



RuBackup

**Система резервного копирования
и восстановления данных**

УТИЛИТЫ КОМАНДНОЙ СТРОКИ

ВЕРСИЯ 2.4.0, 20.06.2025

Содержание

rb_archives	5
rb_schedule	9
rb_tasks	10
rb_bandwidth	11
rb_block_device_check	13
rb_block_devices	14
rb_client_defined_storages	15
rb_client_group	16
rb_clients	17
rb_cloud_task_queue	20
rb_clouds	21
rb_copy2pool	24
rb_dbms	26
rb_dynamic_group_pools	27
rb_global_config	30
rb_global_schedule	31
Примеры создания правила	32
rb_health_check	38
rb_init	39
rb_init (неинтерактивный режим)	39
Конфигурация основного сервера	40
Конфигурация автономного клиента	43
Конфигурация клиент (клиент-серверная архитектура)	45
Конфигурация дополнительного сервера	46
Конфигурация медиасервера	48
rb_interoperation	51
rb_inventory	53
rb_local_filesystems	54
rb_log_viewer	55
rb_logger_settings	56
rb_media_servers	58
rb_modules	60
rb_notifications	61
rb_pools	63
rb_remote_replication	65

rb_repository	67
rb_secret_storage	69
rb_security	72
rb_strategies	75
rb_tape_cartridges	77
rb_tape_libraries	79
rb_task_queue	81
rb_tl_task_queue	83
rb_update	85
rb_user_groups	87
rb_users	88
rbfd	90
Мастер-ключ	91
Алгоритмы защитного преобразования	92
Алгоритмы хеш-функций	92

Вы можете управлять системой резервного копирования RuBackup из графической консоли с помощью оконного менеджера администратора RBM и оконного менеджера клиента RBC, а также с помощью утилит командной строки.

Все утилиты располагаются в каталоге `/opt/rubackup/bin`.

В этом руководстве описаны утилиты командной строки администратора и клиента системы резервного копирования RuBackup.

Таблица 1. Список утилит командной строки клиента

rb_archives	просмотр списка резервных копий клиента, создание срочных резервных копий, их удаление, проверка и восстановление
rb_schedule	просмотр правил клиента в глобальном расписании резервного копирования
rb_tasks	просмотр задач клиента в главной очереди задач системы резервного копирования

Таблица 2. Список утилит командной строки администратора

rb_bandwidth	управление ограничениями пропускной способности при выполнении операций резервного копирования и восстановления
rb_block_devices	управление блочными устройствами
rb_block_device_check	проверка целостности резервной копии на блочном устройстве
rb_client_defined_storages	управление клиентскими хранилищами RuBackup
rb_client_group	управление группами клиентов
rb_clients	управление клиентами
rb_clouds	управление хранилищами резервных копий типа облако S3
rb_cloud_task_queue	информация о текущих задачах, связанных с облачными операциями
rb_copy2pool	управление репликацией резервных копий при их создании
rb_dbms	возможность расширенного вывода информации о резервных копиях
rb_dynamic_group_pools	управление динамическими группами пулов
rbfd	создание и восстановление полных и инкрементальных резервных копий блочных устройств, файлов и каталогов в любых файловых системах
rb_global_config	управление глобальной конфигурацией
rb_global_schedule	управление глобальным расписанием
rb_health_check	проверка конфигурации клиента и его окружения
rb_init	первоначальная настройка клиента или сервера
rb_interoperation	управление импортом и экспортом резервных копий
rb_inventory	инвентаризация резервных копий

rb_local_filesystems	управление хранилищами резервных копий типа файловая система медиасерверов
rb_log_viewer	просмотр журналов
rb_logger_settings	управление параметрами логирования
rb_media_servers	управление медиасerverами
rb_modules	управление модулями RuBackup
rb_notifications	контроль работы очереди уведомлений
rb_pools	управление пулами
rb_remote_replication	управление правилами непрерывной удаленной репликации
rb_repository	управление репозиторием резервных копий
rb_security	работа с журналом событий информационной безопасности
rb_secret_storage	управления доступом к аутентификационной информации в хранилище секретов HashiCorp Vault
rb_strategies	управление стратегиями резервного копирования
rb_tape_cartridges	управление картриджами ленточных библиотек
rb_tape_libraries	управление ленточными библиотеками
rb_task_queue	контроль работы главной очереди задач
rb_tl_task_queue	контроль работы очереди ленточных библиотек
rb_update	утилита сравнения существующей базы данных и sql скрипта создания новой базы данных
rb_user_groups	управление группами пользователей
rb_users	управление пользователями

Пользователи, от имени которых будет осуществляться запуск утилит командной строки RuBackup должны:

- иметь правильно настроенные переменные среды;
- входить в группу rubackup (создана во время установки клиента RuBackup).

Также доступны man-руководства для утилит командной строки, входящих в состав дистрибутива. Для получения доступа необходимо определить переменную `MANPATH` следующим образом:

```
export MANPATH=$MANPATH:/opt/rubackup/man
```

rb_archives

rb_archives - утилита клиента RuBackup для просмотра списка, удаления, проверки и немедленного выполнения резервного копирования по инициативе клиента. **rb_archives** работает только в том случае, если на клиенте работает служба (сервис, демон) клиента RuBackup.

При первом взаимодействии клиента с системой резервного копирования при помощи утилиты **rb_archives** или при помощи RBC потребуется установить пароль клиента.

-p

используйте параметр для изменения пароля клиента

Если вы забыли пароль клиента, то для его сброса следует обратиться к администратору системы резервного копирования RuBackup (см. [rb_clients](#)).

