

RuBackup

Система резервного копирования и восстановления данных

Резервное копирование и восстановление метаданных дедуплицированного пула



RuBackup

Версия 2.3.0

25.10.2024

Содержание

Введение.....	3
Резервное копирование метаданных дедуплицированного пула.....	4
Резервное копирование метаданных дедуплицированного пула с помощью скрипта.....	5
Резервное копирование метаданных дедуплицированного пула вручную.....	6
Восстановление метаданных дедуплицированного пула.....	8
Приложение А.....	10

Введение

Система резервного копирования и восстановления данных RuBackup (далее – Система, СРК) – системное клиент-серверное приложение, предназначенное для автоматизированного выполнения процедур резервного копирования данных серверов, виртуальных машин, баз данных и приложений в центрах обработки данных, а также для восстановления данных из резервных копий по запросу пользователя или системного администратора.

Дедупликация — это процесс исключения дублирующихся данных. Дедупликация позволяет оптимизировать использование дискового пространства систем хранения данных. Для выполнения процесса дедупликации необходимо наличие дедуплицированного пула. Дедуплицированный пул — это пул, в котором выполняется процесс дедупликации.

Резервное копирование метаданных дедуплицированного пула

Метаданные дедуплицированного пула хранятся в следующих таблицах СРК RuBackup:

- `pool_list`.
- `pool_block_device_extention`.
- `storage_block_devices`.
- `deduplicated_block_device_<signature>`.

Существует два способа резервного копирования метаданных дедуплицированного пула:

- С помощью скрипта `script_block_device_metadata.sh`.
- Вручную, используя утилиту `pg_dump`.

Резервное копирование метаданных дедуплицированного пула с помощью скрипта

Для резервного копирования метаданных дедуплицированного пула с помощью скрипта необходимо:

1. Перевести СРК RuBackup в сервисный режим. Для этого необходимо перейти в меню **Настройки** → **Глобальная конфигурация** и включить переключатель **Сервисный режим** (1).

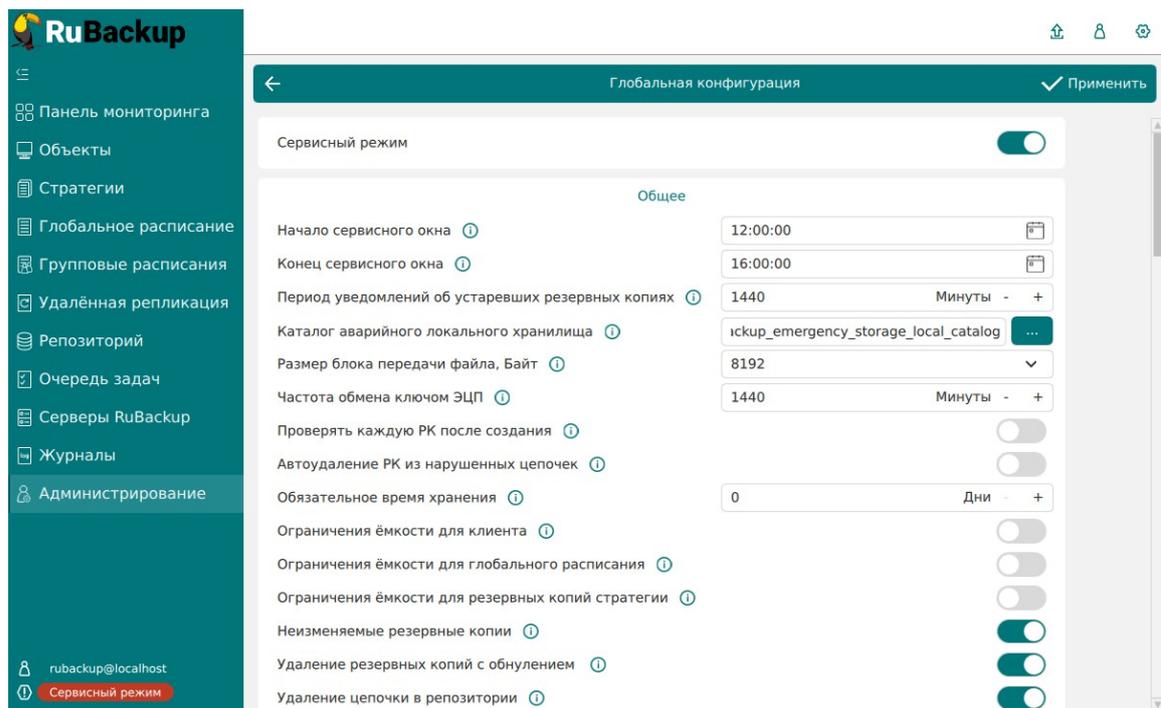


Рисунок 1

2. Открыть на редактирование скрипт `script_block_device_metadata.sh` (Приложение А) и задать обязательные параметры:

- `HOST="localhost"` — адрес хоста с базой данных.
- `DBNAME="rubackup"` — имя базы данных.
- `USER="rubackup"` — имя пользователя базы данных.
- `PASS="12345"` — пароль пользователя базы данных.
- `BACKUP_FILENAME="rb_block_device_metadata_backup.sql"` — имя файла резервной копии выбранных таблиц.

