установка пакетов модулей резервного копирования.....	5
Удаление клиента RuBackup.....	6
Подготовка СУБД Jatoba.....	7
Подготовка сервера с СУБД Jatoba.....	7
Создание пользователя СУБД для безопасного выполнения базовой резервной копии Jatoba.....	9
Принцип выполнения базового резервного копирования Jatoba.....	12
Принцип выполнения инкрементального резервного копирования Jatoba..	14
Принцип восстановления резервной копии Jatoba.....	15
Мастер-ключ.....	17
Защитное преобразование резервных копий.....	18
Менеджер Администратора RuBackup (RBM).....	20
Менеджер клиента RuBackup.....	26
Утилиты командной строки клиента RuBackup.....	31
Восстановление резервной копии СУБД Jatoba.....	32
Восстановление резервной копии в RBC.....	32
Восстановление при помощи утилиты rb_archives.....	33

## Введение

Система резервного копирования (СРК) RuBackup поддерживает резервное копирование СУБД Jatoba 1 (основана на PostgreSQL 11.5).

Принцип резервного копирования Jatoba 1 (aka PostgreSQL 11.5) с использованием RuBackup состоит в периодическом создании базовых резервных копий экземпляра СУБД по определённому 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 Jatoba 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 медиасервере, 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 Jatoba 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 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 Jatoba

Для возможности резервного копирования данных СУБД Jatoba при помощи СРК RuBackup на сервер следует установить следующие пакеты:

- `rubackup-client.deb` – клиент резервного копирования,
- `rubackup-jatoba.deb` – модуль резервного копирования данных Jatoba.

## Установка клиента RuBackup

Для осуществления резервного копирования и восстановления данных СУБД Jatoba 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 Jatoba клиент RuBackup должен работать от имени суперпользователя (`root` в Linux и Unix).

Подробно процедура установки клиента описана в «Руководстве по установке серверов резервного копирования и Linux клиентов RuBackup», для операционной системы Windows — в «Руководстве по установке Windows клиентов RuBackup».

## Установка пакетов модулей резервного копирования

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

Для установки пакета модулей используйте следующий вызов:

Для установки пакета модулей используйте следующий вызов:

```
# dpkg -i rubackup-jatoba.deb
```

Выбор ранее не выбранного пакета `rubackup-jatoba`.

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

```
Подготовка к распаковке rubackup-jatoba.deb ...
```

```
Распаковывается rubackup-jatoba (2021-01-18) ...
```

```
Настраивается пакет rubackup-jatoba (2021-01-18) ...
```

## 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:

```
$ sudo systemctl disable rubackup_client  
$ sudo systemctl daemon-reload
```

Для удаления клиента RuBackup и модуля **rubackup-jatoba** используйте следующие команды.

```
$ sudo apt remove rubackup-kvm  
$ sudo apt remove rubackup-client
```

При необходимости удалить клиент RuBackup из конфигурации СРК, это может сделать системный администратор RuBackup при помощи оконного Менеджера Администратора RBM.

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

1. Выполнить команду:

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

2. Перезагрузить операционную систему:

```
$ init 6
```

## Подготовка СУБД Jatoba

Чтобы подготовить СУБД Jatoba к выполнению резервного копирования при помощи СРК RuBackup необходимо выполнить следующие действия.

### Подготовка сервера с СУБД Jatoba

Для непрерывного архивирования и восстановления СУБД Jatoba требуется включить архивирование WAL.

Для этого в конфигурационном файле СУБД Jatoba `/var/lib/jatoba/1/data/postgresql.conf` необходимо установить параметры (расположение конфигурационного файла может быть другим в других дистрибутивах, проконсультируйтесь по этому поводу у вашего администратора Jatoba):

```
wall_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
```

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/jatoba/1/data'
```

После внесения изменений необходимо перезапустить Jatoba командой:

```
$ sudo systemctl restart jatoba-1
```

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

В настройках RuBackup для каждой СУБД Jatoba 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 Jatoba, где это требуется.

Объем необходимого пространства под архивные файлы WAL на сервере Jatoba 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 потребуются 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.

**Внимание! 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/
```

С помощью этого же подхода можно оценить требуемый объем хранилища на сервере резервного копирования RuBackup.

Для правильной работы клиента RuBackup значения параметр `archive_catalog` в конфигурации RuBackup и параметр `archive_command` в конфигурационном файле Jatoba должны быть идентичны для одной и той же СУБД.

Параметр `archive_timeout` определяет период времени в секундах, по окончании которого сервер Jatoba должен переключится на новый сегмент WAL.