-h

Справочное сообщение

-L

Список системных имен всех модулей резервного копирования, которые поддерживаются на данном клиенте

-v

предоставляет расширенную информацию о наименовании модулей

-l Модуль

Список резервных копий клиента для выбранного модуля (требуется задать системное имя модуля)

-v

предоставляет расширенную информацию о параметрах резервных копий. Если клиент входит в общую группу клиентов, то утилита покажет список резервных копий всех клиентов этой группы

-V Просмотр

Режим просмотра списка резервных копий. Эта опция добавляет столбцы "Rule name", "Strategy name", которые помогают сориентироваться в том случае, если реальное имя ресурса - это абстрактный ID, плохо воспринимаемый человеком

-s Ресурс -m Модуль [-i|-D] [-a Кристоалгоритм] [-R Период хранения] [-e Параметр:Значение, Параметр1:Значение1,...] [-z]

Создать срочную резервную копию определённого ресурса. Эта команда выполняет полное, инкрементальное или дифференциальное резервное копи-

вание. Если ресурс располагается в файловой системе, следует использовать абсолютный путь. (если ни один из этих параметров не используется, то запускается полное резервное копирование)

-m -i | -D

определяет системное имя модуля, для которого необходимо выполнить срочное резервное копирование

-i

запускает инкрементальное резервное копирование

-D

запускает дифференциальное резервное копирование

-a Кристоалгоритм

позволяет выбрать алгоритм защитного преобразования резервной копии (см. [rbfd](#) для полного перечня возможных алгоритмов)

-R Период хранения

позволяет задать период хранения резервной копии, например: 1 year, 1 month, 2 weeks, 3 days

-e Параметр:Значение, Параметр1:Значение1, ...

позволяет задать значения нескольких параметров сразу в виде Параметр:Значение, перечислив их через запятую, например,

```
worker_parallelism:8, enable_multithreading:true, enable_flexible_dedup:true,
network_parallelism:8, memory_threshold:8, deny_memory_exceed:true,
deduplication:true
```

(описание параметров см. в разделе «Объекты» в «Руководстве системного администратора RuBackup)

-z

запускает архивацию резервной копии после её создания

-r ID

Создать запрос в главной очереди задач на удаление резервной копии. Клиент может создать такой запрос только в том случае, если это разрешено правилом глобального расписания, либо если резервная копия была создана как срочная резервная копия по инициативе клиента. При удалении резервной копии требуется ввести пароль клиента

-y ID

Создать запрос в главной очереди задач на проверку резервной копии с идентификатором ID

-x ID [-d Каталог] [-e Параметр:Значение, Параметр1:Значение1, ...]

Восстановить резервную копию или цепочку резервных копий

-x ID

указывает на идентификатор резервной копии, которую необходимо восстановить

-d Каталог

путь к каталогу, где будет происходить восстановление резервной копии

-e Параметр:Значение, Параметр1:Значение1, ...

предоставляет возможность указать дополнительные параметры восстановления резервной копии для различных модулей сред виртуализации, таких как VMware, VMmanager, Rustack и другие. Позволяет настроить процесс восстановления в соответствии с особенностями и требованиями конкретной виртуальной среды восстановления

-X ID [-d Каталог] [-f файл1, файл2, ...] [-e Параметр:Значение, Параметр1:Значение1, ...]

Восстановить без развёртывания (без восстановления в целевой системе, например, без создания виртуальной машины в платформе виртуализации и т.п.) резервную копию или цепочку резервных копий

-x ID

указывает на идентификатор резервной копии, которую необходимо восстановить

-d Каталог

путь к каталогу, где будет происходить восстановление резервной копии

-f файл1, файл2, ...

задается список файлов при гранулярном восстановлении

-e Параметр:Значение, Параметр1:Значение1, ...

предоставляет возможность указать дополнительные параметры восстановления резервной копии для различных модулей сред виртуализации, таких как VMware, VMmanager, Rustack и другие. Позволяет настроить процесс восстановления в соответствии с особенностями и требованиями конкретной виртуальной среды восстановления

-C ID -p ID_пула

Копировать резервную копию с идентификатором ID в пул с идентификатором ID_пула

-M ID -p ID_пула

Переместить резервную копию с идентификатором ID в пул с идентификатором ID_пула

-f ID [-v]

Просмотреть список файлов резервной копии, если это к ней применимо

-A

Вывести список доступных алгоритмов защитного преобразования

-O

Вывести список специальных параметров модуля

-o

Вывести список параметров восстановления модуля

-P

Вывести список доступных пулов RuBackup

-w

Изменить пароль клиента

-S

Показывает размеры архивов в удобном для пользователя формате. В зависимости от размера РК, размер может отображаться в следующих единицах измерения: **B, kB, MB, GB, TB, PB, EB, ZB, YB**

rb_schedule

rb_schedule - утилита клиента RuBackup для просмотра правил глобального расписания клиента в системе резервного копирования.

-h

Справочное сообщение

-v

Выводит список правил глобального расписания клиента. Предоставляет расширенную информацию о правилах глобального расписания клиента

rb_tasks

rb_tasks - утилита клиента RuBackup для просмотра списка задач клиента в системе резервного копирования RuBackup.

-h

Справочное сообщение

-v

Выводит список задач клиента в системе резервного копирования. Предоставляет расширенную информацию о параметрах задач

rb_bandwidth

rb_bandwidth - утилита администратора RuBackup для управления ограничениями пропускной способности при выполнении операций резервного копирования для клиентов или правил глобального расписания. Вы можете установить одно или несколько ограничений пропускной способности для определённого клиента системы резервного копирования или для какого-либо правила глобального расписания.

Если установленные ограничения для клиента пересекаются во времени с аналогичными ограничениями правила глобального расписания для этого же клиента, то действующее ограничение будет определено глобальным параметром «Предпочтительный способ ограничения скорости» (**bandwith_limit_advantage**) - допустимые значения `rule` или `client`.

Если установленные ограничения для одного и того же клиента пересекаются друг с другом, то действующее ограничение будет определено глобальным параметром «Способ ограничения скорости для клиента» (**bandwith_client_limit**) - допустимые значения `maximum` или `minimum`.

Если установленные ограничения для одного и того же правила пересекаются во времени друг с другом или глобальным параметром «Способ ограничения скорости для правила» (**bandwith_rule_limit**) - допустимые значения `maximum` или `minimum`.

