

Система резервного копирования и восстановления данных

POSTGRES PRO

ВЕРСИЯ 2.5.0, 02.07.2025

Содержание

Подготовка хоста с СУБД Postgres Pro
Установка клиента RuBackup
Установка пакетов модулей резервного копирования
Конфигурационный файл модуля
Подготовка к использованию pg_probackup
Для инициализации каталога резервных копий используйте следующую
команду:6
Определение копируемого экземпляра
Настройка копируемого кластера баз данных для использования
pg_probackup
Настройка потокового резервного копирования
Настройка непрерывного архивирования WAL
Настройка копирования в режиме PTRACK
Завершение настройки кластера
Мастер-ключ
Удаление клиента RuBackup
Режим базового резервного копирования кластера Postgres Pro
Режим дифференциального резервного копирования кластера Postgres Pro 18
В режиме DELTA
В режиме РАGE
В режиме РТRACK
Режим восстановления резервной копии кластера Postgres Pro
Менеджер администратора RuBackup (RBM)
Менеджер клиента RuBackup (RBC)
Вкладка «Резервные копии»
Вкладка «Глобальное расписание»
Вкладка «Задачи»
Вкладка «Локальное расписание»
Вкладка «Ограничения»
Утилиты командной строки клиента RuBackup
Восстановление резервной копии кластера Postgres Pro
Восстановление резервной копии в RBC
Восстановление при помощи утилиты rb_archives

Система резервного копирования (СРК) RuBackup поддерживает резервное копирование кластеров СУБД Postgres Pro Standart и Enterprise версий 11, 12, 13, 14, 15, 16.

Принцип резервного копирования кластеров Postgres Pro с использованием RuBackup состоит в периодическом создании базовых резервных копий экземпляра СУБД по определённому расписанию.

Полное резервное копирование – это создание резервной копии всех данных из исходного набора, независимо от того, изменялись данные или нет с момента выполнения последней полной резервной копии.

Дифференциальное резервное копирование сохраняет только данные, изменённые со времени выполнения предыдущего полного резервного копирования.

Инкрементальное резервное копирование сохраняет только данные, изменённые со времени выполнения предыдущей инкрементальной резервной копии, а если такой нет, то со времени выполнения последней полной резервной копии.

В репозитории RuBackup базовые резервные копии будут храниться как **полные резервные копии** (full), а файлы backup.control и backup_content.control, созданные после базовой резервной копии, – как **snap-файл** с расширением «.snap».На основе файлов backup.control и backup_content.control из snap-файла полной резервной копии создаётся **дифференциальная резервная копия** (differential).

После успешного выполнения резервного копирования следует производить периодическую очистку каталога хранения архивных файлов WAL. Автоматическая очистка не предусмотрена.

После окончания операции резервного копирования будут созданы два файла – архивный и снимок состояния – на медиасервере, 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, так как в нём располагается только информация о ресурсе, о режиме, в котором была сделана резервная копия и время старта и окончания резервного копирования. В снимке состояния отсутствуют конфиденциальные данные.

Для выполнения резервного копирования кластеров Postgres Pro 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 (параметр use-local-backupdirectory). 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 с СУБД Postgres Pro

Для возможности резервного копирования данных кластера Postgres Pro при помощи CPK RuBackup на сервер следует установить следующие пакеты:

- rubackup-client.deb клиент резервного копирования;
- rubackup-common.deb общий пакет для CPK RuBackup;
- rubackup-postgres-pro.deb модуль резервного копирования данных Postgres Pro;
- pg-probackup-std-13.deb консольная утилита для создания резервных копий кластеров баз данных PostgreSQL и их восстановления;
- python3-psycopg2 драйвер PostgreSQL, совместимый с DB API 2.0.

Установка клиента RuBackup

Для осуществления резервного копирования и восстановления данных кластера Postgres Pro 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 кластеров СУБД Postgres Pro клиент RuBackup должен работать от имени суперпользователя (root в Linux и Unix).

Соответственно, для версий Postgres Pro 10 и ниже необходимо, чтобы сервер Postgres Pro был запущен от имени root.

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 с СУБД Postgres Pro после установки на него клиента RuBackup.

Для установки пакета модулей используйте следующую команду:

dpkg **-i** rubackup-postgres-pro.deb

Конфигурационный файл модуля

Для работы модуля необходимо произвести ручную настройку конфигурационного файла. Формат конфигурационного файла - **YAML**.

Конфигурационный файл модуля располагается по пути /opt/rubackup/etc/rb_module_postgres_pro.conf.

Содержание конфигурационного файла:

- restore_target_action: возможные значения (PAUSE, PROMOTE, SHUTDOWN). Задаёт действие, которое должен выполнить сервер по достижении цели восстановления.
 - По умолчанию установлено значение PAUSE, при котором после восстановления кластер находится в состоянии Read Only. Для того, чтобы кластер после восстановления был в режиме Read-Write нужно установить значение PROMOTE.
- restore_target: возможные значения (IMMEDIATE, LATEST).
 - LATEST восстановить последнее возможное состояние, исходя из содержимого архива WAL.
 - IMMEDIATE восстановить самое раннее из возможных согласованное состояние кластера;
- restore_mode: возможные значения (NONE, CHECKSUM). Если значение параметра равно CHECKSUM, то pg_probackup при восстановлении будет повторно использовать валидные страницы доступные в каталоге кластера, если они не изменялись.
- pg_probackup: Абсолютный путь до утилиты pg_probackup.



Перед началом работы с модулем обязательно нужно указать актуальный полный путь до утилиты pg_probackup.