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

```
# sudo systemctl restart jatoba-1
```

При настройке резервного копирования Jatoba в ОС Astra Linux SE 1.6 необходимо в файле `/etc/parsec/mswitch.conf` для параметра `zero_if_notfound` установить значение `yes` и затем перезагрузить сервис Jatoba:

```
$ sudo systemctl restart jatoba-1
```



## Создание пользователя СУБД для безопасного выполнения базовой резервной копии Jatoba

Пользователь Jatoba для выполнения операции создания базовой резервной копии должен обладать правами на выполнение функций начала и окончания резервного копирования экземпляра Jatoba. Для настройки необходимо выполнить следующие действия:

1. Вызовите psql при помощи команды:

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

2. 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;
```

Вместо пользователя *rubackup\_backuper* вы можете создать любого другого с соответствующим набором прав. В файле */opt/rubackup/etc/rb\_module\_jatoba1.conf* необходимо указать правильное значение пользователя и его пароль:

```
root@jatoba:~# cat /opt/rubackup/etc/rb_module_jatoba1.conf  
username rubackup_backuper  
password 12345  
port 5432  
archive_catalog /opt/rubackup/mnt/postgresql_archives  
pg_ctl /usr/jatoba-1/bin/pg_ctl  
exclude_file /opt/rubackup/etc/postgresql.exclude  
auto_remove_wal yes  
direct_restore yes  
postgresql_admin postgres
```

Параметры и значения конфигурационного файла */opt/rubackup/etc/rb\_module\_jatoba1.conf* представлены в таблице 1.

После выполнения подготовки сервера СУБД Jatoba 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 Jatoba готов к работе.

Таблица 1 – Параметры и значения конфигурационного файла

Параметр	Назначение	Значение по умолчанию
username	Имя пользователя в СУБД Jatoba, 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_tcl	Местонахождение pg_ctl	/usr/jatoba-1/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 <b>yes</b> архивные WAL будут удалены из каталога archive_catalog после выполнения резервного копирования (если они включены в резервную копию)	yes
direct_restore	<p>При значении <b>yes</b>:            При восстановлении резервной копии служба jatoba 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 jatoba. Старт службы jatoba необходимо выполнить в ручном режиме.</p> <p>При значении <b>no</b>:            Файлы цепочки резервных копий будут восстановлены в выбранный пользователем каталог. Восстановление СУБД в данном случае выполняется администратором СУБД в ручном режиме</p>	yes
postgresql_admin	Login администратора Jatoba в операционной системе	postgres



# Принцип выполнения базового резервного копирования Jatoba

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@jatoba:~$ 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/jatoba/1/data/
```

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);
```

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.

4. 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@jatoba:~$ 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 Jatoba

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 Jatoba

Данный ручной метод может быть использован при ручном восстановлении служебной базы данных RuBackup, если для её работы используется СУБД Jatoba и выполнялось её 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 Jatoba необходимо выполнить следующие действия:

1. Остановить экземпляр Jatoba, если он работает:

```
# sudo -iu postgres /usr/jatoba-1/bin/pg_ctl stop -D  
/var/lib/jatoba/1/data/
```

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/jatoba/1/data/ && tar cfv - *) |  
(cd ~/emergency_copy && tar xf - )
```

3. Очистить каталог кластера баз данных:

```
# sudo -iu postgres rm -rf /var/lib/jatoba/1/data/*
```

4. Восстановить данные из резервных копий (например, установив значение `no` для параметра `direct_restore` в файле `/opt/rubackup/etc/rb_module_jatoba1.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. Создать файл `recoverу.conf` со следующим содержимым:

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

## 6. Запустить восстановление Jatoba:

```
# sudo -iu postgres /usr/jatoba-1/bin/pg_ctl start -D  
/var/lib/jatoba/1/data/
```

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

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



## 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 надежное 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 `rbcrypt`. Ключ для защитного преобразования резервных копий располагается на хосте клиента в файле `/opt/rubackup/keys/master-key`. Защитное преобразование данных при помощи `rbcrypt` 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 `rbcrypt`. При таком режиме работы нет необходимости указывать алгоритм преобразования в правиле резервного копирования, иначе архив будет повторно преобразован с использованием мастер-ключа.

Алгоритмы, доступные для выполнения защитного преобразования представлены в таблице 2.

Таблица 2 – Алгоритмы защитного преобразования, доступные в утилите `rbcrypt`

Алгоритм	Длина ключа, бит	Примечание
Anubis	128, 256	
Aria	128, 256	
CAST6	128, 256	
Camellia	128, 256	
Kalyna	128, 256, 512	Украинский национальный стандарт

Алгоритм	Длина ключа, бит	Примечание
		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	Китайский национальный стандарт для беспроводных сетей
Speck	128, 256	
Threefish	256, 512, 1024	
Twofish	128, 256	

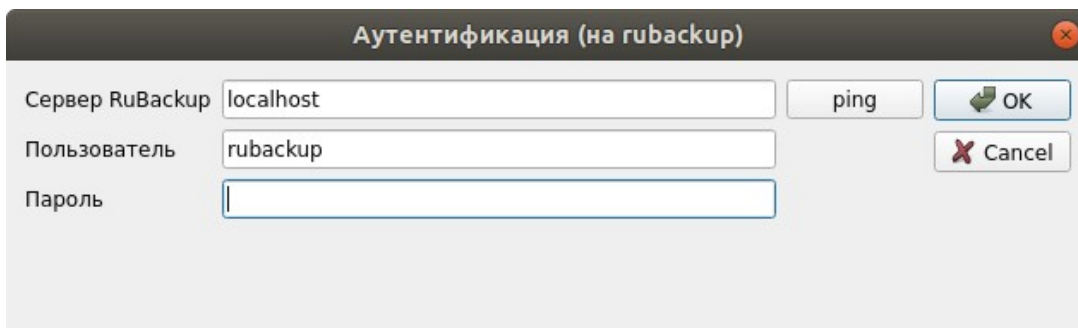
# Менеджер Администратора RuBackup (RBM)

Оконное приложение Менеджер Администратора RuBackup (RBM) предназначено для администрирования серверной группировки RuBackup, включая управление клиентами, глобальным расписанием, хранилищами резервных копий и другими параметрами RuBackup.