-h

Справочное сообщение

-L

Список ограничений для правил глобального расписания

-c Имя_хоста **-b** Ограничение_бэкапа **-r** Ограничение_восстановления **-s** Время_начала **-e** Время_конца

Установить новые ограничения для клиента

-b Ограничение_бэкапа

определяет максимально допустимую пропускную способность при выполнении операций резервного копирования в МБ/с

-r Ограничение_восстановления

определяет максимально допустимую пропускную способность при выполнении операций восстановления в МБ/с

-s Время_начала

определяет время начало действия ограничения, например, 00:01:02

-e Время_конца

определяет время окончания действия ограничения, например, 23:59:59 Значение Время_конца должно быть всегда больше значения Время_начала, иначе ограничение работать не будет

-g ID_правила -b Ограничение_бэкапа -r Ограничение_восстановления -s Время_начала -e Время_конца

Установить новое ограничения для правила глобального расписания с идентификатором ID_правила (узнать идентификаторы правил глобального расписания можно с помощью утилиты [rb_global_schedule](#)).

-b Ограничение_бэкапа

определяет максимально допустимую пропускную способность при выполнении операций резервного копирования в МБ/с

-r Ограничение_восстановления

определяет максимально допустимую пропускную способность при выполнении операций восстановления в МБ/с

-s Время_начала

определяет время начала действия ограничения, например, 00:01:02

-e Время_конца

определяет время окончания действия ограничения, например, 23:59:59 Значение Время_конца должно быть всегда больше значения Время_начала, иначе ограничение работать не будет

-x ID

Удалить из конфигурации клиентских ограничений ограничение с идентификатором ID

-z ID

Удалить из конфигурации ограничений правил глобального расписания ограничение с идентификатором ID.



Используйте `rb_bandwidth -L` чтобы получить значение идентификатора

rb_block_device_check

rb_block_device_check — утилита администратора RuBackup для проверки целостности резервных копий на блочном устройстве.

-h

Справочное сообщение

-s <device_signature> -lu

Указать уникальную цифровую подпись блочного устройства, выданную СРК при регистрации устройства

-l

Вывод списка всех резервных копий на блочном устройстве, в которых присутствуют поврежденные блоки

-u

Обновить информацию о поврежденных блоках на сервере

rb_block_devices

rb_block_devices — утилита администратора RuBackup для управления блочными устройствами RuBackup. Вы можете просматривать, добавлять и удалять блочные устройства в конфигурации RuBackup.

-h

Справочное сообщение

-l [-v]

Список блочных устройств в конфигурации. Параметр -v предоставляет информацию в расширенном формате

-a block_device p pool_ID [-w parallel_workers] [-o] [-d описание]

Добавление блочного устройства с именем block_device (блочное устройство) и, при необходимости, соответствующим описанием, в пул хранения с ID_пула.

-o

позволяет перезаписать файловую систему на блочном устройстве

-c ID -w parallel_workers

Изменение количества параллельных обработчиков

-c ID -d description

Изменение описания. Если нужно добавить значение с пробелами используйте одинарные кавычки, например, 'Новое блочное устройство'

-c ID -n block_device

Изменение имени блочного устройства

-i ID

Увеличение размера блочного устройства в конфигурации, если его фактический размер был увеличен

-r ID

Удаление блочного устройства с идентификатором ID из конфигурации.

-o

Перезапись существующей файловой системы на блочном устройстве.



Используйте *rb_block_devices -l* для получения значения идентификатора блочного устройства

rb_client_defined_storages

rb_client_defined_storages — утилита администратора RuBackup для управления клиентскими хранилищами. Вы можете просматривать, добавлять и удалять клиентские хранилища в конфигурации RuBackup.

-h

Справочное сообщение

-l

Вывод списка клиентских хранилищ, созданных в СРК

-a storage_name -p pool_ID [-P metadata_pool_ID] [-m method] [-d 'some description']

Добавление нового хранилища, определяемое клиентом, которое также может использоваться несколькими клиентами

-r ID

Удаление клиентского хранилища с идентификатором ID

-c ID [-P new_metadata_pool_ID] [-d 'some description']

файловый пул и по умолчанию будет выставлен пул с минимальным ID. Можно изменить только имя хранилища и описание



Эта утилита не создает хранилище на клиенте и не проверяет, существует ли хранилище на клиенте.

rb_client_group

rb_client_group — утилита администратора RuBackup для управления группами клиентов RuBackup. Вы можете просматривать группы клиентов, добавлять их, удалять или изменять их название и описание. Группировка клиентов может потребоваться в случае необходимости выполнения групповых операций резервного копирования в стратегиях RuBackup.

-h

Справочное сообщение

-l

Список пользовательских групп

-a Имя_группы [-d 'Описание']

Добавить в конфигурацию RuBackup группу с именем Имя_группы и, при необходимости, соответствующим ей описанием. Если нужно добавить значение с пробелами используйте одинарные кавычки, например, 'Новая группа'

-r ID

Удалить из конфигурации группу с идентификатором ID.

-s ID [-n Новое_имя_группы] [-d 'Новое описание']

Изменить имя группы с идентификатором ID на новое имя Новое_имя_группы и, при необходимости, изменить описание группы.



Используйте *rb_client_group -l* для получения значения идентификатора группы

rb_clients

rb_clients — утилита администратора RuBackup для управления клиентами RuBackup. Вы можете просматривать список клиентов, а также добавлять, удалять или изменять их. При первоначальной настройке системы резервного копирования нет необходимости специально добавлять клиентов в серверную группировку с помощью параметра `-a`. Достаточно запустить клиентский процесс RuBackup на хосте клиента с правильными адресами основного и резервного (при наличии) серверов в конфигурационном файле. При первом запуске клиент будет зарегистрирован в системе как неавторизованный клиент. С помощью вызова `rb_clients -t` необходимо авторизовать клиента для того, чтобы было возможно выполнять резервное копирование и восстановление данных клиента.

