

**RuBackup**

Система резервного копирования и восстановления данных

# **Резервное копирование и восстановление базы данных**

## **RuBackup**



**RuBackup**

Версия 2.3.0

25.10.2024

# Содержание

Введение.....	3
Настройка резервного копирования базы данных RuBackup.....	4
Восстановление базы данных RuBackup.....	8
Восстановление функционирующей базы данных.....	8
Восстановление поврежденной базы данных.....	9
Использование Менеджера администратора RuBackup (RBM) для восстановления резервных копий СУБД PostgreSQL.....	10
Использование Менеджера клиента RuBackup (RBC) для восстановления резервных копий СУБД PostgreSQL.....	12
Использование rb_archives для восстановления резервной копии СУБД PostgreSQL.....	16

## Введение

Информация о системе резервного копирования RuBackup хранится в базе данных СУБД PostgreSQL. В данном руководстве описана настройка резервного копирования базы данных системы резервного копирования RuBackup.

Перед прочтением данного документа рекомендуется ознакомиться с руководством «Резервное копирование и восстановление PostgreSQL (Модуль Universal)».

# Настройка резервного копирования базы данных RuBackup

Перед настройкой резервного копирования базы данных убедитесь, что в вашей серверной группировке сконфигурирован основной сервер резервного копирования. В том случае, если база данных располагается на том же сервере, что и RuBackup, клиент должен быть запущен от имени пользователя root.

## 1. Подготовьте СУБД PostgreSQL, в которой располагается база данных RuBackup

Установите и настройте модуль PostgreSQL Universal на хост, где располагается база данных RuBackup, согласно документации «Резервное копирование и восстановление СУБД PostgreSQL».

Внимание! Каталог `/opt/rubackup/mnt/postgresql_archives/` должен существовать и быть доступен для записи и чтения пользователю `postgres`, а также пользователю, под контролем которого работает клиент RuBackup. Обеспечить это можно командами:

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

## 2. Создайте правило или стратегию резервного копирования базы данных RuBackup

Для возможности быстрого и надежного восстановления базы данных RuBackup рекомендуется выполнять ежедневное резервное копирование архивных WAL-файлов и один раз в неделю - полное резервное копирование архивных WAL-файлов.

Для этого создайте стратегию - зайдите в раздел «Стратегии» и нажмите на кнопку «Добавить» (рисунок 1).

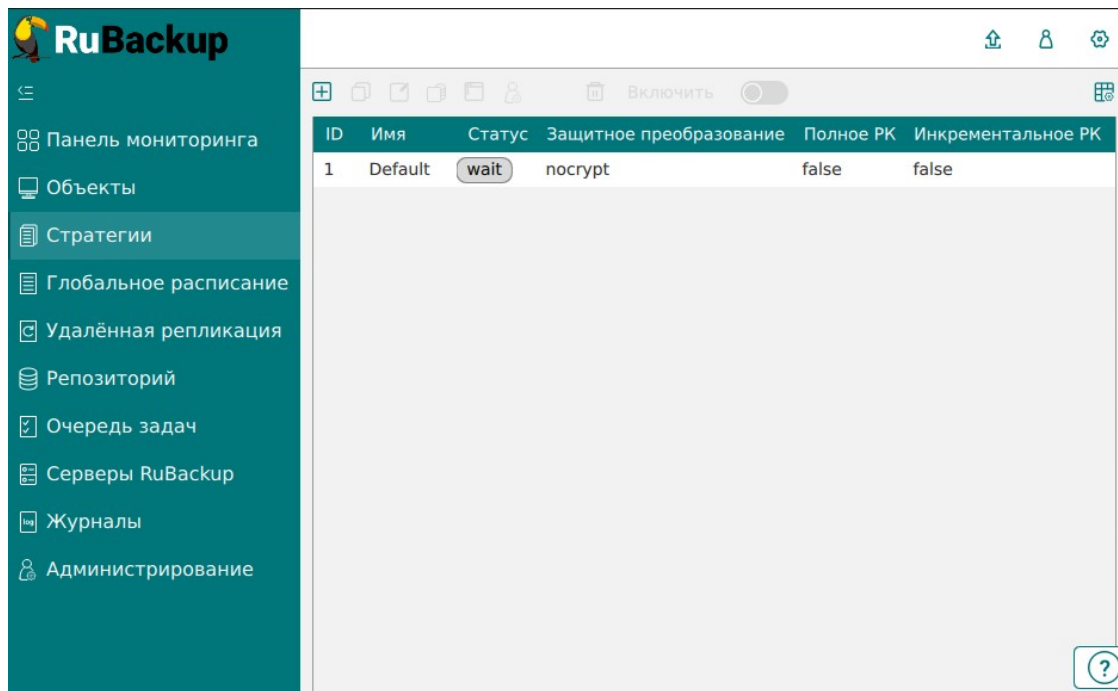


Рисунок 1

Выберите следующие настройки Полного резервного копирования и Инкрементального резервного копирования и нажмите на кнопку «Применить» (Рисунок 2).