Подготовка к использованию pg_probackup

Для выполнения резервного копирования кластера Postgres Pro используется утилита pg_probackup. Перед использованием модуля необходимо выполнить следующие действия:

- Инициализировать каталог резервных копий;
- Добавить копируемый экземпляр в каталог копий;
- Настроить копируемый кластер баз данных для использования pg_probackup.

Для инициализации каталога резервных копий используйте следующую команду:

pg_probackup init -B /opt/rubackup/mnt/pg_probackup

He изменяйте данную команду. Если директория /opt/rubackup/mnt/pg_probackup уже существует, то она должна быть пустой.

После выполнения данной команды утилита pg_probackup создаст подкаталоги в каталоге резервных копий /opt/rubackup/mnt/pg_probackup:

wal

каталог для файлов WAL

backups

каталог для файлов резервных копий.

Определение копируемого экземпляра

Утилита pg_probackup может сохранять резервные копии разных кластеров баз данных в одном каталоге резервных копий. Для создания необходимых подкаталогов вы должны добавить копируемый экземпляр в каталоге копий для каждого кластера баз данных, копию которого вы будете делать.

Для добавления копируемого экземпляра при помощи pg_probackup выполните команду:

pg_probackup add-instance **-B** /opt/rubackup/mnt/pg_probackup **-D** каталог_данных --instance имя_экземпляра (по названию каталога данных)

каталог_данных

это каталог, в котором хранятся все данные кластера баз данных. Например, для кластера postgres СУБД Postgres Pro 13 каталогом данных будет /var/lib/pgpro/std-13/data/.



Имя экземпляра должно совпадать с именем каталога данных. Например, добавляется каталог данных /var/lib/pgpro/std-13/data/, значит именем экземпляра будет data.



Директория /opt/rubackup/mnt/pg_probackup и все вложенные в неё папки должны быть доступны для записи и чтения пользователю

postgres, а также пользователю, под контролем которого работает клиент RuBackup.

Обеспечить доступ можно следующим образом:

```
sudo chgrp postgres -R /opt/rubackup/mnt/pg_probackup
sudo chmod g+rwx -R /opt/rubackup/mnt/pg_probackup
```

Если планируется использование модуля для резервного копирования кластеров Postgres Pro в составе patroni, то необходимо выполнить следующую команду:

```
pg_probackup set-config -B /opt/rubackup/mnt/pg_probackup/ --instance=patroni
--pghost ip_кластера
```

ір_кластера

ір адрес копируемого инстанса patroni.

Настройка копируемого кластера баз данных для использования pg_probackup

Для выполнения резервного копирования в защищённом режиме необходимо создать роль с ограниченными правами и базу данных резервного копирования. Имя роли, пароля и базы данных условны и могут быть изменены по усмотрению пользователя. Последовательность действий следующая:

 Сначала создайте базу данных резервного копирования. Данная операция производится от имени пользователя postgres:

```
su postgres
createdb backupdb
```

- Далее описан процесс создания роли для выполнения резервного копирования.
 В целях обеспечения безопасности копируемых данных, создаваемая роль будет обладать минимальными правами, необходимыми для выполнения резервного копирования экземпляра Postgres Pro. В этом примере такой ролью будет rubackup_backuper.
 - Выполните подключение к базе данных backupdb от имени пользователя postgres:

sudo -u postgres psql -d backupdb

 Далее, в psql создайте роль rubackup_backuper (имя пользователя может быть изменено) и задайте пароль (в качестве пароля укажите желаемый пароль вместо 12345). Затем, при помощи приведённого ниже скрипта, определите следующие разрешения на сервере Postgres Pro (только в базе данных, к которой производится подключение).

Для Postgres Pro версии 14 и ниже:

BEGIN: CREATE ROLE rubackup_backuper WITH LOGIN; ALTER USER rubackup_backuper WITH PASSWORD '12345'; GRANT USAGE ON SCHEMA pg_catalog TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.current_setting(text) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.set_config(text, text, boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_is_in_recovery() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_start_backup(text, boolean, boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_stop_backup(boolean, boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_create_restore_point(text) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_switch_wal() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_last_wal_replay_lsn() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.txid_current() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.txid_current_snapshot() TO rubackup backuper: GRANT EXECUTE ON FUNCTION pg_catalog.txid_snapshot_xmax(txid_snapshot) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_control_checkpoint() TO rubackup_backuper; ALTER ROLE rubackup_backuper WITH REPLICATION; COMMIT

Для Postgres Pro 15:

```
BEGIN;
CREATE ROLE rubackup_backuper WITH LOGIN;
ALTER USER rubackup_backuper WITH PASSWORD '12345';
GRANT USAGE ON SCHEMA pg_catalog TO rubackup_backuper;
GRANT EXECUTE ON FUNCTION pg_catalog.current_setting(text) TO
```

rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.set_config(text, text, boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_is_in_recovery() TO rubackup backuper: GRANT EXECUTE ON FUNCTION pg_catalog.pg_backup_start(text, boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_backup_stop(boolean) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_create_restore_point(text) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_switch_wal() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_last_wal_replay_lsn() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.txid_current() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.txid_current_snapshot() TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.txid_snapshot_xmax(txid_snapshot) TO rubackup_backuper; GRANT EXECUTE ON FUNCTION pg_catalog.pg_control_checkpoint() TO rubackup_backuper; ALTER ROLE rubackup_backuper WITH REPLICATION; COMMIT

• Создайте файл .pgpass в каталоге /root. В файле /root/.pgpass необходимо указать данные для подключения к ранее созданной базе данных и для репликации.

Это распространённый способ хранения информации о соединении в PostgreSQL вместо ввода пароля при каждой совершённой операцией с pg_probackup.