В том случае, если IP или MAC адрес клиента изменится, то сервер автоматически изменит их в базе данных RuBackup, что отразится в файле журнала.

Основным идентификатором клиента в базе данных RuBackup является его имя хоста.

Клиент имеет право размещать в системе резервного копирования резервные копии, общий объём которых не должен превышать значение заданной емкости хранилища (в ГБ).

Чтобы восстановить или удалить резервную копию клиент должен предоставить серверу пароль. Первый раз пароль устанавливается при запуске RBC или утилиты `rb_archives`, если в этот момент в базе данных RuBackup не содержится информация о пароле клиента. Чтобы сбросить пароль клиента используйте команду `rb_clients -y`.

-h

Справочное сообщение

-l [-o] [-v]

Список авторизованных клиентов.

-o

выдаёт информацию только о клиентах, которые работают в данный момент времени

-v

предоставляет расширенную информацию

-u [-o] [-v]

Список неавторизованных клиентов

-o

Выдаёт информацию только о клиентах, на которых в данный момент времени запущен клиентский процесс RuBackup

-v

предоставляет расширенную информацию

```
-a hostname -H hwid [-g group_ID] [-p pool_ID] [-s storage_capacity] [-w  
lsf_allow_deny] [-z lrl_allow_deny] [-B local_backup_allow_deny] [-R  
local_restore_allow_deny] [-4 ipv4] [-6 ipv6] [-m macaddr] [-d 'some  
description'] (add new client)
```

Добавить в конфигурацию RuBackup нового клиента.

-g

позволяет соотнести клиента с клиентской группой

-p

определяет расположение или путь, где будет храниться резервная копия

-w allow

используйте параметр если необходимо разрешить клиенту исполнять правила локального расписания

-w deny

используйте параметр если необходимо запретить клиенту исполнять правила локального расписания

-s

определяет максимальный объём всех резервных копий клиента в системе резервного копирования.

-4

определяет IP адрес v4

-6

определяет IP адрес v6

-m

определяет MAC адрес

Локальные правила резервного копирования для клиента находятся в файле `/opt/rubackup/etc/rubackup.lsf`. Если на клиенте имеются какие-либо файлы, которые не должны попасть в резервные копии, то необходимо соответствующим образом настроить локальный список запретов, который содержится в файле `/opt/rubackup/etc/rubackup_restriction.list`. Локаль-

ный список запретов представляет собой простой список файлов, которые не должны попасть в резервную копию. При составлении списка можно использовать регулярные выражения, например, `.png` будет означать что все файлы с расширением `png` будут исключены из резервной копии.

-z allow

используйте параметр чтобы разрешить клиенту использовать локальный лист запретов

-z deny

используйте параметр чтобы запретить клиенту использовать локальный лист запретов

-v

Выводит список авторизованных клиентов в расширенном формате

-r ID

Удалить авторизованного клиента с идентификатором ID из конфигурации RuBackup

-x ID

Удалить неавторизованного клиента с идентификатором ID из списка неавторизованных клиентов

-t ID

Авторизовать неавторизованного клиента с идентификатором ID

-c ID [-d 'new description'] [-g group_ID] [-p pool_ID] [-s storage_capacity] [-w lsf_allow_deny] [-z lrl_allow_deny] [-B local_backup_allow_deny] [-R local_restore_allow_deny] (change client)

Изменить описание, группу, пул для резервных копий инициированных локальным расписанием, максимальный объем резервных копий клиента (в ГБ), разрешения выполнять локальное расписание и учёт локального списка запретов для клиента с идентификатором ID

-y ID

Сбросить пароль клиента с идентификатором ID

-k ID -M

Вывести список параметров восстановления модуля

rb_cloud_task_queue

rb_cloud_task_queue — утилита администратора RuBackup для просмотра задач, которые связаны с облачными операциями. При хранении резервных копий в облаке S3 вам может потребоваться загрузить резервную копию в облако или выгрузить какой-либо из файлов резервной копии из облака.

-h

Справочное сообщение

-l

Получить информацию о всех задачах

-e

Удалить ошибочные задачи

rb_clouds

rb_clouds — утилита администратора RuBackup для просмотра конфигурации, добавления или удаления облаков S3 в системе резервного копирования RuBackup.

-h

Справочное сообщение

-v

Получить информацию о конфигурации всех облаков S3. Предоставляет расширенную информацию

```
-a cloud_name -p pool_ID -k access_key -s secret_key_access [-e  
endpoint_override] [-t proxy_host] [-p proxy_port] [-u proxy_username] [-w  
proxy_password] [-m region] [-b bucket_name] [-o allow auto bucket creation]
```

Добавить в конфигурацию облако и ассоциировать его с пулом pool_ID. Дополнительные параметры позволяют задать параметры подключения к частному облаку.

-e

задает адрес S3 объектного хранилища

-t

задает прокси хост S3 объектного хранилища

-p

задает прокси порт S3 объектного хранилища

-u

задает имя пользователя

-w

задает пароль пользователя для прокси хоста S3 объектного хранилища.

-m

задает регион S3 объектного хранилища

-b

задает имя для бакета

-j

задает количество потоков передачи данных в облако

-o

определяет может ли СРК самостоятельно создавать бакеты в S3 объектном хранилище



С каждым пулом может быть ассоциировано только одно облако.