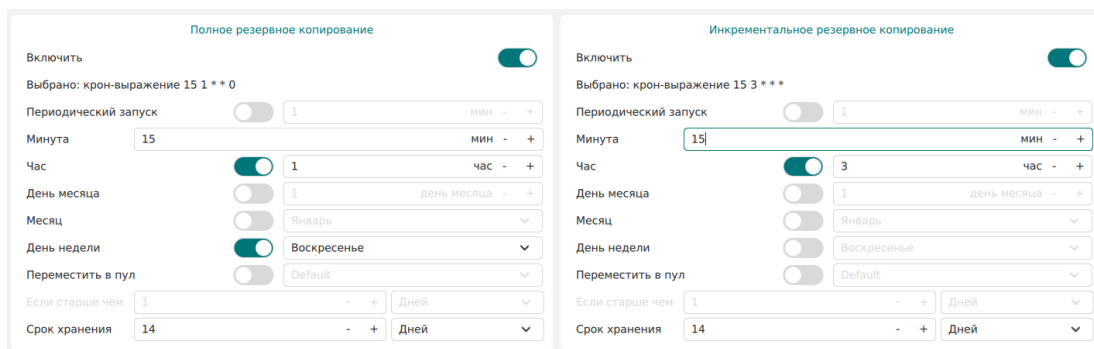


Рисунок 2

В разделе «Стратегии» выберите созданную стратегию щелчком по левой кнопке мыши и нажмите на значок «Правила» (Рисунок 3).

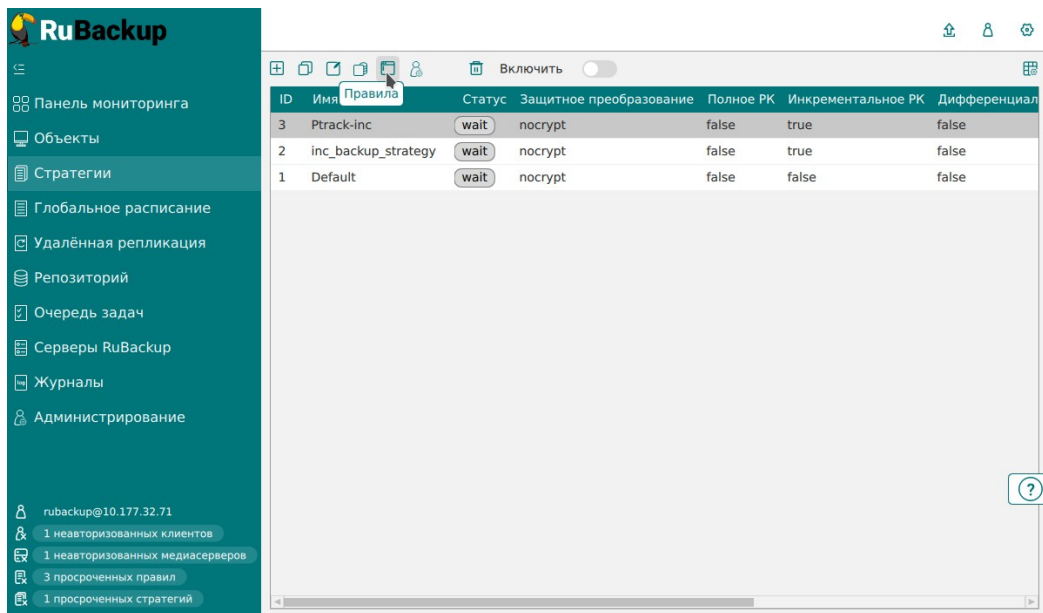


Рисунок 3

Добавьте правило с помощью кнопки «Добавить». Выберите Клиент, Тип ресурса (PostgreSQL universal) и Ресурс (рисунок 4).

Параметры правила

Клиент

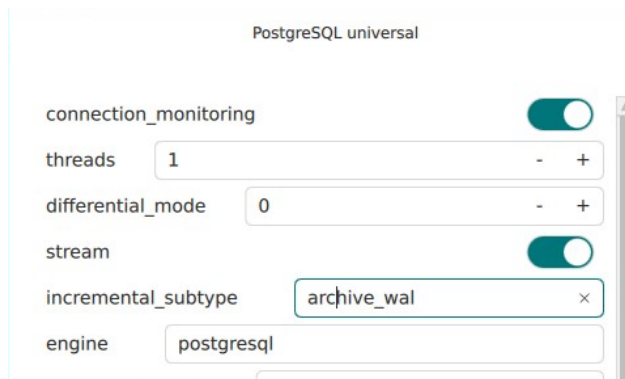
Тип ресурса  ...

Ресурс \*  x ...

Рисунок 4

Нажмите на «...» справа от Типа ресурса и укажите следующие параметры (рисунок 5):

- Подтип «Инкрементальное резервное копирование» (incremental\_subtype) — archive\_wal.
- Движок (Engine) — postgresql.



PostgreSQL universal

connection\_monitoring

threads 1 - +

differential\_mode 0 - +

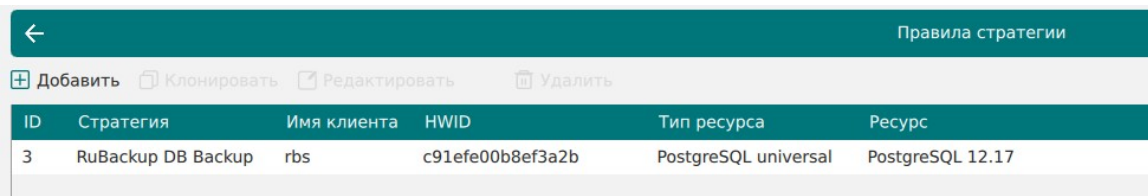
stream

incremental\_subtype archive\_wal x

engine postgresql

Рисунок 5

После этого нажмите на кнопку «ОК» и «Применить» - правило стратегии будет создано (рисунок 6):



ID	Стратегия	Имя клиента	HWID	Тип ресурса	Ресурс
3	RuBackup DB Backup	rbs	c91efe00b8ef3a2b	PostgreSQL universal	PostgreSQL 12.17

Рисунок 6

После того, как вы запустили в работу правило или стратегию резервного копирования базы данных RuBackup, все необходимые процедуры будут выполняться автоматически. Также рекомендуется настроить уведомления в адрес системного администратора RuBackup об успешном и неудачном выполнении процедур резервного копирования.