3. Запустить скрипт `script_block_device_metadata.sh` с параметром `dump`:

```
# bash ./script_block_device_metadata.sh dump
```

```
root@eakulinina-primary:/home/u# bash ./script_dump_block_device_metadata.sh dump
RuBackup script handler saving Dedup pool metadata started
A backup copy of the table is saved in a file rb_block_device_metadata_backup.sql
```

В результате в текущем каталоге будет создана резервная копия выбранных таблиц в формате `.sql`.

Резервное копирование метаданных дедулицированного пула вручную

Для резервного копирования метаданных дедулицированного пула вручную необходимо:

1. Перевести СРК RuBackup в сервисный режим. Для этого необходимо перейти в меню **Настройки** → **Глобальная конфигурация** и включить переключатель **Сервисный режим** (1).

2. С помощью команды **pg_dump** выполнить резервное копирование следующих таблиц из базы данных RuBackup:

- `pool_list`.
- `pool_block_device_extention`.
- `storage_block_devices`.
- `deduplicated_block_device_<signature>`.

Пример команды для резервного копирования таблицы `pool_list` в файл `backup.sql`:

```
# pg_dump -h localhost -d rubackup -U rubackup -t pool_list >backup.sql
```

Для таблицы `deduplicated_block_device_<signature>` необходимо получить параметр `signature`. `Signature` — это уникальная подпись для каждого блочного устройства. Значение `signature` можно получить следующими способами:

- С помощью утилиты `rb_block_devices`:

```
$ rb_block_devices -v
```

```
u@eakulinina-primary:~$ rb_block_devices -v
Id | Device | Usage | Dedup.c. | Pool Name | Blocks | Block size | Shared | Signature
---+---+---+---+---+---+---+---+---
4 | /dev/vda | 0% | 0,250000 | Dedup | 655359 | 16384 | | 7fb0b3bac69d4fa0
```

- В RBM в разделе «Блочные устройства» в колонке «Подпись» (2).

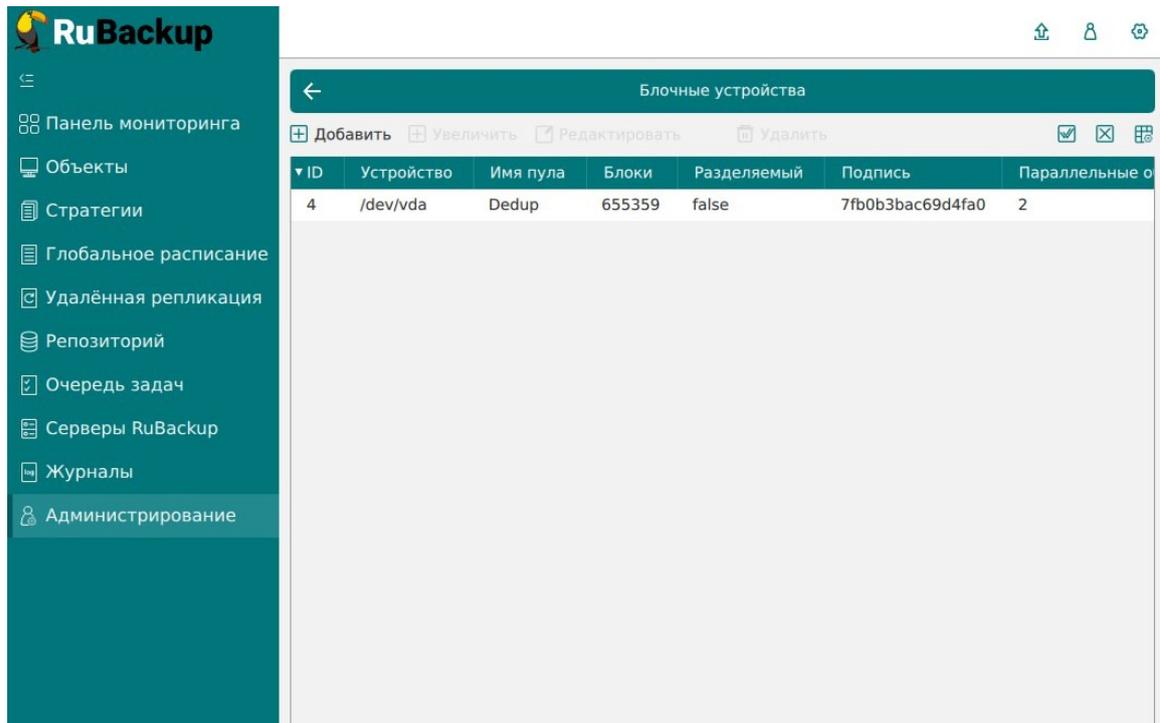


Рисунок 2

Пример команды для резервного копирования таблицы `deduplicated_block_device_<signature>` в файл `backup.sql`:

```
# pg_dump -h localhost -d rubackup -U rubackup -t  
deduplicated_block_device_7fb0b3bac69d4fa0 >backup.sql
```

В результате в текущем каталоге будет создана резервная копия выбранных таблиц в формате `.sql`.