Для запуска RBM следует выполнить команду:

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

После чего необходимо в открывшееся окно «Аутентификация» ввести наименование сервера Rubackup, имя пользователя и пароль (рисунок 1).



Аутентификация (на rubackup)

Сервер RuBackup localhost ping OK

Пользователь rubackup Cancel

Пароль

Рисунок 1

Для резервного копирования 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 2). Все новые клиенты должны быть авторизованы в системе резервного копирования RuBackup.

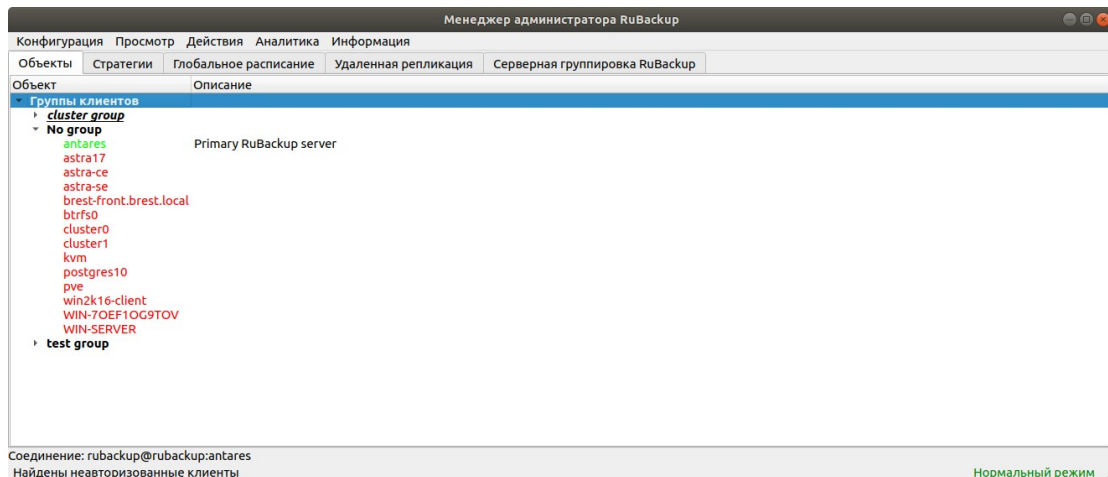


Рисунок 2

Для авторизации неавторизованного клиента в RBM выполните следующие действия:

1. Откройте меню **Действия** → **Клиенты** → **Авторизовать клиентов** (рисунок 3).

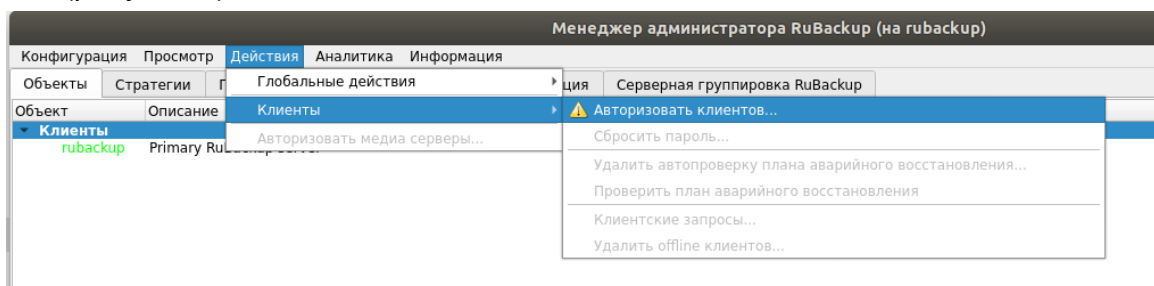


Рисунок 3

2. Выберите нужного неавторизованного клиента и нажмите **Авторизовать** (рисунок 4).

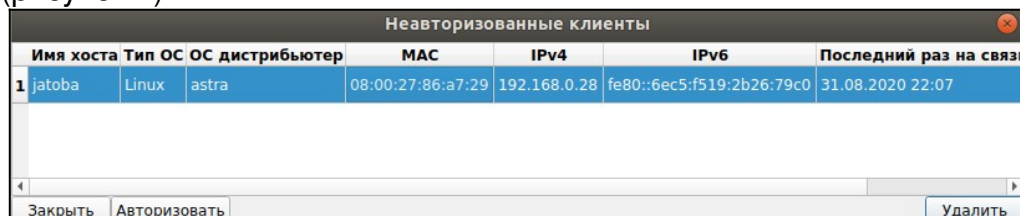


Рисунок 4

После авторизации новый клиент будет виден в главном окне RBM (рисунок 5):

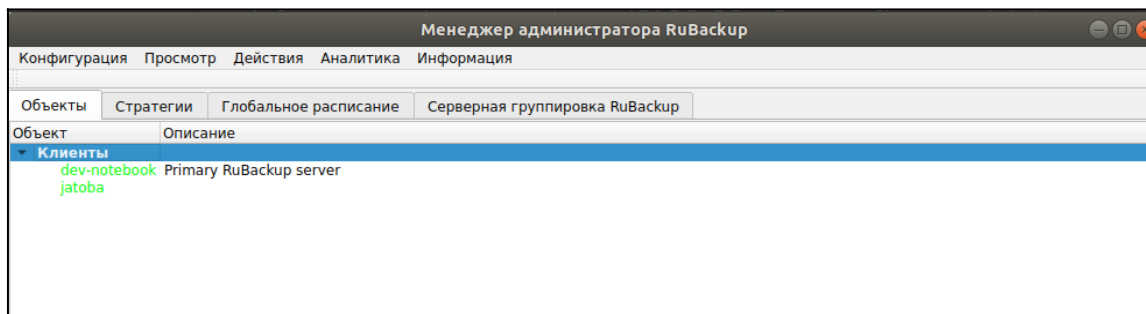


Рисунок 5

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*).