# Восстановление базы данных

## RuBackup

### Восстановление функционирующей базы данных

Если база данных RuBackup функционирует, выполните следующие шаги:

1. С помощью Менеджера администратора RuBackup (RBM), Менеджера клиента (RBC) или утилиты `rb_archives` восстановите без развертывания цепочку резервных копий в определенный каталог (см. разделы Использование Менеджера администратора RuBackup (RBM) для восстановления резервных копий СУБД PostgreSQL, Использование Менеджера клиента RuBackup (RBC) для восстановления резервных копий СУБД PostgreSQL, Использование `rb_archives` для восстановления резервной копии СУБД PostgreSQL);

2. Остановите все серверы RuBackup (основной, резервный и медиасерверы) командой:

```
sudo systemctl stop rubackup_server
```

3. Остановите базу данных командой:

```
sudo systemctl stop postgresql
```

4. Восстановите базу данных в ручном режиме (Point in time recovery (PITR)):

1. Удалите файлы поврежденной базы данных с помощью команды:

```
rm -rf /var/lib/postgresql/12/main/*
```

```
rm -rf /opt/rubakup/mnt/postgresql_acrhives/*
```

2. Переместите файлы из папки, в которую была восстановлена резервная копия (пункт 1) с помощью команд:

```
cd /restore_dir
```

```
mv var/lib/postgresql/12/main/* /var/lib/postgresql/12/main/
```

```
mv opt/rubakup/mnt/postgresql_acrhives/*  
/opt/rubakup/mnt/postgresql_acrhives/
```

3. Смените владельца и группу для восстановленных файлов:



```
chown -R postgres: /var/lib/postgresql/12/main/
```

```
chown -R postgres: /opt/rubackup/mnt/postgresql_archives/
```

4. Запустите базу данных командой:

```
systemctl start postgresql
```

5. Запустите в работу все серверы RuBackup командой:

```
sudo systemctl start rubackup_server
```

**Важно! Настоящее руководство является описанием функционала и не является точной инструкцией во восстановлению СУБД в любой ситуации, которая может произойти!**

В случае, когда требуется восстановление на определенный момент времени или на определенную транзакцию, добавьте в зависимости от версии PostgreSQL необходимую метку (`recovery_target_time`) в конфигурационный файл `postgresql.conf` (версии PostgreSQL  $\geq 12$ ) или в файл `recovery.conf` в соответствии с документацией PostgreSQL требуемой версии, например: <https://postgrespro.ru/docs/postgrespro/12/continuous-archiving#BACKUP-PITR-RECOVERY>.

Конкретная точка восстановления должна быть установлена в соответствии с <https://postgrespro.ru/docs/postgrespro/12/runtime-config-wal#RUNTIME-CONFIG-WAL-RECOVERY-TARGET>.

## Восстановление поврежденной базы данных

В том случае, если база данных RuBackup повреждена и серверы RuBackup не могут с ней работать, выполните следующее:

1. Заново настройте RuBackup с помощью утилиты `rb_init` (подробнее см. документ «Руководство по установке и обновлению серверов резервного копирования и Linux-клиентов RuBackup»);
2. Импортируйте резервные копии базы данных RuBackup с помощью утилиты `rb_inventory`. Воспользуйтесь командой:  

```
rb_inventory -vi /path_to_pool
```
3. Выполните шаги 1 — 5 из раздела Восстановление функционирующей базы данных.

Перед процедурой восстановления базы данных RuBackup рекомендуется ознакомиться с официальной документацией по восстановлению базы данных PostgreSQL.

## Использование Менеджера администратора RuBackup (RBM) для восстановления резервных копий СУБД PostgreSQL

Для восстановления резервной копии с помощью RBM зайдите в раздел «Репозиторий» и выберите значок (либо пункт контекстного меню) «Восстановить» (рисунок 7).

