

RuBackup

Система резервного копирования и восстановления данных

Резервное копирование и восстановление базы данных

RuBackup



RuBackup

Версия 1.9

2022 г.

Содержание

Введение.....	3
Настройка резервного копирования базы данных RuBackup.....	4
Восстановление базы данных RuBackup.....	7

Введение

Информация о системе резервного копирования RuBackup хранится с базе данных rubackup 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 СУБД PostgreSQL».

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

Для непрерывного архивирования и восстановления СУБД PostgreSQL требуется включить архивирование WAL. В конфигурационном файле `/etc/postgresql/10/main/postgresql.conf` необходимо установить параметры (расположение конфигурационного файла может быть другим в других дистрибутивах, проконсультируйтесь по этому поводу у вашего администратора PostgreSQL) :

```
wal_level = archive

archive_mode = on

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

archive_timeout = 300
```

После внесения изменений необходимо перезапустить PostgreSQL.

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

В настройках RuBackup для каждой СУБД PostgreSQL имеется параметр `archive_catalog`, содержащий значение каталога, в котором предполагается создание архивных WAL файлов. Значение этого параметра по умолчанию:

```
/opt/rubackup/mnt/postgresql_archives/
```

2. Создайте пользователя PostgreSQL для безопасного выполнения резервного копирования базы данных RuBackup

Вызовите psql при помощи команды:

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

В psql создайте пользователя rubackup_backuper, в качестве пароля укажите желаемый пароль вместо qwerty1234:

```
# create user rubackup_backuper password 'qwerty1234';
```

```
# alter role rubackup_backuper with login;
```

```
# grant execute on function pg_start_backup to rubackup_backuper;
```

```
# grant execute on function pg_stop_backup(bool, bool) to rubackup_backuper;
```

3. Создайте правила резервного копирования базы данных RuBackup

Для возможности быстрого и надежного восстановления базы данных RuBackup рекомендуется выполнять один раз в неделю базовое резервное копирование и шесть дней в неделю резервное копирование архивных WAL файлов. Для этого при помощи менеджера системного администратора RBM или используя утилиту командной строки rb_schedule требуется создать два правила резервного копирования в глобальном расписании RuBackup, для базового резервного копирования и для резервного копирования архивных WAL файлов. Самый простой способ создать правило - импортировать его из предварительно подготовленного файла. В каталоге /opt/rubackup/rc/ находятся несколько шаблонов для импорта правил в глобальное расписание. В этих шаблонах всего вам потребуется поправить имя клиента (установите hostname сервера, где располагается база данных RuBackup), также необходимо установить желательные значения для начала (Validity start period) и окончания (Validity end period) действия правила, условия старта процедуры резервного копирования (Minute, Hour, Day of month, Month, Day of week). Так же целесообразно размещать резервные копии базы данных RuBackup в выделенном пуле, чтобы в случае необходимости восстановить базу данных быстро определить местонахождение резервных копий.

Правило для базового резервного копирования

- 1) Отредактируйте файл /opt/rubackup/rc/rule.export.postgresql.full.self.
- 2) Импортируйте правило при помощи команды

```
# rb_global_schedule -i /opt/rubackup/rc/rule.export
```

- 3) Определите ID правила с помощью команды

```
# gb_global_schedule -l
```

Скорее всего это будет последнее по номеру правило в глобальном расписании с именем “RuBackup self PostgreSQL DB”, типом резервного копирования “full” и статусом “wait”.

- 4) Запустите правило в работу при помощи команды

```
# gb_global_schedule -r ID
```

ID – это номер правила, который вы определили шагом ранее.

Правило для копирования архивных WAL файлов

- 1) Отредактируйте файл /opt/rubackup/rc/rule.export.postgresql.inc.self
- 2) Импортируйте правило при помощи команды

```
# gb_global_schedule -i /opt/rubackup/rc/rule.export
```

Скорее всего это будет последнее по номеру правило в глобальном расписании с именем “RuBackup self PostgreSQL DB”, типом резервного копирования “incremental” и статусом “wait”.

- 3) Запустите правило в работу при помощи команды

```
# gb_global_schedule -r ID
```

ID - это номер правила, который вы определили шагом ранее.

После того, как вы запустили в работу правила резервного копирования базы данных RuBackup все необходимые процедуры будут выполняться автоматически. В созданных правилах глобального расписания рекомендуется настроить уведомления в адрес системного администратора RuBackup об успешном и неудачном выполнении процедур резервного копирования.

Восстановление базы данных

RuBackup

В случае необходимости восстановить базу данных RuBackup можно воспользоваться клиентским менеджером `rbc_postgresql`. Если RuBackup функционирует, то потребуется восстановить цепочку резервных копий в определенный каталог, остановить все серверы RuBackup и после этого восстанавливать базу данных согласно процедуре, описанной в руководстве “Резервное копирование и восстановление СУБД PostgreSQL”.

В том случае, если база данных RuBackup повреждена и серверы RuBackup не могут работать с такой базой данных, необходимо вручную перенести необходимые резервные копии базы данных в определенный каталог и после этого восстанавливать базу данных согласно процедуре, описанной в руководстве “Резервное копирование и восстановление СУБД PostgreSQL”. Если резервные копии были преобразованы, то перед восстановлением необходимо провести их обратное преобразование, используя утилиту `rbcrypt`.

Перед процедурой восстановления базы данных RuBackup рекомендуется ознакомиться с соответствующей документацией PostgreSQL.