Чтобы выполнять регулярное резервное копирование СУБД Jatoba, необходимо создать правило в глобальном расписании (в случае групповых операций можно так же использовать стратегии резервного копирования). С этой целью выполните следующие действия:

1. Выберите клиентский хост, на котором находится СУБД Jatoba, и добавьте правило резервного копирования (рисунок 6):

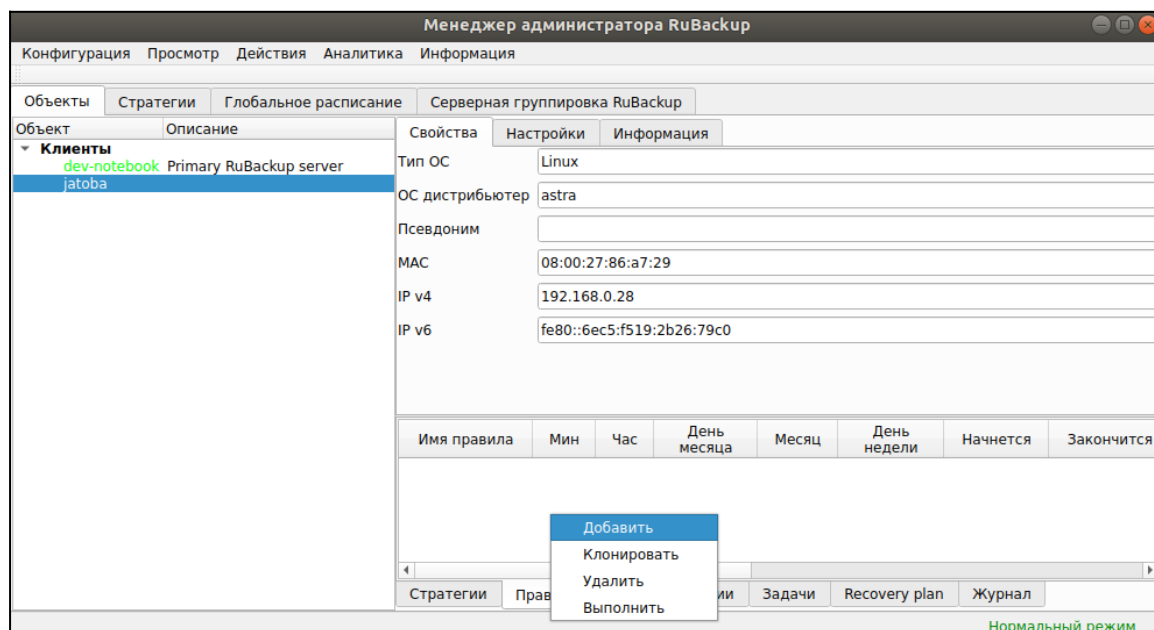


Рисунок 6

2. Выберите тип ресурса: «jatoba 1» (рисунок 7):

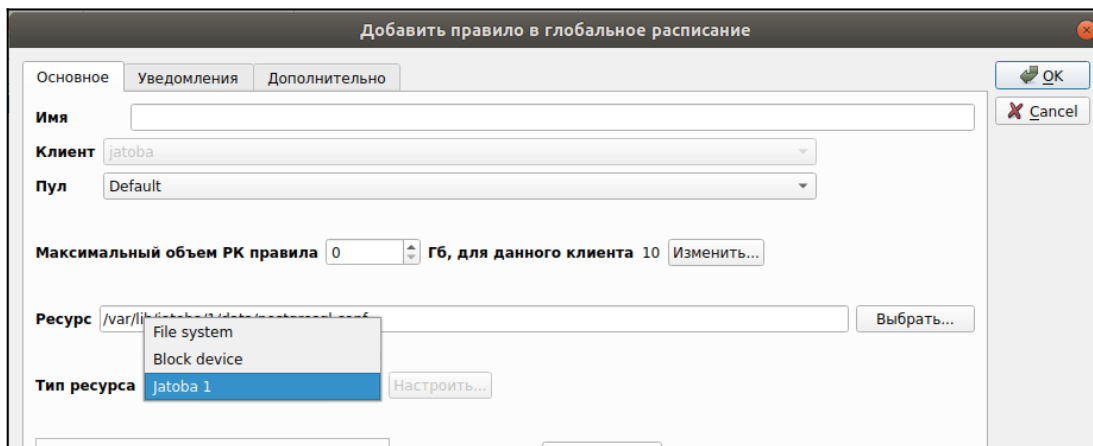


Рисунок 7

В качестве ресурса автоматически будет подставлено значение главного конфигурационного файла СУБД: `/var/lib/jatoba/1/data/postgresql.conf`.