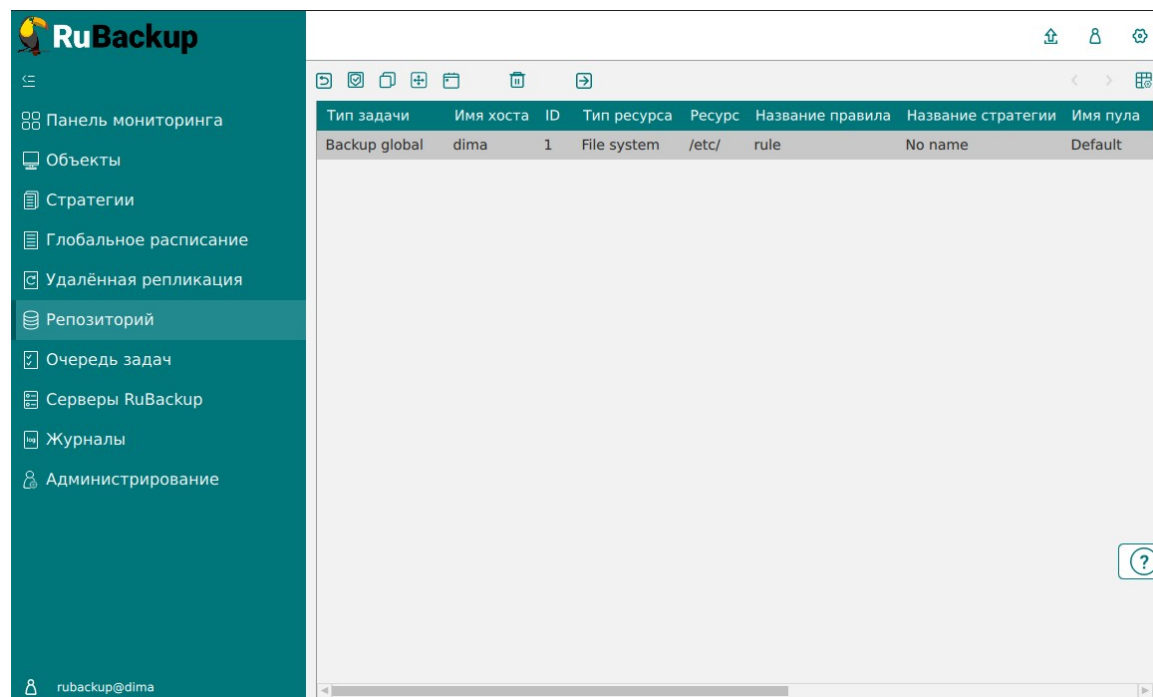


Рисунок 7

Выберите клиента для восстановления, каталог распаковки и при необходимости сдвиньте переключатель «Восстановить на целевом ресурсе» (рисунок 8). При необходимости воспользуйтесь гранулярным восстановлением. Нажмите «Применить».

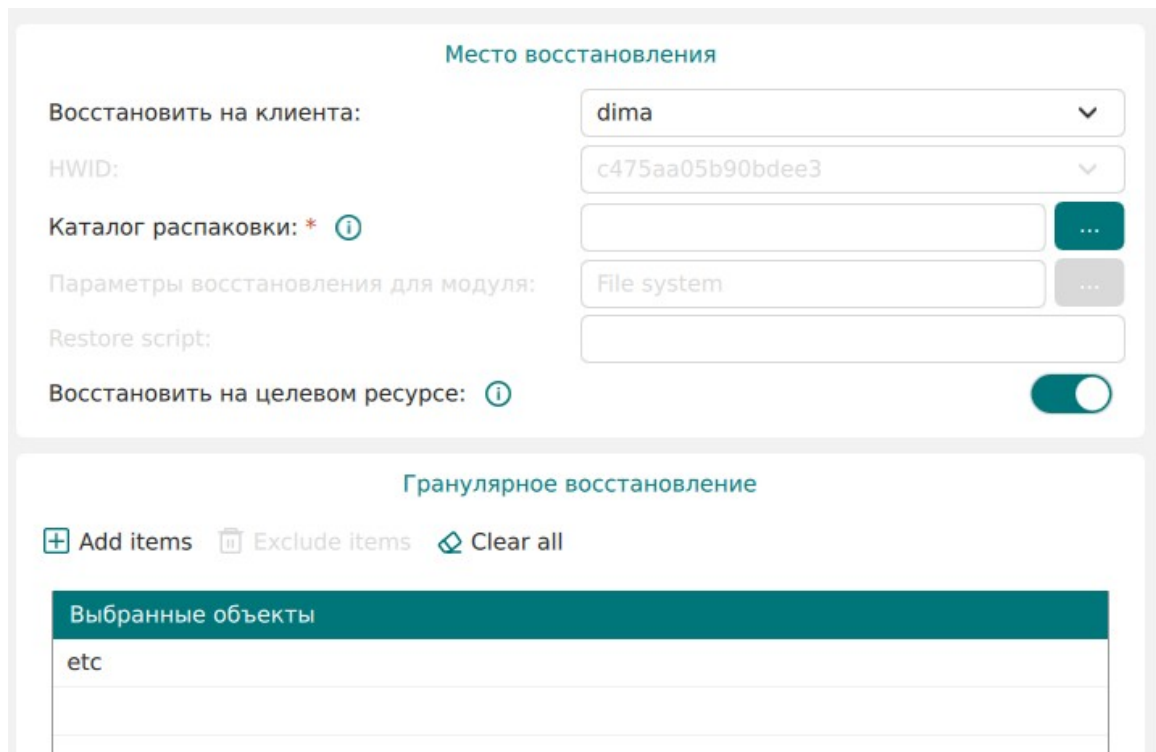


Рисунок 8

Статус задачи восстановления можно посмотреть в разделе «Очередь задач» (рисунок 9). При успешном завершении восстановления задача переходит в статус «Done».