Этот файл должен содержать строки в таком формате:

сервер:порт:база_данных:имя_пользователя:пароль

Пример:

localhost:5432:backupdb:rubackup_backuper:12345 localhost:5432:replication:rubackup_backuper:12345

Файл .pgpass обязательно должен находиться в домашнем каталоге суперпользователя /root. Не меняйте местами информацию в строке, она должна быть указана как в примере. Также, маска разрешений файла должна соответствовать маске 0600. Если одно из этих условий будет

нарушено, то выполнение резервной копии будет прервано ошибкой.

• Далее необходимо произвести изменения в файле pg_hba.conf. Найти конфигурационный файл, относящийся к настраиваемому кластеру, можно так:

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

Вызовите psql при помощи команды:

sudo -u postgres psql

Если для пользователя postgres не установлен пароль, установите его изменив 12345 на подходящий:

```
alter user postgres with password '12345'
```

В конец файла pg_hba.conf добавьте следующие строки:

host	backupdb	rubackup_backuper	127.0.0.1/32	md5	
host	replication	rubackup_backuper	127.0.0.1/32	md5	

Вместо peer везде установите md5. Пример итогового файла:

# TYPE	DATABASE	USER	ADDRESS		METHOD
# "local	l" is for Unix do	omain socket conn	ections	only	
local	all	all			md5
# IPv4 l	local connections				
host	all	all	127.0.0.	1/32	md5
# IPv6 l	local connections				
host	all	all	::1/128		md5
# Allow	replication conn	nections from loc	alhost,	by a user with	the
# replic	ation privilege.				
local	replication	all			md5
host	replication	all	127.0.0.	1/32	md5
host	replication	all	::1/128		md5
host	backupdb	rubackup backupe	r	127.0.0.1/32	md5
host	replication	rubackup_backupe	г	127.0.0.1/32	md5

Проверяем не было ли опечаток и перечитываем конфигурацию:

psql - psql -	-c 'se] -c 'se]	lect * f Lect pg_	rom pg_hba_f reload_conf(ile_rul)'	les'			
				<i>6</i> :11				
розідгез@розі Пароль пользон	grespro-cli вателя post	gres:	c select ^ rrom pg_nba	a_rile_rules				
line_number	type	database	user_name	address	netmask	auth_method	options	еггог
80	++ local {	all}	+			+ md5	+ 	
82	host {	all}	{all}	127.0.0.1	255.255.255.255	md5		i i
84	host {	all}	{all}	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:	md5		í i
87	local {	replication}	{all}			md5		1 1
88	host {	replication}	{all}	127.0.0.1	255.255.255.255	md5		l i
89	host {	replication}	{all}	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff	md5		1 ,
90	host {	backupdb}	{rubackup_backuper}	127.0.0.1	255.255.255.255	md5		1 ,
91	host {	replication}	{rubackup_backuper}	127.0.0.1	255.255.255.255	md5		i ,
(8 строк)								
		ont, Cocol	s isolast as soland sor					
	Batend Dost	ares:	c select pg_leload_col	10				
napone noneson na reload com	nf	gres.						
t								
(1 строка)								

После реализации данных настроек модуль сможет выполнять полное резервное копирование и дифференциальное резервное копирование в режиме **DELTA** используя режим доставки WAL по умолчанию (ARCHIVE).

Настройка потокового резервного копирования

Для настройки потокового резервного копирования (STREAM) требуется произвести изменения в файле postgresql.conf, который находится внутри копируемого кластера. Например, директорией кластера СУБД Postgres Pro 13 является /var/lib/pgpro/std-13/data/. Обратите внимание на то, что расположение файла может отличаться в зависимости от дистрибутива Linux, версии Postgres Pro и кластера баз данных.

Выполните следующие действия:

- установите для параметра max_wal_senders достаточно большое значение, предусматривающее минимум одно подключение для процесса резервного копирования;
- задайте для параметра wal_level значение выше minimal.

Если вы не планируете производить дальнейшую настройку, то после внесённых изменений в файл postgresql.conf необходимо перезагрузить сервер Postgres Pro при помощи команды:

sudo systemctl restart postgrespro-std-13.service

После реализации данных настроек модуль сможет выполнять **полное** резервное копирование и **дифференциальное** резервное копирование в режимах **DELTA** используя потоковую доставку WAL (STREAM).

Настройка непрерывного архивирования WAL

Для выполнения копирования в режиме **PAGE** и восстановления резервных копий на момент времени (recovery-target) должно осуществляться непрерывное архивирование WAL.

Чтобы настроить непрерывное архивирование, выполните следующие действия:

- задайте для параметра wal_level значение выше minimal.
- если вы настраиваете резервное копирование на ведущем сервере, параметр archive_mode должен иметь значение on или always. Для выполнения резервного копирования на ведомом требуется значение always.
- установите параметр archive_command:

archive_command = '/путь_инсталляции/pg_probackup archive-push -B /opt/rubackup/mnt/pg_probackup --instance имя_экземпляра --wal-file-path %p --wal-file-name %f [параметры_удалённого_режима]'

путь_инсталляции

путь к каталогу установленной версии pg_probackup, которую вы хотите использовать

имя_экземпляра

должно указывать на уже проинициализированный для данного кластера БД копируемый экземпляр

параметры_удалённого_режима

должны задаваться только в случае расположения архива WAL в удалённой системе.

0

Если вы планируете выполнять страничное копирование и/или делать копии с ведомого сервера, используя режим доставки **WAL ARCHIVE**, при недостаточной транзакционной активности может потребоваться долго ждать заполнения очередного сегмента WAL.

Чтобы ограничить время ожидания, вы можете воспользоваться параметром archive_timeout на ведущем сервере.

Значение этого параметра должно быть меньше значения --archive -timeout (по умолчанию 5 минут), чтобы заполненный сегмент успел передаться ведомому серверу и попасть в архив WAL, прежде чем копирование прервётся по тайм-ауту, заданному параметром --archive -timeout.