- Установите прочие настройки: название правила пул хранения данных, расписание резервного копирования, тип резервного копирования, максимальный объем для резервных копий данного правила (в ГБ), срок хранения, через какой промежуток времени требуется выполнить проверку резервной копии (рисунок 8):

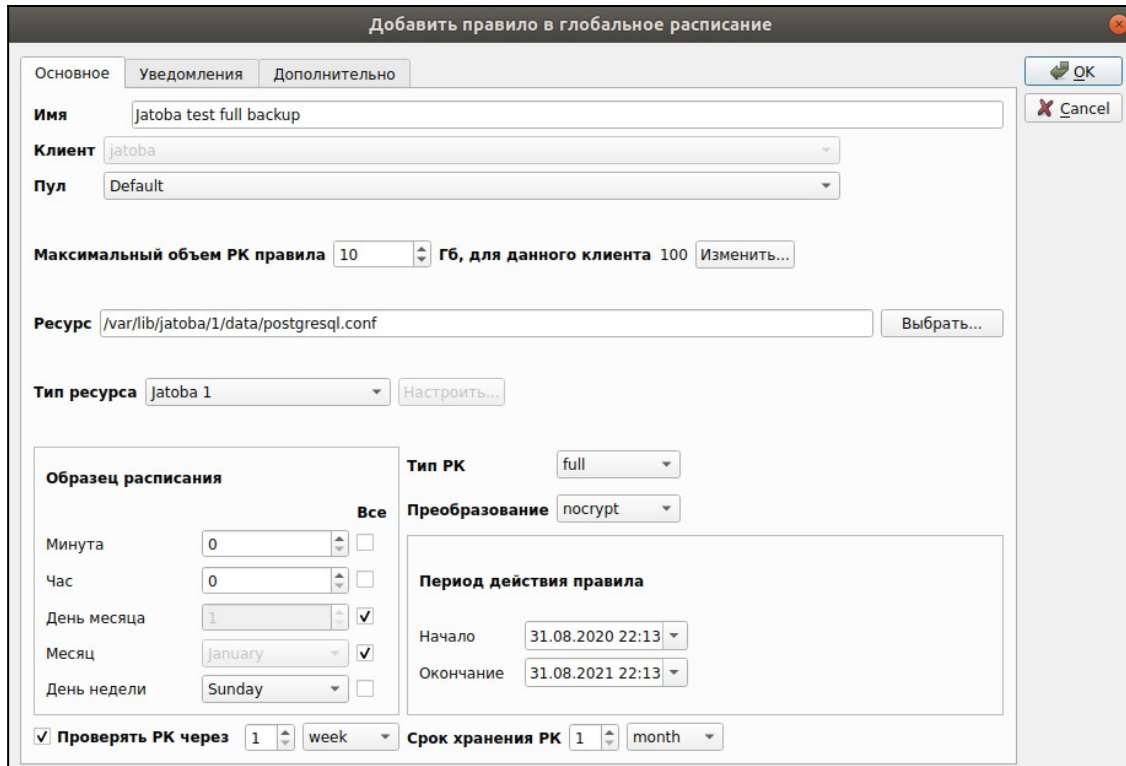


Рисунок 8

- 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 9).

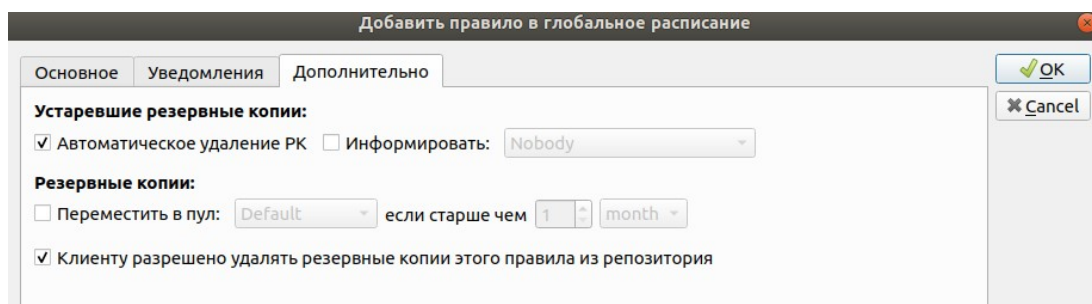


Рисунок 9

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».

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;
- выполнить сжатие резервной копии на клиенте или на сервере после передачи ему резервной копии;
- периодически выполнять проверку целостности резервной копии;
- хранить резервные копии определенный срок, по окончании которого удалять их из хранилища резервных копий и из записей репозитория, либо уведомлять клиента об окончании срока хранения;
- через определенный срок после создания резервной копии автоматически переместить ее 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 `rb_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

Принцип взаимодействия клиентского менеджера с системой резервного копирования состоит в том, что пользователь может сформировать ту или иную команду (желаемое действие) и отправить её 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 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 во вкладке «Задачи»).

Графический интерфейс клиентского менеджера поддерживает русский и английский языки.