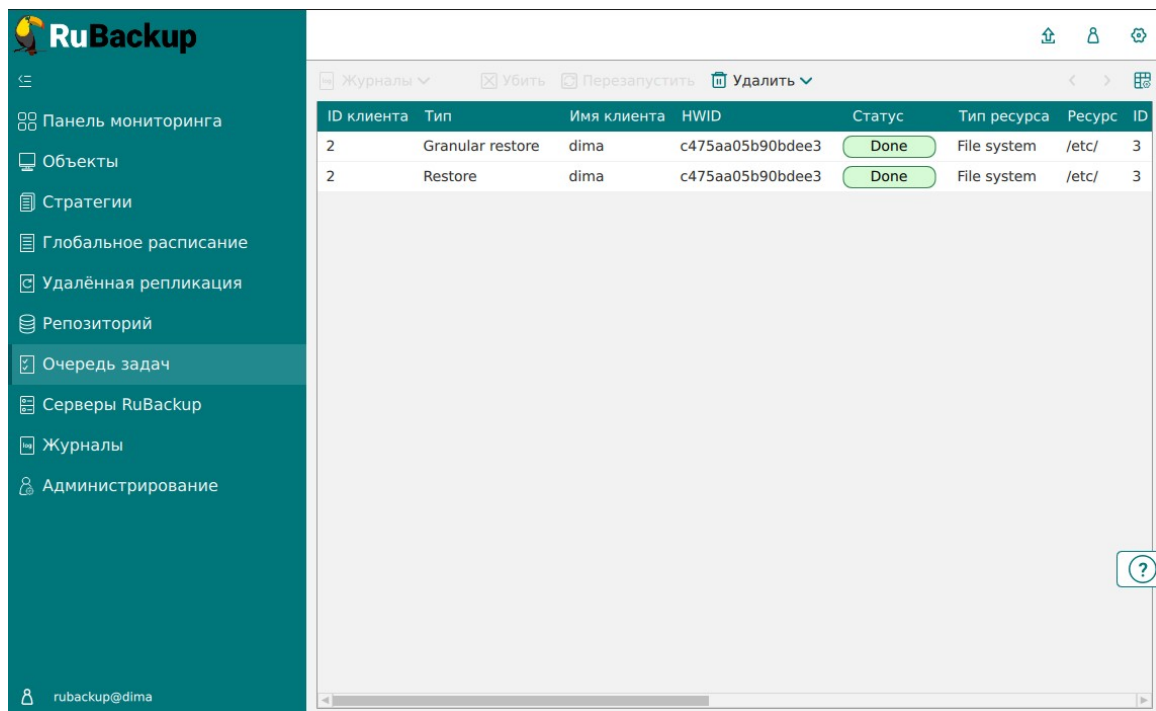


Рисунок 9

## Использование Менеджера клиента RuBackup (RBC) для восстановления резервных копий СУБД PostgreSQL

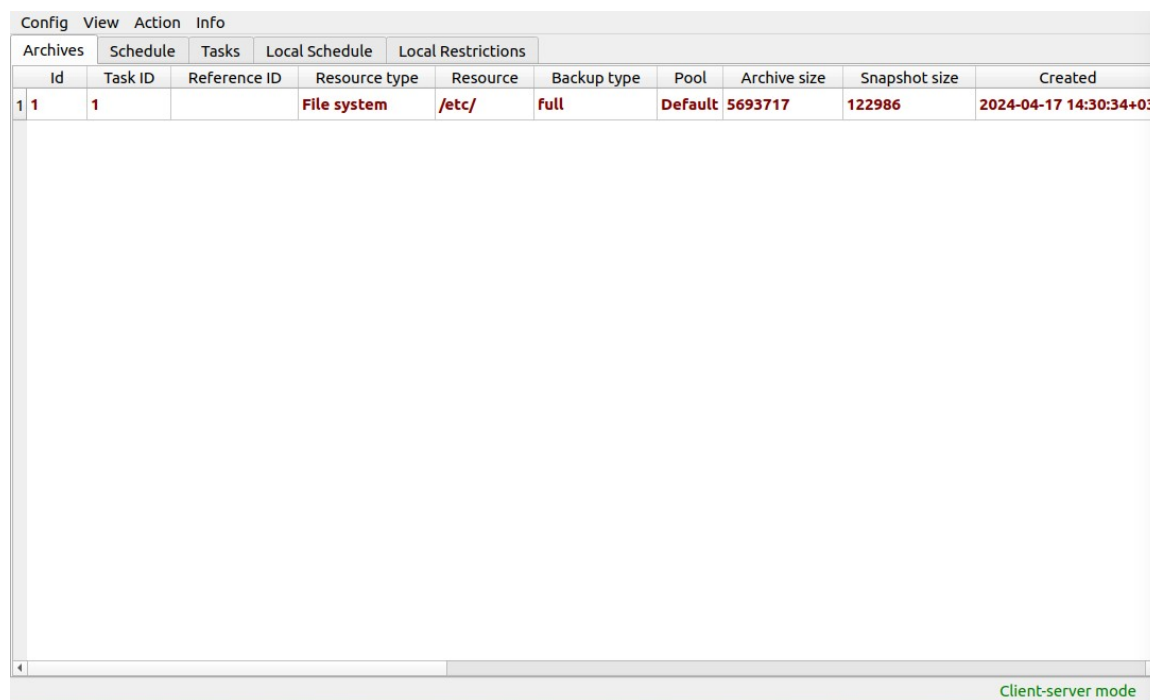
В том случае, если на хосте клиента не установлены графические пакеты, рекомендуется использовать утилиты командной строки. Если вы все-таки хотите использовать RBC для управления, вам может понадобиться установка на хост клиента нескольких пакетов, которые обеспечат возможность запуска RBC (см. документ «Руководство по установке серверов резервного копирования и Linux-клиентов RuBackup», раздел «Пакеты для ОС без графической оболочки»).