Если вы не планируете производить дальнейшую настройку, то после внесённых изменений в файл postgresql.conf необходимо перезагрузить сервер Postgres Pro

при помощи команды:

sudo systemctl restart postgrespro-std-13.service

После реализации данных настроек модуль сможет выполнять **дифференциаль**ное резервное копирование в режиме **PAGE** используя режимы доставки ARCHIVE и STREAM.

Настройка копирования в режиме PTRACK

Перед выполнением дифференциальной резервной копии в режиме PTRACK выполните следующие подготовительные действия:

• Зайдите от имени администратора БД в backupdb:

sudo -u postgres psql -d backupdb

и выполните запрос:

CREATE EXTENSION ptrack

- Далее перейдите к редактированию конфигурационного файла postgresql.conf:
 - задайте для параметра shared_preload_libraries значение ptrack:
 - добавьте в конец конфигурационного файла параметр ptrack.map_size и установите его значение по следующим правилам:

Для оптимальной производительности рекомендуется задавать ptrack.map_size равным N ÷ 1024, где N — объём кластера Postgres Pro в мегабайтах. Увеличивать значение ptrack.map_size сверх рекомендуемого не имеет большого практического смысла. Максимально допустимое значение — 1024.

- Если до этих изменений была сделана полная резервная копия, то после вступления изменений в силу необходимо сделать новую полную резервную копию, иначе дифференциальное резервное копирование в режиме РТRACK прервётся ошибкой.
- Выполните команду для перезапуска сервиса:

sudo systemctl restart postgrespro-std-13.service

После реализации данных настроек модуль сможет выполнять **дифференциальное** резервное копирование в режиме **PTRACK** используя режимы доставки ARCHIVE и STREAM.

Завершение настройки кластера

После выполнения подготовки целевого кластера к выполнению резервного копирования необходимо перезапустить клиента RuBackup:

```
rubackup_client stop
rubackup_client start
```

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

```
Try to check module: Postgres Pro ...
Execute OS command: /opt/rubackup/modules/rb_module_postgres_pro -t 2>&1
... module Postgres Pro was checked successfully
```

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.



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



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

```
hexdump /opt/rubackup/keys/master-key
0000000 79d1 4749 7335 e387 9f74 c67e 55a7 20ff
0000010 6284 54as 83a3 2053 4818 e183 1528 a343
0000020
```

Удаление клиента RuBackup

При необходимости вы можете удалить с хоста 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-postgres-pro-13 используйте команды:

```
apt remove rubackup-postgres-pro-13
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:

• Выполнить команду:

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

• Перезагрузить операционную систему

sudo init 6

Режим базового резервного копирования кластера Postgres Pro

В ходе базового резервного копирования выполняется взаимодействие с утилитой pg_probackup.

Команда на выполнение резервного копирования в режиме доставки WAL ARCHIVE :

pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance =имя_экземпляра --backup-mode=FULL -j 1 --pguser=rubackup_backuper --pgdatabase=backupdb --no-password

-j

количество потоков, в которые выполняется программа;

pguser

роль для резервного копирования кластера;

pgdata

база данных резервного копирования;

no-password

опция, исключающая пользовательский ввод пароля во время выполнения резервного копирования.

Команда на выполнение резервного копирования в режиме доставки WAL STREAM:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=FULL --stream -j 1 --pguser=rubackup_backuper
--pgdatabase=backupdb --no-password
```

Если планируется, что резервная копия будет сделана на одном сервере Postgres Pro, а восстанавливаться на другом, то необходимо, чтобы принимающий сервер имел такие же значения параметров block_size и wal_blocksize, что и на основном сервере и одинаковую основную версию. В зависимости от конфигурации кластера, Postgres Pro может накладывать дополнительные ограничения, например, по архитектуре процессора и версии libc/libicu.

Режим дифференциального резервного копирования кластера Postgres Pro

Выполнение дифференциального резервного копирования кластера Postgres Pro может производиться в трёх разных режимах:

В режиме DELTA

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL ARCHIVE:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=DELTA -j 1 --pguser=rubackup_backuper
--pgdatabase=backupdb --no-password
```

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL STREAM:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=DELTA --stream -j 1 --pguser=rubackup_backuper
--pgdatabase=backupdb --no-password
```

В режиме РАСЕ



Перед выполнением резервного копирования произведите настройку непрерывного архивирования WAL.

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL ARCHIVE:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=PAGE -j 1 --pguser=rubackuper --pgdatabase
=backupdb --no-password
```

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL STREAM:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=PAGE --stream -j 1 --pguser=rubackuper
--pgdatabase=backupdb --no-password
```

В режиме PTRACK



Перед выполнением резервного копирования произведите настройку резервного копирования в режиме PTRACK.

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL ARCHIVE:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=PTRACK -j 1 --pguser=rubackuper --pgdatabase
=backupdb --no-password
```

Команда на выполнение дифференциального резервного копирования в режиме доставки WAL STREAM:

```
pg_probackup backup -B /opt/rubackup/mnt/pg_probackup --instance
=имя_экземпляра --backup-mode=PAGE --PTRACK -j 1 --pguser=rubackuper
--pgdatabase=backupdb --no-password
```

Режим восстановления резервной копии кластера Postgres Pro

Данный метод может быть использован при ручном восстановлении служебной базы данных сервера RuBackup, если для её работы используется СУБД Postgres Pro и выполнялось её резервное копирование в составе кластера.



Перед восстановлением базы данных рекомендуется сделать резервную копию всех имеющихся файлов в каталоге кластера баз данных.

Для восстановления кластера СУБД Postgres Pro необходимо выполнить следующие действия:

• Остановить сервер Postgres Pro, если он работает:

sudo systemctl stop postgrespro-std-13.service

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