-r ID

Удалить облако из конфигурации

```
-c ID [-n_name] [-k access_key] [-s secret_key_access] [-e endpoint_override] [-t proxy_host] [-x proxy_port] [-u proxy_username] [-w proxy_password] [-m region] [-b bucket_name] [-o allow_auto_bucket_creation]
```

-c

позволяет редактировать параметры ранее заданных данных для S3 объектных хранилищ, является идентификатором S3 объектного хранилища для редактирования

-n

задает новое имя S3 объектного хранилища

-k

задает новый ключ S3 объектного хранилища

-s

задает новый секретный ключ S3 объектного хранилища

-e

задает новый адрес S3 объектного хранилища

-t

задает новый прокси хост S3 объектного хранилища

-x

задает новый прокси порт S3 объектного хранилища

-u

задает новое имя пользователя для прокси хоста S3 объектного хранилища

-w

задает новый пароль пользователя для прокси хоста S3 объектного хранилища

-m

задает новый регион S3 объектного хранилища

-b

задает новое имя бакета для хранения резервной копии S3 объектного хранилища

-j

задает количество потоков передачи данных в облако

-o

определяет может ли сервер СРК самостоятельно создавать бакеты для хранения резервной копии в S3 объектном хранилище

rb_copy2pool

rb_copy2pool — утилита администратора RuBackup для управления репликацией.

rb_copy2pool предоставляет возможность создавать точные копии (реплики) резервных копий для правил резервного копирования и для стратегий резервного копирования. Реплики резервных копий могут располагаться в других пулах, нежели пул назначения, определяемый правилом резервного копирования или стратегией резервного копирования. Это обеспечивает возможность после создания резервной копии сделать нужное количество реплик, которые будут располагаться в других местах хранения. Пул для репликации может располагаться как на том медиасервере, на котором располагается оригинальная резервная копия, так и на другом медиасервере серверной группировки RuBackup.

Для каждого правила резервного копирования или стратегии резервного копирования может быть любое количество правил репликации, но не более чем общее количество пулов в системе резервного копирования.

-h

Справочное сообщение

-l

Список пулов, в которых будут созданы реплики резервных копий для правил глобального расписания системы резервного копирования

-L

Список пулов, в которых будут созданы реплики резервных копий для стратегий системы резервного копирования

-a ID_правила -p ID_пула [-c capacity] [-P period] [-s window_start -e window_end]

Добавить пул с идентификатором ID_пула в список пулов для репликации для правила ID_правила

-c

задает процент заполнения исходного пула, при достижении которого будет создана копия всех резервных копий, созданных со времени последнего срабатывания данного правила репликации

-P

задает период от момента создания правила, спустя который будет создана копия всех резервных копий, созданных со времени последнего срабатывания данного правила репликации

-s

указывает время начала рабочего окна

-e

указывает время окончания рабочего окна (рабочее окно — интервал, в течение которого выполняется копирование резервной копии)

-A ID_стратегии -p ID_пула [-c capacity] [-P period] [-s window_start -e window_end]

Добавить пул с идентификатором ID_пула в список пулов для репликации для стратегии ID_стратегии

-c

задает процент заполнения исходного пула, при достижении которого будет создана копия всех резервных копий, созданных со времени последнего срабатывания данного правила репликации

-P

задает период от момента создания правила, спустя который будет создана копия всех резервных копий, созданных со времени последнего срабатывания данного правила репликации

-s

указывает время начала рабочего окна

-e

указывает время окончания рабочего окна (рабочее окно — интервал, в течение которого выполняется копирование резервной копии)

-r ID

Удалить запись из пулов репликации правил резервного копирования

-R ID

Удалить запись из пулов репликации стратегий резервного копирования

rb_dbms

rb_dbms – утилита администратора RuBackup для просмотра расширенного вывода информации о резервных копиях для СУБД, такой как:

- *record_id* - id резервной копии
- *client_hostname* - имя (хост) клиента, с которого сделана резервная копия
- *client_hwid* - уникальный айди клиента, описанный в официальной документации
- *resource* - ресурс резервной копии
- *backup_type* - тип резервной копии (full, incremental)

Для PostgreSQL:

- *incremental_subtype* - инкрементальный подтип. Колонка может принимать значения *archive_wal*, *page*, *delta*, *ptrack* для резервных копий сделанных в соответствующих режимах
- *TLI* - timeline id начального и конечного WAL файла в резервной копии, разделенный символом /
- *WAL* - количество WAL файлов в резервной копии, выражается в байтах, килобайтах, мегабайтах и тд с соответствующими приставками
- *START_LSN* - начальный LSN при вызове *pg_start_backup()*
- *STOP_LSN* - конечный LSN при вызове *pg_stop_backup()*

Значение в колонке может быть **None** и подсвечиваться красным цветом, если СРК не имеет информации для соответствующей резервной копии

-h

Справочное сообщение

-t show -r postgresql

Вывод информации о СУБД PostgreSQL

rb_dynamic_group_pools

rb_dynamic_group_pools – утилита администратора RuBackup для управления динамическими группами пулов RuBackup. Вы можете просматривать, добавлять и удалять динамические группы пулов в конфигурации RuBackup, управлять их параметрами и управлять их содержимым. Динамические группы пулов используются для автоматической балансировки задач системы резервного копирования между пулами в зависимости от установленных на группу ограничений. В том случае, если задача резервного копирования изначально должна была использовать определенный пул, но этот пул загружен более чем по одному из параметров (количество задач на пул, количество задач на медиа-сервер, максимальная утилизация CPU, максимальное количество операций ввода-вывода хоста), и этот пул включен в динамическую группу, то система резервного копирования подменит этот пул на наименее загруженный из числа пулов динамической группы. Таким образом, задача может быть автоматически направлена на другой пул, в том числе и на другой медиа-сервер.

-h

Справочное сообщение

-l [-j]

Выводит список динамических групп пулов и их параметров.