Для использования Менеджера клиента (RBC) необходимо выполнить следующие действия:

1. Запустите RBC при помощи команды:

```
# rbc
```

При этом откроется окно (рисунок 10):



Config View Action Info										
Archives		Schedule		Tasks		Local Schedule		Local Restrictions		
Id	Task ID	Reference ID	Resource type	Resource	Backup type	Pool	Archive size	Snapshot size	Created	
1	1		File system	/etc/	full	Default	5693717	122986	2024-04-17 14:30:34+03	

Рисунок 10

2. Правой кнопкой мыши вызовите контекстное меню и выберите «Восстановить» (рисунок 11):

Конфигурация		Вид		Действия		Информация			
Резервные копии		Глобальное расписание		Задачи		Локальное расписание		Ограничения	
Id	Задача	Ссылка	Тип ресурса	Ресурс	Тип РК	Пул	Размер РК	Размер снимка	Создано
13	4			/home	full	Default	74570883	200702	2023-02-15 13:05:44+03

- Срочное РК
- Восстановить
- Удалить
- Перезагрузить
- Проверить

Рисунок 11

- Для восстановления резервной копии введите пароль клиента (задается при первом использовании rb\_archives или RBC со стороны клиента. В том случае если вы не знаете пароль, обратитесь к системному администратору СРК чтобы сбросить его и задать заново) (рисунок 12):

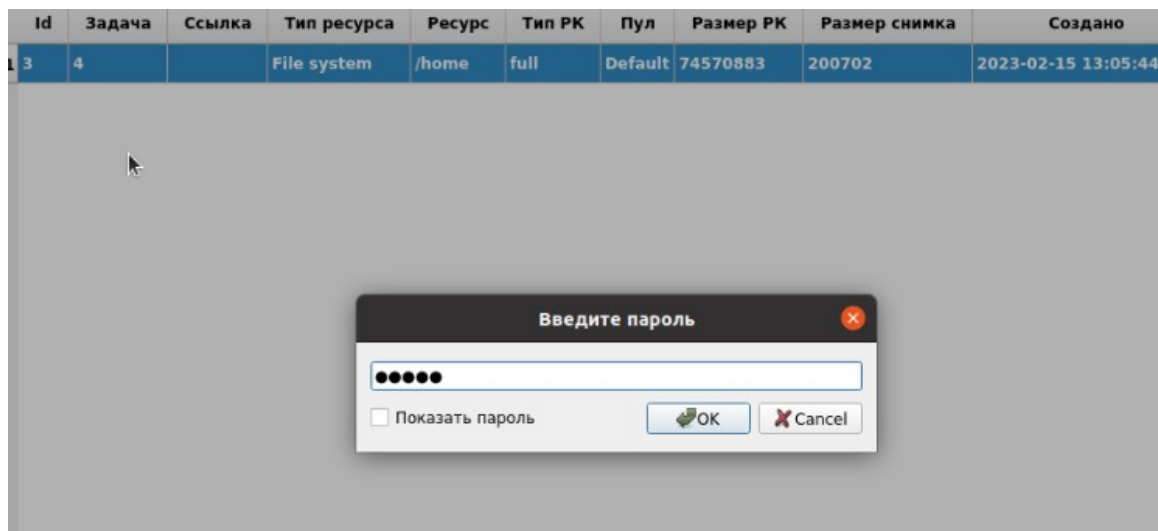


Рисунок 12

- В том случае, если резервная копия не проверена, подтвердите что ее действительно нужно восстанавливать (рисунок 13):

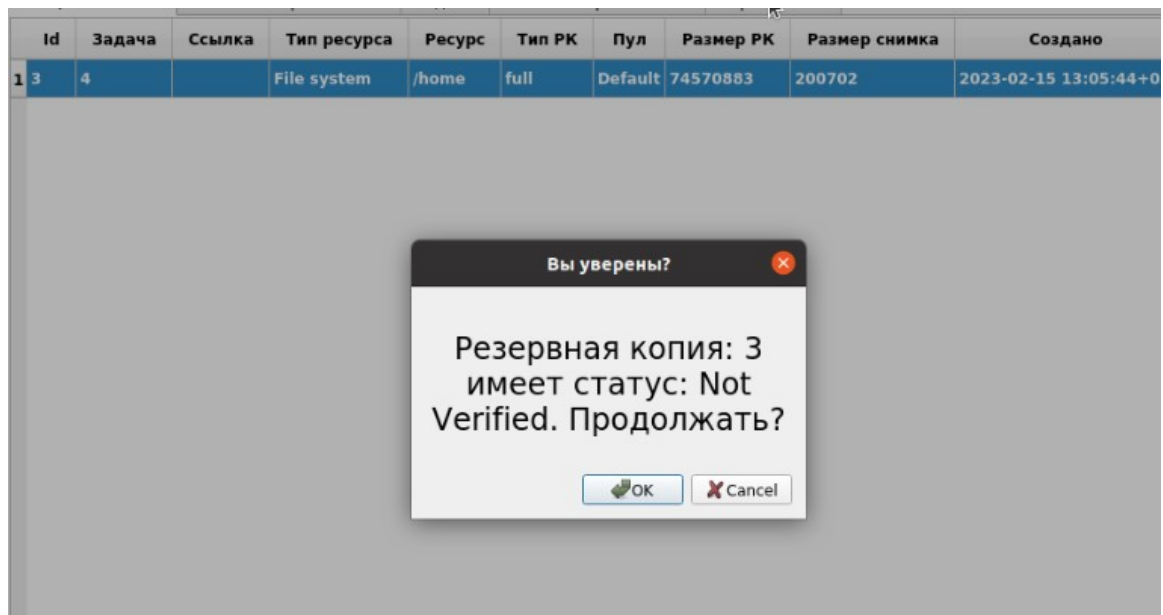


Рисунок 13

5. Выберите файл или каталог для восстановления (рисунок 14):