```
sudo -iu postgres (cd /var/lib/pgpro/std-13/data && tar cfv - *) | (cd
~/emergency_copy && tar xf - )
```

• Очистить каталог кластера баз данных:

sudo -iu postgres rm -rf /var/lib/pgpro/std-13/data/*

- Восстановить данные из резервных копий в директорию при помощи Менеджера Клиента RuBackup (RBC) или утилиты командной строки rb_archives.
- Запустить восстановление кластера Postgres Pro:

```
pg_probackup restore -B /postgreBackups/ --instance=postgres -i
ид_резервной_копии --no-validate -I режим_восстановления
```

 Восстановить права на владение файлами кластера для группы и пользователя postgres:

sudo chown -R postgres:postgres /var/lib/pgpro/std-13/data/

• Запустить сервер Postgres Pro:

sudo systemctl start postgrespro-std-13.service

Менеджер администратора RuBackup (RBM)

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

RBM может быть запущено администратором на основном сервере резервного копирования RuBackup.

Для запуска менеджера администратора RBM необходимо выполнить команду:

ssh -X user@rubackup_server
/opt/rubackup/bin/rbm

Пользователь, запускающий RBM, должен входить в группу rubackup.

На вкладке **Объекты** в левой части представлен список клиентов системы резервного копирования, в котором указано имя, уникальный HWID и описание. Клиенты, которые в данный момент находятся в online, будут отмечены зеленым цветом. Клиенты в состоянии offline – красным (Рисунок 1).

Для резервного копирования кластера Postgres Pro на хосте должен быть установлен клиент RuBackup и необходимые модули. Клиент должен быть авторизован администратором RuBackup.

В том случае, если клиент RuBackup был установлен, но не авторизован, в нижней части окна RBM будет сообщение о том, что найдены неавторизованные клиенты. Все новые клиенты должны быть авторизованы в системе резервного копирования.



Рисунок 1.

Для авторизации неавторизованного клиента в RBM выполните следующие действия:

 Нажмите на вкладку «Администрирование» и выберите иконку «Клиенты» (Рисунок 2):

🗲 Ru Backup			۵	ଡ
<u> </u>				
🖵 Объекты	Пользователи			
🗐 Стратегии	0 -			
🗏 Глобальное расписание	Пользователи	СС пользователей		
🖸 Удалённая репликация				
🕞 Репозиторий	Супервайзеры	Сопровождающие		
🖉 Очередь задач		<u> </u>		
🗄 Серверы RuBackup				
폐 Журналы				
& Администрирование	Объекты			
	Клиенты	요፲요 고드 고드		
	Медиасерверы			0
	Хранилища			
8 rubackup@rbfd-stress-primary & 1 неавторизованных клиентов	Пулы	Группы пулов		V

Рисунок 2.

• На верхней панели перейдите на вкладку «Неавторизованные клиенты» (Рисунок 3).

				RuBackup mar	nager (на rb-primary)			
RuBackup								۵ (
	÷				Неавторизованн	ые клиенты		
эекты	🗞 Авторизовать	🔟 Удалить						
	ID Имя	Тип ОС ОС дистрибы	отор МАС	IPv4	IPv6	HWID	Версия	
альное расписание	1 rb-client01	2 3	6e:t3:86:da:68:00	10.177.32.108	Te80::6CT3:86TTTeda:6800	a1811269C771Dd33	2023.02.21 10:35	
енная репликация								
иторий								
дь задач								
ры RuBackup								
истрирование								
				-	•			



• 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 курсивом).

Чтобы выполнять регулярное резервное копирование кластера СУБД Postgres Pro, необходимо создать правило в глобальном расписании. Для этого выполните следующие действия: Находясь в разделе Объекты, выберите вкладку Правила и нажмите на иконку
 + (Рисунок 6).

🗣 RuBackup								۵	۲
£	Поиск	Задачи	Правила	Правила стратегии	Репозиторий	Удалённая репликация	План восстановления		
🖵 Объекты	► rbfd-stress-fsclient	•							₿
🗐 Стратегии	 rbfd-stress-media01 rbfd-stress-media02 	ID Имя	глобальног	о расписания Статус	ID клиента И	мя клиента HWID	Имя пула	Ёмко	ость
🗏 Глобальное расписание	 rbfd-stress-pgclient 								
🖻 Удалённая репликация	► rbfd-stress-primary Prim								
🕞 Репозиторий	► rbfd-stress-secondary								
🕄 Очередь задач									
🗄 Серверы RuBackup									
폐 Журналы									
& Администрирование									
									0
A rubackup@rbfd-stress-primary & 1 неавторизованных клиентов		•	_						•

Рисунок 6.

• Выберите тип ресурса «Postgres Pro» (Рисунок 7):

🗣 Ru Backup 👘					<u>ድ</u> ይ @
⊆	÷		Добавить правило глобал	ьного расписания	🗸 Применить
00 Панель мониторинга					
🖵 Объекты		Правила гло	бального расписания		
🗐 Стратегии		Парам	иетры правила		
🗐 Глобальное расписание	Название правила				
С Удалённая репликация	Клиент	node10 (806aace	fc80e3900)	~	
Репозиторий	Тип ресурса	Postgres Pro		× •	
Очерель залач	Pecypc *				
	Тип РК	Полная		~	
		Дополнительн	ые параметры правила		
∞ журналы	Ёмкость хранилища		Inf	Гб - +	
🚡 Администрирование	Ёмкость хранилища к	лиента	Inf	Гб - +	
	Приоритет		100	- +	