Восстановление метаданных

дедуплицированного пула

Для восстановления метаданных дедуплицированного пула необходимо:

1. После сбоя СРК, заново настроить RuBackup согласно руководству по установке (см. документ «Руководство по установке и обновлению серверов резервного копирования и Linux клиентов RuBackup»).

2. Перевести СРК RuBackup в сервисный режим (1).

3. Восстановить резервные копии таблиц одним из двух способов:

- С помощью скрипта `script_block_device_metadata.sh` (Приложение А) с параметром `restore`:

```
# bash ./script_block_device_metadata.sh restore
```

```
root@eakulinina-primary:/home/u# bash ./script_dump_block_device_metadata.sh restore
RuBackup script handler restores Dedup pool metadata started
SET
SET
SET
SET
SET
SET
set_config
```

- С помощью команды `psql` восстановить заранее сохраненные таблицы (`pool_list`, `pool_block_device_extention`, `storage_block_devices`, `deduplicated_block_device_<signature>`):

```
# psql -h localhost -d rubackup -U rubackup -f
rb_block_device_metadata_backup.sql
```

4. С помощью утилиты `rb_inventory` внести в базу данных RuBackup информацию о всех резервных копиях, которые были сделаны до сбоя (подробнее об утилите `rb_inventory` смотрите в документе «Утилиты командной строки RuBackup»):

```
$ rb_inventory -i /dir
```

5. С помощью утилиты `rb_block_devices` обновить имя устройства, если оно было изменено:

```
$ rb_block_devices -c ID -n block_device
```

Где:

- ID – уникальный номер блочного устройства, который можно узнать, запустив утилиту `rb_block_devices -v`:

\$ rb_block_devices -v

```
u@eakulinina-primary:~$ rb_block_devices -v
Id | Device   | Usage | Dedup.c. | Pool Name | Blocks | Block size | Shared | Signature
---+-----+-----+-----+-----+-----+-----+-----+-----
4  | /dev/vda | 0%    | 0,250000 | Dedup     | 655359 | 16384      |        | 7fb0b3bac69d4fa0
```

- `block_device` – новое имя блочного устройства.

6. Перезапустить сервер RuBackup:

\$ sudo systemctl stop rubackup_server

\$ sudo systemctl start rubackup_server

В результате будут восстановлены метаданные дедуплицированного пула.

Приложение А

(справочное)

Скрипт script_block_device_metadata.sh

```
#!/bin/bash

# Параметры подключения к базе данных
HOST="localhost"
DBNAME="rubackup"
USER="rubackup"
PASS="12345"
BACKUP_FILENAME="rb_block_device_metadata_backup.sql"

TABLE_LIST="-t pool_list -t pool_block_device_extention -t
storage_block_devices" # Список таблиц для резервного копирования

if [ "$#" -eq 1 ]; then
    if [ "$1" = "dump" ]; then
        echo "RuBackup script handler saving Dedup pool metadata
started"

        # Извлечение подписей из столбца "signature" в таблице
"storage_block_devices"
        SIGNATURES=$(PGPASSWORD=$PASS psql -h $HOST -d $DBNAME -U $USER
-qt -c "SELECT DISTINCT signature FROM storage_block_devices")

        # Формирование строки с перечислением подписей
        for signature in $$SIGNATURES; do
            table_name="deduplicated_block_device_{$signature}" #
Формирование имени таблицы
            TABLE_LIST="$TABLE_LIST -t $table_name" #
Добавляем таблицу к списку
        done

        # Создание резервной копии всех таблиц в одном файле
        PGPASSWORD=$PASS pg_dump -h $HOST -d $DBNAME -U $USER
$TABLE_LIST >$BACKUP_FILENAME

        echo "A backup copy of the table is saved in a file
$BACKUP_FILENAME"
```

```
        exit 0
    fi

    if [ "$1" = "restore" ]; then
        echo "RuBackup script handler restores Dedup pool metadata
started"
        # Восстановление
        PGPASSWORD=$PASS psql -h localhost -d rubackup -U rubackup -f
$BACKUP_FILENAME
        echo "RuBackup script handler restores Dedup pool metadata
finished"
        exit 0
    fi

    echo "Incorrect argument. Type 'dump' or 'restore'"
    exit 1
fi

echo "Argument required. Type 'dump' or 'restore'"
exit 1
```