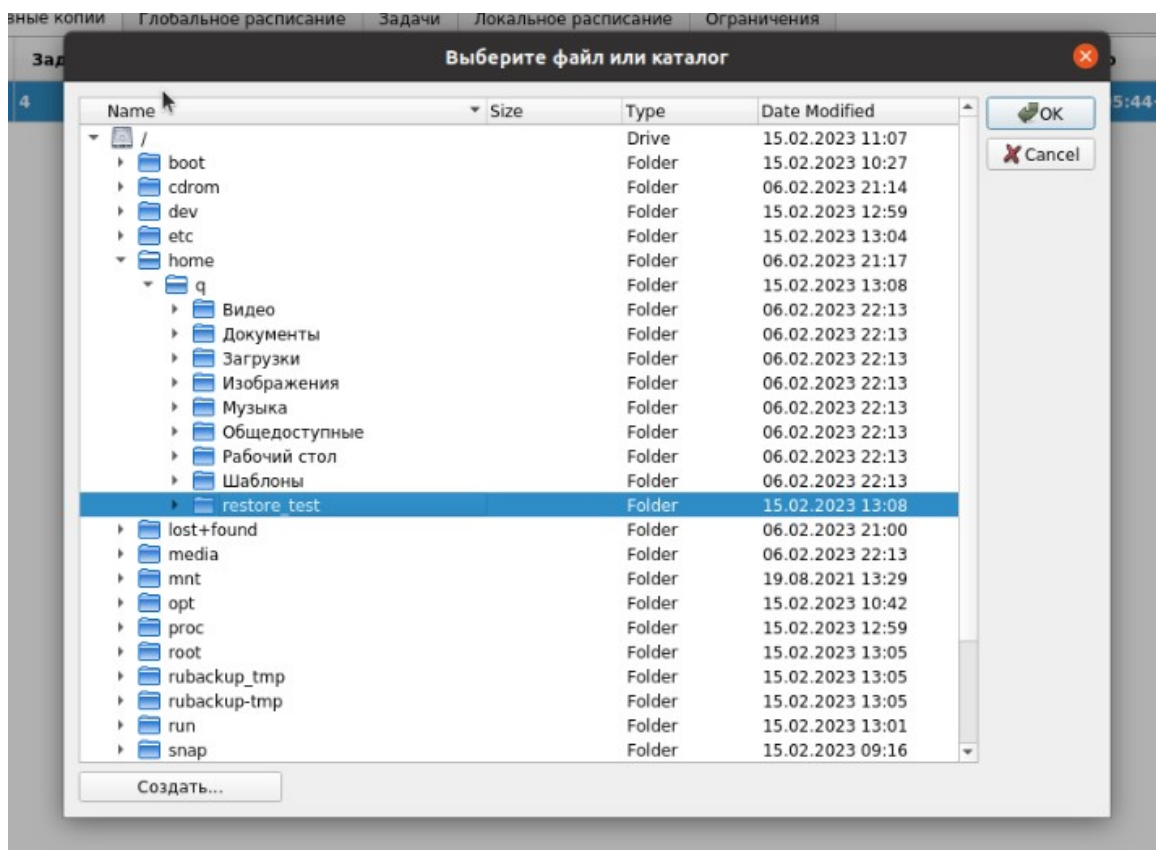


Рисунок 14

6. После выбора каталога для восстановления в системе резервного копирования будет создана задача (рисунок 15):

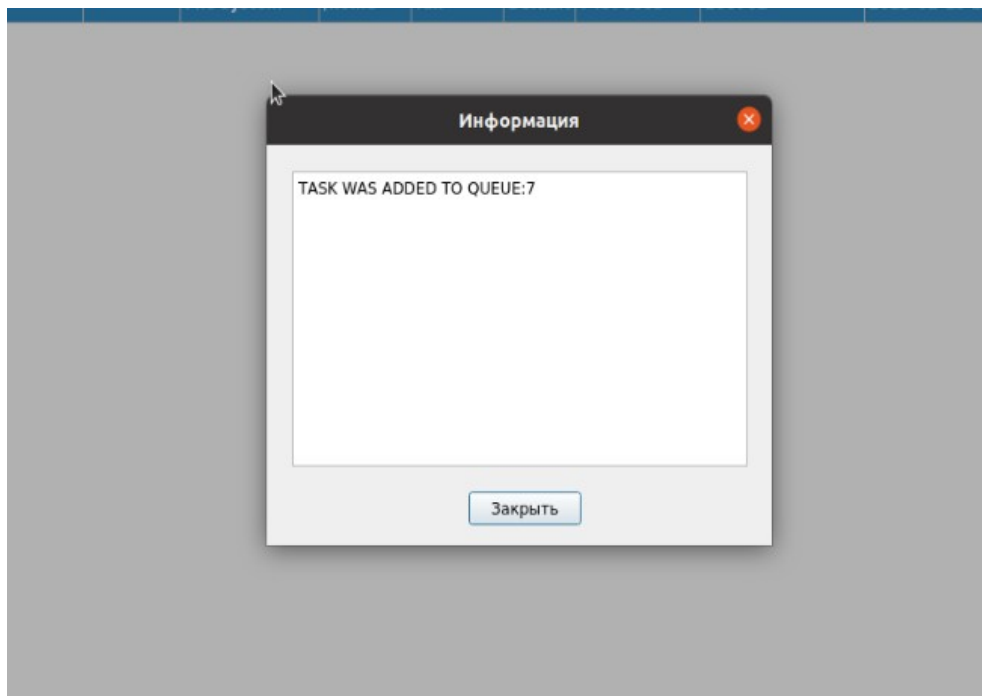
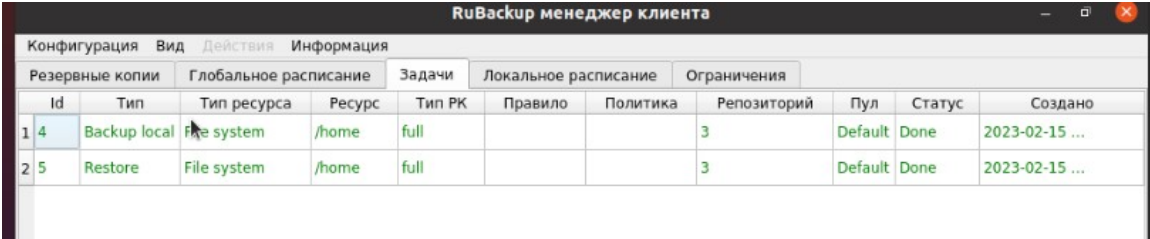