Для запуска RBC следует выполнить команды:

```
# ssh X user@jatoba-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 (рисунок 10):

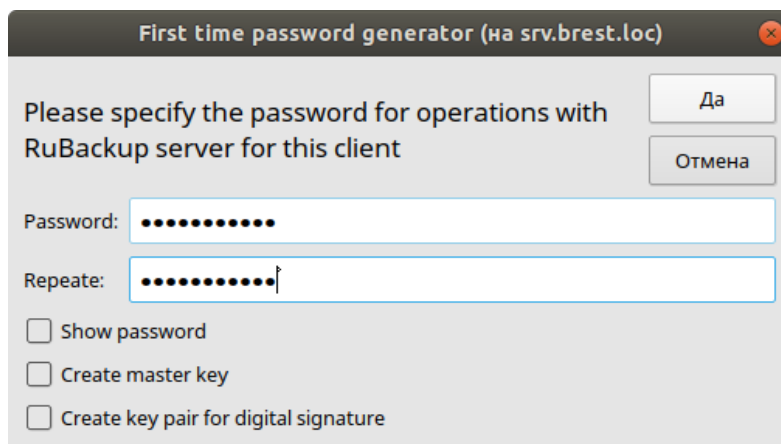


Рисунок 10

В случае успешного выполнения появится окно (рисунок 11):

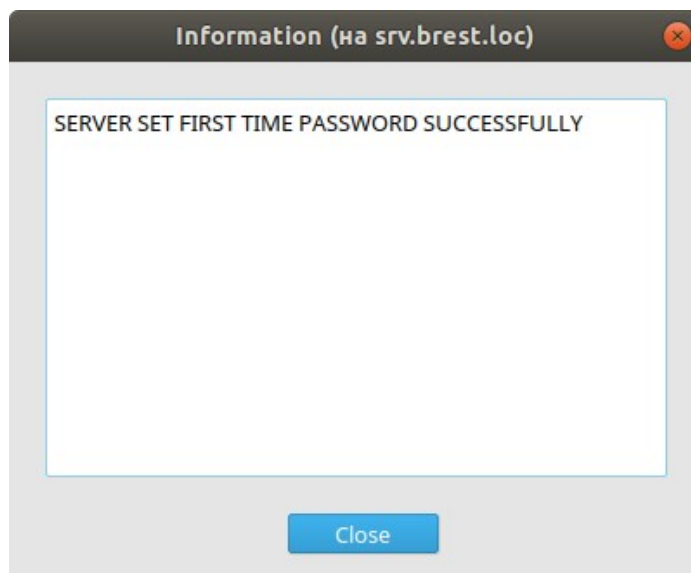


Рисунок 11

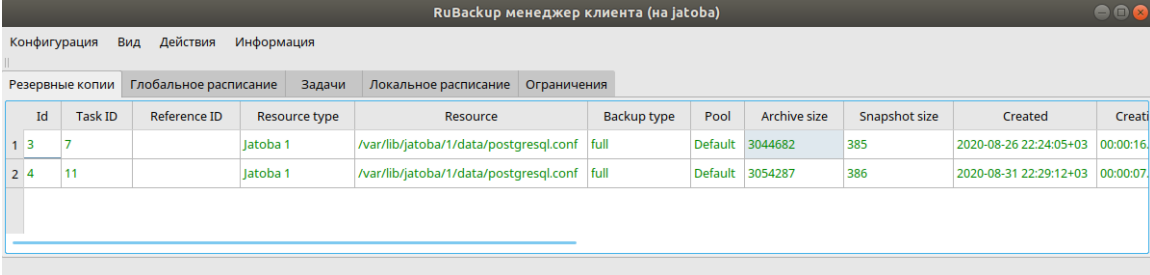
Хэш пароля восстановления хранится в базе данных RuBackup сервера. При необходимости можно изменить пароль при помощи клиентского менеджера (меню «**Конфигурация**» → «**Изменить пароль**»).

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

### **Вкладка «Резервные копии»**

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 (рисунок 12):



Id	Task ID	Reference ID	Resource type	Resource	Backup type	Pool	Archive size	Snapshot size	Created	Created
1	3	7	Jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full	Default	3044682	385	2020-08-26 22:24:05+03	00:00:16
2	4	11	Jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full	Default	3054287	386	2020-08-31 22:29:12+03	00:00:07

Рисунок 12

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 вести 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_jatoba1.conf` параметр `direct_restore` имеет значение **yes**, то произойдёт остановка сервиса Jatoba, 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 jatoba. Старт службы jatoba необходимо выполнить в ручном режиме. В том случае, если в файле `/opt/rubackup/etc/rb_module_jatoba1.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.

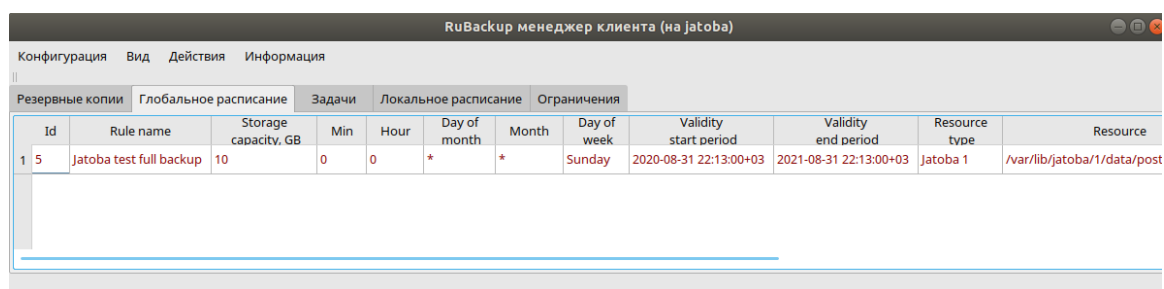
Клиентский менеджер не ожидает окончания восстановления всех резервных копий, пользователь должен проконтролировать во вкладке «Задачи» что все созданные задачи на восстановление данных завершились успешно (статус задач «Done»). Для успешного выполнения этого действия требуется наличие достаточного свободного 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 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.

### Вкладка «Глобальное расписание»



RuBackup менеджер клиента (на jatoba)												
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	5	Jatoba test full backup	10	0	0	*	*	Sunday	2020-08-31 22:13:00+03	2021-08-31 22:13:00+03	Jatoba 1	/var/lib/jatoba/1/data/post

Рисунок 13

В таблице вкладки «Глобальное расписание» содержится информация обо всех правилах в глобальном расписании RuBackup для этого клиента (рисунок 13).

Во вкладке «Глобальное расписание» пользователю доступны следующие действия:

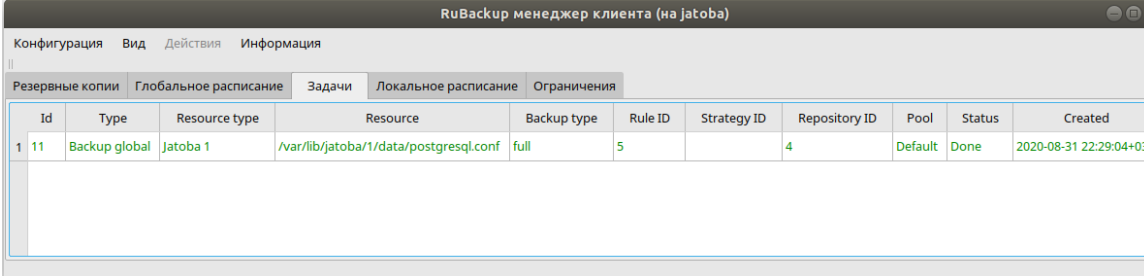
#### 1) Запросить новое правило.

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	11	Backup global	jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full	5	4	Default	Done	2020-08-31 22:29:04+03

Рисунок 14

В таблице вкладки «Задачи» (рисунок 14) 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 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»).

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»

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

Управление RuBackup со стороны клиента на хосте СУБД Jatoba осуществляется утилитами командной строки. При первом использовании будет запрошен пароль для хоста. Этот пароль будет требоваться при восстановлении или удалении резервных копий.

### rb\_archives

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