-j

предоставляет информацию в json формате. Вывод содержит информацию об имени группы пулов и условия переключения задачи на другой пул: максимальное количество задач на пул, максимальное количество задач на медиа-сервер, максимальная загрузка CPU хоста медиа-сервера в процентах, максимальное количество операций ввода дисковой подсистемы хоста медиа-сервера, максимальное количество операций вывода дисковой подсистемы хоста медиа-сервера, а так же какой период в минутах для расчета этих условий будет использован. Кроме этого динамическая группа пулов может сопровождаться расширенным описанием

-L id [-j]

Выводит информацию о пулах, находящихся в динамической группе с идентификатором «id». Если в качестве «id» использовать all, то будет выведена информация о всех пулах для всех имеющихся динамических групп

-j

предоставляет информацию в json формате

-a group_name [-t max_pool_tasks] [-m max_media_server_tasks] [-u max_cpu_usage] [-i max_in_operations] [-o max_out_operations] [-C calculation_period_min] [-d group_description]

Добавляет динамическую группу с заданным именем «name» и набором параметров

-t max_pool_tasks

максимальное количество одновременных задач в пуле

-m max_media_server_tasks

максимальное количество задач на медиасервере

-u max_cpu_usage

максимальная загрузка процессора на медиасервере в процентах

-i max_in_operations

максимальное количество операций ввода дисковой подсистемы хоста медиасервера

-o max_out_operations

максимальное количество операций вывода дисковой подсистемы хоста медиасервера

-C calculation_period_min

период времени в минутах для расчета показателей пула

-d group_description

описание динамической группы пулов

-r Group_ID

Удаляет динамическую группу с идентификатором «Group_ID»

-c Group_ID [-n group_name] [-t max_pool_tasks] [-m max_media_server_tasks] [-u max_cpu_usage] [-i max_in_operations] [-o max_out_operations] [-C calculation_period_min] [-d group_description]

Редактирует динамическую группу пулов с идентификатором «Group_ID» и набором параметров группы

-n group_name

новое имя группы

-t max_pool_tasks

максимальное количество одновременных задач в пуле

-m max_media_server_tasks

максимальное количество задач на медиасервере

-u max_cpu_usage

максимальная загрузка процессора на медиасервере в процентах

-i max_in_operations

максимальное количество операций ввода дисковой подсистемы хоста медиасервера

-o max_out_operations

максимальное количество операций вывода дисковой подсистемы хоста медиасервера

-C calculation_period_min

период времени в минутах для расчета показателей пула

-d group_description

описание динамической группы пулов

-A Group_ID -p Pool_ID

Добавляет пул (параметр «-p») с идентификатором «Pool_ID» в группу с идентификатором «Group_ID»

-R Group_ID -p Pool_ID

Удаляет пул (параметр «-p») с идентификатором «Pool_ID» из группы с идентификатором «Group_ID»

-E Group_ID -p Pool_ID

Включает пул (параметр «-p») с идентификатором «Pool_ID» в группе с идентификатором «Group_ID»

-D Group_ID -p Pool_ID

Отключает пул (параметр «-p») с идентификатором «Pool_ID» в группе с идентификатором «Group_ID»

rb_global_config

rb_global_config — утилита администратора RuBackup для управления параметрами глобальной конфигурации серверной группировки RuBackup. Параметры глобальной конфигурации действительны для всех серверов, входящих в кластер серверов RuBackup.

-h

Справочное сообщение

-l

Вывод всех имеющихся параметров с их значениями

-o Параметр

Вывод значения Параметра

-s Параметр=Значение

Установить Значение для Параметра



Утилита `rb_global_config` не производит проверку правильности значений. Сверьтесь с документацией RuBackup сервера о допустимости использования тех или иных значений. Неправильные значения параметров могут изменить работу системы резервного копирования нежелательным для вас образом.

rb_global_schedule

rb_global_schedule — утилита администратора RuBackup для управления глобальным расписанием RuBackup. Глобальное расписание состоит из отдельных правил, которые могут выполняться по определённым условиям для определённого ресурса на клиенте системы резервного копирования. При помощи **rb_global_schedule** можно просматривать список правил глобального расписания, экспортировать настройки правила в файл и импортировать правило из файла в глобальное расписание, удалять правила из глобального расписания, останавливать функционирование правила или запускать его в работу, а также немедленно создавать задачу на основе правила глобального расписания.

Система резервного копирования иницирует новые задачи на основании временных параметров запуска правил глобального расписания. В качестве параметров старта правила вы можете определить минуту, час, день месяца, месяц и день недели.

Правило может быть приостановлено (статус `wait`) и запущено в работу (статус `run`). Приостановленное правило не будет иницировать создание задач, но, при необходимости, можно выполнить срочную задачу, используя параметр `-x`.

Свойства правила определяют тип резервного копирования (`full`, `incremental` или `differential`), тип ресурса, ресурс (файл, директория, виртуальная машина, база данных и т.д.), для которого будет создана резервная копия, необходимость защитного преобразования резервной копии, продолжительность хранения резервной копии, необходимость и условия перемещения резервных копий правила, их удаления или верификации, а также определяет кому будут высланы уведомления по тому или иному статусу.

-h

Справочное сообщение

-a

Список правил глобального расписания и их расширенные параметры

-l

Вывод в консоль всех созданных правил глобального расписания

-s

Вывод временных данных всех правил глобального расписания

-n

Список правил глобального расписания и параметры уведомления

-X ID

Список дополнительных параметров для правила глобального расписания

-i Файл_импорта

Импорт правила из заранее подготовленного файла

-e ID [-f Файл_экспорта]

Экспорт параметров правила в вывод терминала.

-f

позволяет выполнить экспорт в файл

-d ID

Удалить правило с идентификатором ID из глобального расписания

-w ID

Приостановить исполнение правила с идентификатором ID

-r ID

Возобновить исполнение правила с идентификатором ID

-x ID

Немедленно создать задачу на основании правила с идентификатором ID

-C ID -E Параметр:Значение, Параметр1:Значение1, ...

Изменить дополнительные параметры для правила с идентификатором ID. При этом возможно изменить сразу несколько параметров в виде Параметр:Значение, перечислив их через запятую

Примеры создания правила

Таблица 3. Пример создания нового правила

Название параметра	Описание параметра	Значения параметра
Name: New rule name	Название правила (пример шаблона правила для модуля File system).	
Client: clienthostname (client_hwid)	Имя клиента (hostname) и (hardware id) (hwid).	
Pool: Default	Имя пула, в котором будут располагаться резервные копии.	
Storage capacity: 50	Суммарный разрешенный размер всех резервных копий правила в ГБ.	