Рисунок 15

7. Проконтролируйте ход выполнения задачи восстановления (рисунок 16):



RuBackup менеджер клиента										
Резервные копии		Глобальное расписание		Задачи		Локальное расписание		Ограничения		
Id	Тип	Тип ресурса	Ресурс	Тип РК	Правило	Политика	Репозиторий	Пул	Статус	Создано
1	4	Backup local	File system	/home	full		3	Default	Done	2023-02-15 ...
2	5	Restore	File system	/home	full		3	Default	Done	2023-02-15 ...

Рисунок 16

При успешном завершении восстановления задача переходит в статус «Done».

## Использование `rb_archives` для восстановления резервной копии СУБД PostgreSQL

Использование утилиты командной строки `rb_archives` позволяет посмотреть список резервных копий:

```
root@postgresql:~# rb_archives
Id | Ref ID | Resource           | Resource type   | Backup type | Created                | Crypto | Signed | Status
-----+-----+-----+-----+-----+-----+-----+-----+-----
1  |        | PostgreSQL 12.13  | PostgreSQL universal | full       | 2023-02-16 11:11:03+03 | nocrypt | True  | Not Verified
root@postgresql:~#
```

В первой колонке указаны идентификаторы резервных копий (ID). Чтобы восстановить резервную копию без развертывания, нужно использовать команду

```
# rb_archives -X ID -d /path_to_restore
```

Опция `-X` указывает, что нужно выполнить операцию восстановления без развертывания

Опция `-d` указывает путь, в который нужно восстановить резервную копию. Если не используется опция `-d`, резервная копия будет восстановлена в каталог для временных операций с резервными копиями. В случае восстановления резервной копии без развертывания всегда рекомендуется использовать опцию `-d` с указанием каталога на клиенте, в котором есть достаточно места для восстановления резервной копии.

Для восстановления резервной копии необходимо ввести пароль клиента (задается при первом использовании `rb_archives` или `RBC` со стороны клиента. В том случае если вы не знаете пароль, обратитесь к системному администратору СРК, чтобы сбросить его и задать заново).

Проконтролировать выполнение задачи восстановления можно при помощи утилиты командной строки `rb_tasks`:

```
root@postgresql:~# rb_tasks
Id | Task type | Resource           | Backup type | Status | Created
-----+-----+-----+-----+-----+-----
1  | Backup global | PostgreSQL 12.13  | full       | Done  | 2023-02-16 11:10:42+03
2  | Restore      | PostgreSQL 12.13  | full       | Done  | 2023-02-16 11:21:20+03
root@postgresql:~#
```



Так же можно получить детальную информацию о ходе восстановления из журнального файла задачи, находящегося в папке /opt/rubackup/log/:

```
root@ubuntu-server:~# cat /opt/rubackup/log/task_2.log
Wed Feb 15 12:30:18 2023: Media server q1 has 'New' task in the queue. Task ID: 2. Task type: Backup global
Wed Feb 15 12:30:18 2023: Task ID: 2. New status: Assigned
Wed Feb 15 12:30:18 2023: Task ID: 2. New status: At_Client
Wed Feb 15 12:30:18 2023: Task ID: 2. New status: Execution
Wed Feb 15 12:30:19 2023: Set unlimited bandwidth for task ID: 2
Wed Feb 15 12:30:22 2023: Task ID: 2. New status: Start_Transfer
Wed Feb 15 12:30:22 2023: Set unlimited bandwidth for task ID: 2
Wed Feb 15 12:30:23 2023: Transfer of snapshot client2_TaskID_2_NORuleOrStrategy_0_D2023_2_15H09_30_18_BackupType_1_ResourceType_11 has succeeded. Task ID: 2
Wed Feb 15 12:30:23 2023: Task ID: 2. New record ID was created in repository: 2
Wed Feb 15 12:30:23 2023: Task ID: 2. New status: Transmission
Wed Feb 15 12:30:24 2023: Task ID: 2. New status: Done
Thu Feb 16 11:21:20 2023: Media server ubuntu-server has 'New' task in the queue. Task ID: 2. Task type: Restore
Thu Feb 16 11:21:20 2023: Task ID: 2. New status: Assigned
Thu Feb 16 11:21:21 2023: Task ID: 2. New status: At_Client
Thu Feb 16 11:21:21 2023: Task ID: 2. New status: Start_Transfer
Thu Feb 16 11:21:21 2023: Task ID: 2. New status: Transmission
Thu Feb 16 11:21:21 2023: Set unlimited bandwidth for task ID: 2
Thu Feb 16 11:21:24 2023: Blocks are ready, time: 2
Thu Feb 16 11:21:26 2023: Task ID: 2. New status: Done
```