```
root@jatoba:~#
root@jatoba:~#
root@jatoba:~# rb_archives
```

Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
3		/var/lib/jatoba/1/data/postgresql.conf	Jatoba 1	full	2020-08-26 22:24:05+03	nocrypt	True	Trusted
4		/var/lib/jatoba/1/data/postgresql.conf	Jatoba 1	full	2020-08-31 22:29:12+03	nocrypt	True	Trusted

```
root@jatoba:~#
```

### rb\_schedule

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

```
root@jatoba:~#
root@jatoba:~# rb_schedule
```

Id	Name	Resource type	Resource	Backup type	Status
5	Jatoba test full backup	Jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full	wait

```
root@jatoba:~#
```

### rb\_tasks

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

```
root@jatoba:~#
root@jatoba:~# rb_tasks
```

Id	Task type	Resource	Backup type	Status	Created
11	Backup global	/var/lib/jatoba/1/data/postgresql.conf	full	Done	2020-08-31 22:29:04+03

```
root@jatoba:~#
```

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

# Восстановление резервной копии

## СУБД Jatoba

Ход восстановления резервной копии СУБД Jatoba зависит от значения параметра `direct_restore` в файле конфигурации модуля резервного копирования `/opt/rubackup/etc/rb_module_jatoba1.conf`.

В том случае, если в файле `/opt/rubackup/etc/rb_module_jatoba1.conf` параметр `direct_restore` имеет значение **yes**, то произойдёт остановка сервиса `jatoba`, 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 `jatoba`. Старт службы `jatoba` необходимо выполнить в ручном режиме.

В том случае, если в файле `/opt/rubackup/etc/rb_module_jatoba1.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.

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

В случае восстановления инкрементальной резервной копии будет сформирована цепочка восстановления: вначале будет восстановлена полная резервная копия, на которую будут наложены изменения из инкрементальных резервных копий.

## Восстановление резервной копии в RBC

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

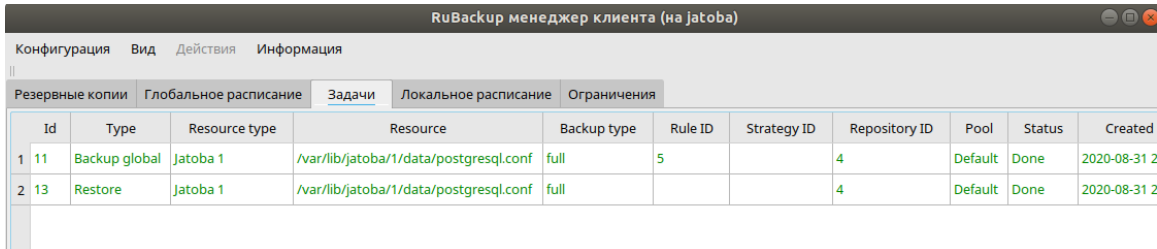
1. Выделите нужную резервную копию и в контекстном меню выберите «Восстановить».
2. Для восстановления потребуется ввести пароль клиента. Затем RBC выведет информационное сообщение о дальнейших действиях.



3. Укажите в качестве временного места восстановления резервных копий каталог, отдельный от каталога кластера баз данных (/var/lib/jatoba/1/data).

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):



Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created
1	11	Backup global	Jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full	5	4	Default	Done	2020-08-31 2
2	13	Restore	Jatoba 1	/var/lib/jatoba/1/data/postgresql.conf	full		4	Default	Done	2020-08-31 2

Рисунок 15

Если восстановление резервной копии происходило с заменой содержимого каталога кластера баз данных и остановкой сервиса jatoba, запустить СУБД необходимо вручную при помощи команды:

```
# /usr/bin/sudo -iu postgres /usr/jatoba-1/bin/pg_ctl start -D
/var/lib/jatoba/1/data
```

## Восстановление при помощи утилиты rb\_archives

Для восстановления резервных копий клиент может использовать утилиту командной строки rb\_archives. Вызов следующий:

```
# rb_archives
```

Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
3		/var/lib/jatoba/1/data/postgresql.conf	Jatoba 1	full	2020-12-01 12:02:00	nocrypt	True	Trusted
4		/var/lib/jatoba/1/data/postgresql.conf	Jatoba 1	full	2020-12-01 15:02:08	nocrypt	True	Trusted

В приведённом примере в системе резервного копирования присутствуют три резервные копии с идентификаторами 3 и 4. Для восстановления резервной копии 4 необходимо выполнить команду:

```
# rb_archives -x 4
Password:
----> Restore archive chain: 4 < ----
```

Record ID: 4 has status: Trusted

[RBC] Request to restore next archive(s) ID from repository: 4  
to: /root

TASK WAS ADDED TO QUEUE:13

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

Id	Task type	Resource	Backup type	Status	Created
11	Backup global	/var/lib/jatoba/1/data/postgresql.conf	full	Done	2020-12-02 15:06:45+03
13	Restore	/var/lib/jatoba/1/data/postgresql.conf	full	Done	2021-01-29 17:45:00+03

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

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

```

Fri Jan 29 17:45:00 2021: Connected to RuBackup media server: 192.168.0.50
Fri Jan 29 17:45:00 2021: Set unlimited bandwidth for task ID: 13
Fri Jan 29 17:45:00 2021: Create a file:
rootjatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23.tgz
Fri Jan 29 17:45:01 2021: md5sum of transferred file is ok: e4d8916db28063a6f05e8c8f379b2fd8
Fri Jan 29 17:45:01 2021: Transfer file is succeeded:
/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23.tgz
Fri Jan 29 17:45:01 2021: Execute OS command: /opt/rubackup/modules/rb_module_jatoba1 -r
/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23.tgz -z 4 -
e
last:true,tmp_catalog:tmp,rbd_hash_algorithm:sha,rbd_hash_length:512,rbd_block_size:10485
76,granul
ar_restore:no,without_deployment_restore:no,start_wal_location:000000020000000000000007,s
top_wal_lo
cation:000000020000000000000007,config_file:/var/lib/jatoba/1/data/postgresql.conf -d
/root 2>&l
Fri Jan 29 17:45:02 2021: Remove obsoleted file:
/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23.tgz
Fri Jan 29 17:45:02 2021: Direct restore PostgreSQL database cluster: /var/lib/jatoba/1/data...
Fri Jan 29 17:45:02 2021: PostgreSQL is up now. Shutdown required
Fri Jan 29 17:45:03 2021: PostgreSQL shutted down
Fri Jan 29 17:45:03 2021: PostgreSQL DB files catalog:
/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23/var/
lib/jatoba/1/data
Fri Jan 29 17:45:03 2021: WAL files catalog:
/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23/opt/
rubackup/mnt/postgresql_archives
Fri Jan 29 17:45:03 2021: Found tablespace map...
Fri Jan 29 17:45:03 2021: Clean PostgreSQL directory: /var/lib/jatoba/1/data
Fri Jan 29 17:45:03 2021: Clean PostgreSQL directory: /opt/rubackup/mnt/postgresql_archives
Fri Jan 29 17:45:03 2021:

```

```
Copy:/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23/
var/lib/jatoba/1/data to: /var/lib/jatoba/1/data
Fri Jan 29 17:45:03 2021:
Copy:/root/jatoba_TaskID_11_RuleID_5_D2021_1_29H17_45_00_BackupType_1_ResourceType_23/
opt/rubackup/mnt/postgresql_archives          to          /opt/rubackup/mnt/postgresql_archives
Fri Jan 29 17:45:03 2021: PostgreSQL was shutted down. Start required
Fri Jan 29 17:45:03 2021: To start PostgreSQL recovery you MUST run: /usr/bin/sudo -iu postgres
/usr/jatoba-1/bin/pg_ctl                       start          -D              /var/lib/jatoba/1/data
Fri Jan 29 17:45:03 2021: Task was done. ID: 13
```

Если восстановление резервной копии происходило с заменой содержимого каталога кластера баз данных и остановкой сервиса jatoba, запустить СУБД необходимо вручную при помощи команды:

```
# /usr/bin/sudo -iu postgres /usr/jatoba-1/bin/pg_ctl start -D/var/lib/jatoba/1/data
```