Название параметра	Описание параметра	Значения параметра
Minute: 0	Условия старта. означает все значения минуты(0-59). В данном случае правило создаст задачу в 0 минут при соблюдении остальных условий старта.	
Hour: 0	Часы(0-23). В данном случае правило создаст задачу в 0 часов при соблюдении остальных условий старта.	
Day of month: *	День месяца(1-31). В данном случае правило будет создавать задачу каждый день при соблюдении остальных условий старта.	
Month: *	Месяц (1-12). В данном случае правило будет создавать задачу каждый месяц при соблюдении остальных условий старта.	
Day of week: *	День недели (0-7, 1 - понедельник, 0 и 7 - воскресенье). В данном случае правило будет создавать задачу каждый день недели при соблюдении остальных условий старта.	
Validity start period: 2019-06-17 18:15:00+03	Начало действия правила принимает формат YYYY-MM-DD hh:mm:ss ±hh.	
Validity end period: 2020-06-17 18:15:00+03	Окончание действия правила принимает формат YYYY-MM-DD hh:mm:ss ±hh.	
Storage duration: 2 days	Продолжительность хранения резервных копий (days, weeks, months, years). Регистр не учитывается.	Допустимы значения (day(s) , week(s) , month(s) , year(s)).
Verify flag: true	Проверять или не проверять резервные копии после их создания (true, false). Регистр не учитывается.	Допустимы значения: t , true , y , yes , on , 1 , – для значения true ; f , false , n , no , off , 0 – для значения false .
Verify interval: 1 day	Интервал проверки резервных копий (days, weeks, months, years). Регистр не учитывается.	Допустимы значения (day(s) , week(s) , month(s) , year(s)).
Resource type: File system	Тип ресурса (см.актуальный список типов ресурсов и их названия в вашей лицензии).	

Название параметра	Описание параметра	Значения параметра
Resource: /home/andreyk/ST/	Ресурс (файл, директория, виртуальная машина, база данных и пр.) в зависимости от типа ресурса.	Ресурс задается строкой, для filesystem — это путь к файлу (папке), для postgresql — это строка PostgreSQL <version>, например: Resource: PostgreSQL 12.14.
Backup type: full	Тип резервной копии (full, incremental, differential).	Регистр учитывается, если написать не в нижнем регистре, подставится значение по умолчанию.
Crypto: kuznyechik	Алгоритм защитного преобразования резервных копий.	актуальный перечень алгоритмов (см. rbfd) Регистр учитывается, если написать не в нижнем регистре, подставится значение по умолчанию.
Status: run	Статус правила.	Допустимые значения: wait, run.
Auto delete obsoleted archive: true	Удалять или не удалять резервные копии, срок хранения которых истек (true, false). Регистр не учитывается.	Допустимы значения: t, true, y, yes, on, 1, — для значения true; f, false, n, no, off, 0 — для значения false.
Who will be informed when archive is obsoleted: RuBackup administrators	Группа пользователей (см. rb_user_groups), которые будут проинформированы о наличии устаревшей резервной копии или об ее удалении.	
Auto moving archive: true	Перемещать ли резервную копию автоматически в другой пул по достижении определенного срока ее хранения (true, false). Регистр не учитывается.	Допустимы значения: t, true, y, yes, on, 1, — для значения true; f, false, n, no, off, 0 — для значения false.
Move to pool: TL pool	Пул, в которой будут автоматически перемещены резервные копии по достижении определенного срока хранения (см. rb_pools).	
Move archive if older than: 1 day	Срок, по истечении которого, резервные копии правила могут быть перемещены в другой пул.	
Notify normal: RuBackup administrators	Группа пользователей (см. rb_user_groups), которые будут проинформированы об успешном выполнении задачи резервного копирования данного правила.	

Название параметра	Описание параметра	Значения параметра
Notify normal CC: you@mail.ru	Отправить копию по адресу в случае успешного выполнении задачи резервного копирования.	
Normal execution script:	Выполнить скрипт.	
Notify exception: Nobody	Группа пользователей (см. rb_user_groups), которые будут проинформированы об ошибочном выполнении задачи резервного копирования данного правила.	
Notify exception CC:	Отправить копию по адресу в случае ошибочного выполнении задачи резервного копирования.	
Exception execution script:	Выполнить скрипт в случае ошибочного выполнения задачи резервного копирования.	
Notify verify: Nobody	Группа пользователей (см. rb_user_groups), которые будут проинформированы о результатах проверки резервной копии	
Notify verify CC:	Отправить копию.	
Notify if end of rule validity: Nobody	Группа пользователей (см. rb_user_groups), которые будут проинформированы об окончании срока действия правила.	
Notify if end of rule validity CC:	Отправить копию.	
Notify if end of storage capacity: Nobody	Группа пользователей (см. rb_user_groups), которые будут проинформированы об окончании квоты, выделенной для хранения резервных копий данного правила.	
Notify if end of storage capacity CC:	Отправить копию.	
Allow the client to remove archives: false	Разрешено ли клиенту создавать задачу для удаления своих резервных копий (true, false). Регистр не учитывается.	Допустимы значения: t, true, y, yes, on, 1, – для значения true; f, false, n, no, off, 0 – для значения false.
Priority: 100	Приоритет выполнения правила.	

Название параметра	Описание параметра	Значения параметра
Periodic launch:	Период выполнения правила (в минутах).	
Restore script:	Путь к скрипту восстановления.	