	Защитное преобразов	зание	nocrypt	~	
	Скрипт при нормальн	ом выполнении			
	Скрипт при выполнен	ии с ошибками			(
	Скрипт при восстанов	злении			
▲ rubackup@10.177.32.5		Шаблон гло	бального расписания		

Рисунок 7.

В качестве ресурса будет автоматически подставлен путь до каталога кластера. Если вам необходимо сделать резервную копию другого ресурса - укажите полный путь до него вручную или воспользуйтесь кнопкой **Выбрать...**.

 Установить настройки правила: название правила, пул хранения данных, максимальный объём для резервных копий правила (в ГБ), тип резервного копирования, расписание резервного копирования, срок хранения и необязательный временной промежуток проверки резервной копии (Рисунок 8):

🗲 Ru Backup										£	۵ t	۲
⊆	÷			Доба	вить правило глобального расписания					\sim	Приме	нить
🔠 Панель мониторинга				Прарияа слоба							_	
🖵 Объекты				правила глоба	льного расписания							
🗐 Стратегии			Параметры правила		Дополнител	ьные пар	аметры правила					
🗐 Глобальное расписание	Название правила				Ёмкость хранилища	Inf		Гб -				
 Удалённая репликация 	Клиент	node10	806aacefc80e3900)	~	Ёмкость хранилища клиента	Inf		Гб -				
😫 Репозиторий	Тип ресурса	Postgres	Pro	× •	Приоритет	100			+			
С Очерель залач	Pecypc *				Защитное преобразование	nocry	/pt		<u> </u>			
	Тип РК	Полная		~	Скрипт при нормальном выполнении							
Е серверы коваскор					Скрипт при выполнении с ошибками							
ы Журналы -					Скрипт при восстановлении							
			Harroğun	Добав Шаблон глоба	ять правило в шаблон льного расписания	Pageuga						
	Включить после созда	ния	пастроики		Выбрано: крон-выражение 0 0 1 * *	Гасписа	чие					
	Пул		Default		Периодический запуск							
	Начало периода дейст	вия	04.04.2024 10:03	0	Минута		0		+			
	Окончание периода де	ействия	04.04.2025 10:03	0	Час	Ō	0		+			
					День месяца	0	1		+			
					Месяц							(?)
					День недели							
	Проверка резервных к	опий каж	Проверка		С Хранить резервные копии в течение	Срок хран	ения					
8 rubackup@10.177.32.5	1			+ Месяцев 🗸	1		- +	Лет	~			Ŧ

Рисунок 8.

При необходимости, администратор может приостановить работу правила или немедленно запустить его (т.е. инициировать немедленное создание задачи при статусе правила wait).

- Нажав на иконку ... рядом с выбранным типом ресурса, установите дополнительные настройки правила резервного копирования. Доступны следующие параметры:
 - threads количество потоков в которые будет выполняться резервное копирование или восстановление. По умолчанию количество потоков равно 1;
 - differential_mode выбор режима резервного копирования (О DELTA, 1 -PAGE, 2 - PTRACK). По умолчанию всегда DELTA;
 - stream выбор режима доставки WAL файлов. Если параметр имеет значение True (включен), то режимом доставки при выполнении резервного копирования будет STREAM, если значение примет False (выключен), то режимом будет ARCHIVE. По умолчанию значение равно True (включен).

Правила глобального расписания имеют срок жизни, определяемый при их созда-

нии, а также предоставляют следующие возможности:

- Выполнить защитное преобразование резервной копии на клиенте.
- Периодически выполнять проверку целостности резервной копии.
- Хранить резервные копии определённый срок, а после его окончания удалять их из хранилища резервных копий и из записей репозитория, либо просто уведомлять пользователей системы резервного копирования об окончании срока хранения.
- Через определённый срок после создания резервной копии автоматически переместить её на 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 исполнение правил может как администратор, с помощью RBM, так клиент при помощи RBC.

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

Менеджер клиента RuBackup (RBC)

Принцип взаимодействия клиентского менеджера (RBC) с системой резервного копирования состоит в том, что пользователь может сформировать ту или иную команду (желаемое действие) и отправить его серверу резервного копирования RuBackup. Взаимодействие пользователя с сервером резервного копирования производится через клиента (фоновый процесс) резервного копирования. Клиентский менеджер отправляет команду пользователя клиенту, клиент отправляет её серверу. В том случае, если действие допустимо, то сервер RuBackup отдаст обратную команду клиенту и/или перенаправит её медиасерверу RuBackup для дальнейшей обработки. Это означает, что, как правило, 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 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 клиентского менеджера поддерживает русский и английский языки.

Для запуска RBC следует выполнить команды:

```
ssh -X user@postgreshost
/opt/rubackup/bin/rbc
```

Пользователь, запускающий RBC, должен входить в группу rubackup.

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

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

Вкладка «Резервные копии»

В таблице вкладки «Резервные копии» содержится информация обо всех резервных копиях клиента, которые хранятся в репозитории RuBackup (Рисунок 9). Дифференциальные резервные копии ссылаются на полные резервные копии, 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.

					RuBackup ме	неджер клиен	та			- 🗆 😣					
Конфі	игурация В	ид Действия I	Информация	a 🛛											
Резер	езервные копии Глобальное расписание Задачи Локальное расписание Ограничения														
Id	Task ID	Reference ID	Resource	e type	Resource	Backup type	Pool	Archive size	Snapshot size	Created					
1 32	116		Postgres	Pro	/var/lib/pgpro/std-11/data/	full	Default	35130416	312276	2022-07-08 18:47:48+03					
2 33	117	32	Postgres Pr	ro	/var/lib/pgpro/std-11/data/	differential	Default	468950	263013	2022-07-08 18:48:43+03					
3 34	120		Postgres	Pro	/var/lib/pgpro/std-11/data/	full	Default	35076011	312276	2022-07-08 18:49:15+03					
4 35	121	34	Postgres Pr	o	/var/lib/pgpro/std-11/data/	differential	Default	523347	263013	2022-07-08 18:49:35+03					
5 36	122	34	Postgres Pr	o	/var/lib/pgpro/std-11/data/	differential	Default	468950	263013	2022-07-08 18:49:55+03					
					Duc										

Рисунок 9.

Во вкладке «Резервные копии» пользователю доступны следующие действия:

- 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.
- 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 клиент должен выбрать место для восстановления файлов резервной копии. Рекомендуется использовать временный каталог для операций с резервными копиями (например, /rubackup-tmp). Далее вы сможете провести восстановление кластера в ручном режиме. RBC не ожидает окончания восстановления всех резервных копий. Клиент должен проконтролировать на вкладке «Задачи» успешное завершение созданных задач на восстановление данных (статус Done). Для успешного выполнения этого действия требуется наличие достаточного свободного места в каталоге, предназначенном для создания и временного хранения резервных копий (см. параметр use-localbackup-directory).
- Проверить резервную копию. Это действие инициирует создание задачи проверки резервной копии. Если резервная копия была подписана цифровой подписью, то будет проверен размер файлов резервной копии и сама резервная

копия.

Вкладка «Глобальное расписание»

В таблице вкладки «Глобальное расписание» содержится информация обо всех правилах в глобальном расписании RuBackup для этого клиента (Рисунок 10).

	RuBackup менеджер клиента — 🗆 😣													
K	Конфигурация Вид Действия Информация													
Ρ	езерв	ные копии	Глобальное расписа	ние	Задачи	Локальное р	асписание	Ограниче	ния					
	Id	Rule name	Storage capacity, GB	Min	Hour	Day of month	Month	Day of week	Validity start period	Validity end period	Resource type			
1	6	pgpro full rule	0	0	0	1	January	Monday	2022-07-08 18:42:56.408+03	2023-07-08 18:42:56.408+03	Postgres Pro	/va		
2	7	pgpro diff rule	0	0	0	1	January	Monday	2022-07-08 18:43:17.303+03	2023-07-08 18:43:17.303+03	Postgres Pro	/va		



Во вкладке «Глобальное расписание» пользователю доступны следующие действия:

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

В таблице вкладки «Задачи» содержится информация обо всех задачах в главной очереди заданий RuBackup для этого клиента (Рисунок 11). 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).

	RuBackup менеджер клиента — 🗆 🛛														
K	Конфигурация Вид Действия Информация														
P	Резервные копии Глобальное расписание Задачи Локальное расписание Ограничения														
	Id	Туре	Resource type		Resource		Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created		
1	110	Backup global	Postgres Pro	/var/lib	/pgpro/std-	11/data/	full	6		30	Default	Done	2022-07-08 18:43:50+03		
2	111	Backup global	Postgres Pro	/var/lib	/pgpro/std-:	11/data/	full	6		31	Default	Done	2022-07-08 18:43:54+03		
з	116	Backup global	Postgres Pro	/var/lib	/pgpro/std-:	L1/data/	full	6		32	Default	Done	2022-07-08 18:47:35+03		
4	117	Backup global	Postgres Pro	/var/lib	/pgpro/std-:	L1/data/	differential	7		33	Default	Done	2022-07-08 18:48:27+03		
5	120	Backup global	Postgres Pro	/var/lib	/pgpro/std-:	1/data/	full	6		34	Default	Done	2022-07-08 18:48:59+03		
6	121	Backup global	Postgres Pro	/var/lib	/pgpro/std-:	L1/data/	differential	7		35	Default	Done	2022-07-08 18:49:26+03		
7	122	Backup global	Postgres Pro	/var/lib	/pgpro/std-	L1/data/	differential	7		36	Default	Done	2022-07-08 18:49:38+03		

Рисунок 11.

Вкладка «Локальное расписание»

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

Вкладка «Ограничения»

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

Утилиты командной строки клиента RuBackup

Для управления RuBackup со стороны клиента, помимо клиентского оконного менеджера, можно воспользоваться утилитами командной строки: rb_archives, rb_schedule и rb_tasks.

rb_archives

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

rb_archives

LOO.	t@post	gres	Pro-client:~# rb_archives						
Id	Ref	ID	Resource	Resource type	Backup type	Created	Crypto	Signed	Status
	+	+				+	+	+	
32			/var/lib/pgpro/std-11/data/	Postgres Pro	full	2022-07-08 18:47:48+03	nocrypt	True	Trusted
33	32	- 1	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:48:43+03	nocrypt	True	Trusted
34	Ì	Í	/var/lib/pgpro/std-11/data/	Postgres Pro	full	2022-07-08 18:49:15+03	nocrypt	True	Trusted
35	34	Í	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:49:35+03	nocrypt	True	Trusted
36	34	Í	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:49:55+03	nocrypt	True	Trusted
	1 a - a								

rb_schedule

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

rb_schedule

root Id	C@postgresPro-clier Name	it:~# rb_schedule Resource type	Resource	Backup type Status
б	pgpro full rule	Postgres Pro	/var/lib/pgpro/std-11/data/	full wait
7	pgpro diff rule	Postgres Pro	/var/lib/pgpro/std-11/data/	differential wait

rb_tasks

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

rb_tasks

гоот	root@postgresPro-client:~# rb_tasks								
Id	Task type	Resource	Backup type	Status	Created				
	-++	+		+					
110	Backup global	/var/lib/pgpro/std-11/data/	full	Done	2022-07-08 18:43:50+03				
111	Backup global	/var/lib/pgpro/std-11/data/	full	Done	2022-07-08 18:43:54+03				
116	Backup global	/var/lib/pgpro/std-11/data/	full	Done	2022-07-08 18:47:35+03				
117	Backup global	/var/lib/pgpro/std-11/data/	differential	Done	2022-07-08 18:48:27+03				
120	Backup global	/var/lib/pgpro/std-11/data/	full	Done	2022-07-08 18:48:59+03				
121	Backup global	/var/lib/pgpro/std-11/data/	differential	Done	2022-07-08 18:49:26+03				
122	Backup global	/var/lib/pgpro/std-11/data/	differential	Done	2022-07-08 18:49:38+03				

Восстановление резервной копии кластера Postgres Pro

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

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

Восстановление резервной копии в RBC

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

- Выделить нужную резервную копию и в контекстном меню выбрать **Восстано**вить.
- Ввести пароль клиента и далее RBC выведет информационное сообщение о дальнейших действиях.
- Указать в качестве временного места восстановления резервных копий каталог, отдельный от копируемого каталога кластера баз данных (/var/lib/pgpro/std-13/data/).
- Далее появится информационное сообщение о создании задачи на восстановление.
- Проконтролировать результат процесса восстановления можно после автоматического переключения RBC на вкладку Задачи (Рисунок 12):

	RuBackup менеджер клиента — 🗆 🕴										_ 0 😣
K	Конфигурация Вид Действия Информация										
P	Резервные копии Глобальное расписание Задачи Локальное расписание Ограничения										
	Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created
1	110	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		30	Default	Done	2022-07-08
2	111	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		31	Default	Done	2022-07-08
3	116	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		32	Default	Done	2022-07-08
4	117	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ differential	7		33	Default	Done	2022-07-08
5	120	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		34	Default	Done	2022-07-08
6	121	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ differential	7		35	Default	Done	2022-07-08
7	122	Backup global	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ differential	7		36	Default	Done	2022-07-08
8	128	Restore	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		32	Default	Done	2022-07-08
9	129	Restore	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ full	6		34	Default	Done	2022-07-08
10	130	Restore	Postgres Pro	/var/lib/pgpro/std-11/d	ata/ differential	7		35	Default	Done	2022-07-08

Рисунок 12.

Восстановление при помощи утилиты rb_archives

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

rb_archives

-									
root@postgresPro-client:~# rb_archives									
Id	Ref ID	Resource	Resource type	Backup type	Created	Crypto	Signed S	tatus	
		· +			•				
22				611			T		
32		/var/llb/pgpro/std-11/data/	Postgres Pro	TULL	2022-07-08 18:47:48+03	ποςгурτ	irue [i	rusted	
33	32	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:48:43+03	nocrypt	True T	rusted	
34		/var/lib/pgpro/std-11/data/	Postgres Pro	full	2022-07-08 18:49:15+03	nocrypt	True T	rusted	
35	34	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:49:35+03	nocrypt	True T	rusted	
36	34	/var/lib/pgpro/std-11/data/	Postgres Pro	differential	2022-07-08 18:49:55+03	nocrypt	True T	rusted	

В приведённом примере в системе резервного копирования присутствуют пять резервных копий с идентификаторами 32, 33, 34, 35 и 36. Для восстановления резервной копии 32 необходимо выполнить команду:

```
rb_archives -x 32
root@postgresPro-client:~# rb_archives -x 32
Password:
The archive will be restored in the directory: /rubackup-tmp
----> Restore archive chain: 32 < ----
Record ID: 32 has status: Trusted
TASK WAS ADDED TO QUEUE:131</pre>
```

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:

rl	b_	ta	s	ks
	_			

root@postgresPro-client:~# rb_tasks								
Id Task type	Resource		Backup type	Status	Created			
+	+	-+-		++				
110 Backup global	/var/lib/pgpro/std-11/data/	T	full	Done	2022-07-08			
111 Backup global	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			
116 Backup global	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			
117 Backup global	/var/lib/pgpro/std-11/data/		differential	Done	2022-07-08			
120 Backup global	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			
121 Backup global	/var/lib/pgpro/std-11/data/		differential	Done	2022-07-08			
122 Backup global	/var/lib/pgpro/std-11/data/		differential	Done	2022-07-08			
128 Restore	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			
129 Restore	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			
130 Restore	/var/lib/pgpro/std-11/data/		differential	Done	2022-07-08			
131 Restore	/var/lib/pgpro/std-11/data/		full	Done	2022-07-08			

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

tail	tail -f /opt/rubackup/log/RuBackup.log							
rootanos	toresPro-client:~# tail -f /opt/rubackup/log/RuBackup.log							
Fri Jul	8 20:41:17 2022: INFO: Backup files are restored. Transfered bytes: 32MB. time elapsed: 1s							
Fri Jul	8 20:41:17 2022: INFO: Restore incremental ratio (less is better): 6% (32MB/575MB)							
Fri Jul	8 20:41:17 2022: INFO: Syncing restored files to disk							
Fri Jul	8 20:41:19 2022: INFO: Restored backup files are sunced time plansed: 2s							
Fri Jul	8 20:41:19 2022: INFO: Restore of backup REPI7B completed.							
Fri Jul	8 20:41:21 2022: The beginning of the process to clean up the '/ont/rubackup/mnt/ng probackup/' directory							
Fri Jul	8 20:41:21 2022: INFO: Resident data size to free by delete of backup REPL7R : 575MB							
Fri Jul	8 20-41-21 2022: INFO: Delete: BEDI 78 2022-07-08 18-47-38-03							
Fri Jul	8 20-41-21 2022: The clean up process is completed							
Fri Jul	8 20:41:21 2022: Task was done. ID: 131							