Таблица 4. Пример создания нового правила для модуля Universal

Название параметра	Описание параметра	Значения параметра
connection_monitoring:t	Мониторинг соединения с СУБД (параметр используется только для подмодуля postgresql).	Возможные значения true (t), false (f).
engine:postgresql	Выбор подмодуля для резервного копирования.	Возможные варианты: postgresql, pg_probackup, superb.
incremental_subtype:archival	Подтип инкрементального резервного копирования (параметр используется только для подмодуля postgresql).	
snapshot_type:lvm	Тип снимка (параметр используется только для подмодуля superb).	Возможные значения: lvm, dattobd.
snapshot_size:10	Размер мгновенного снимка состояния в % (параметр используется только для подмодуля superb).	
entire_snapshot_backup:f	Выполнять резервное копирование мгновенного снимка состояния СУБД.	Возможные значения true (t), false (f).
pg_pro_threads:1	Количество параллельных потоков при резервном копировании (параметр используется только для подмодуля pg_probackup).	
pg_pro_backup_mode:PTRACK	Подтип инкрементального резервного копирования (параметр используется только для подмодуля pg_probackup).	
pg_pro_stream:t	Режим доставки WAL (параметр используется только для подмодуля pg_probackup).	Возможные значения true (t), false (f).
secret_method:3	Метод получения аутентификационных данных (секрета) для подключения к резервируемой СУБД.	

Название параметра	Описание параметра	Значения параметра
worker_parallelism:8	Количество потоков, которые будут участвовать в процессе восстановления блоков данных ресурса.	
enable_multithreading:f	Многопоточная передача данных. Используются сетевые потоки в количестве, указанном в параметре network_parallelism.	Возможные значения true (t), false (f).
enable_flexible_dedup:f	Использование нескольких таблиц дедупликации вместо одной.	Возможные значения true (t), false (f).
network_parallelism:8	Количество потоков, которые будут передавать блоки данных на медиасервер.	
memory_threshold:0	Верхняя граница использования оперативной памяти (в Гб) при восстановлении резервной копии.	
deny_memory_exceed:f	Выполнение резервного копирования при превышении заданной верхней границы использования оперативной памяти. Используется в случае, если предыдущий инкрементальный бэкап был сделан версией СРК RuBackup ранее 2.1.1-a.84, а текущая работа производится в версии 2.1.1-a.84 и выше.	Возможные значения true (t) - резервное копирование прекратится с ошибкой, false (f) - будет показано предупреждение и резервное копирование будет продолжено.
deduplication:t	Исключение дублирующих копий повторяющихся данных.	Возможные значения true (t), false (f).

rb_health_check

rb_health_check — утилита клиента RuBackup для проверки конфигурации клиента и его окружения. Выполняется проверка переменных окружения, версии медиа-сервера. Проверяется подключение клиента к базе данных, серверу, медиасерверу и толстому клиенту. Утилита так же проверяет подключение к Postgres, параметры настроек и наличие свободного пространства, где располагается Postgres, а так же проверка свободного пространства в пулах и наличие хранилищ в пулах.

После запуска команды выводится таблица с текущими состояниями проверяемых систем. Варианты вывода информации:

- Info - выводится зеленым цветом – ОК
- Warning - выводится желтым цветом - необходимо обратить внимание и проверить состояние системы
- Error - выводится красным цветом - ошибка, необходима проверка

rb_init

rb_init - утилита администратора RuBackup для первоначальной настройки клиента или сервера RuBackup сразу после развёртывания пакета исполняемых файлов RuBackup на клиенте или сервере. Утилита **rb_init** создаёт главный конфигурационный файл `/opt/rubackup/etc/config.file` с параметрами, которые определяет пользователь. В ходе работы утилиты могут быть созданы мастер-ключ клиента RuBackup, а также приватный и открытый ключи цифровой подписи клиента RuBackup. В том случае, если происходит настройка основного (primary) сервера RuBackup, может быть создана база данных RuBackup в СУБД PostgreSQL.

Утилита позволяет сделать апгрейд (`rb_update`) и дамп базы данных перед апгрейдом.

Пользователь перед удалением базы данных имеет возможность сделать дамп базы перед ее удалением, если произведен выбор такого сценария.

При создании дампа базы данных RuBackup имеется возможность выбрать директорию для файла дампа, а также формат дампа. На выбор доступны два формата: `plain` (текстовый sql скрипт), `custom` (custom-архив, восстановление выполняется с помощью `pg_restore`). Дамп в `custom` формате занимает меньше места на диске по сравнению с `plain` форматом. Также для `custom` формата доступен выбор степени сжатия (0-9), чем выше степень сжатия, тем меньше `custom`-архив занимает места на диске, и тем дольше выполняется процедура дампа базы данных.

В том случае, если на хосте установлен только клиент, то серверные ветки не предлагаются.

Более подробно о процедуре установки RuBackup см. «Руководство по установке серверов резервного копирования и Linux клиентов RuBackup».

rb_init (неинтерактивный режим)

rb_init non-interactive mode Неинтерактивный режим необходим для сценариев массового развёртывания, например, при использовании `ansible`. В этом режиме администратор имеет возможность настраивать RuBackup в `bash/shell` однострочной командой и, как следствие, использовать эту команду в скриптах для автоматизации процесса.

-h

Справочное сообщение

-y

--agree-with-eula

Вы должны указать эту опцию для входа в неинтерактивный режим (автоматизации процесса).

тическое принятие условий лицензионного соглашения)

-n node type

Определяет тип конфигурируемого узла

[-P parallelizm-num]

Определяет количество потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое не отрицательное число не равное нулю

[-v]

Подробный вывод сервисных процессов в терминал и логи

Конфигурация основного сервера

--audit

Включение / выключение журналирования события ИБ

--audit-for-tasks

Включается дополнительно для журналирования задач (используется совместно с опцией `--audit`)

[-H dbhost]

Указатель на сервер, где располагается база данных rubackup.

Аргумент

следует указать hostname сервера, где располагается/будет располагаться база данных rubackup (не рекомендуется использовать ip сервера). Если не указать этот параметр, будет использоваться значение по-умолчанию — `localhost`

[-N dbname]

Определяет имя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

[-U dbuser]

Определяет имя суперпользователя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

-X dbpasswd-postgres

Пароль для пользователя postgres (СУБД postgres содержащей БД rubackup).

Аргументы`postgres_user_passwd`**-Y dbpasswd-rubackup**

Пароль для пользователя rubackup (владельца БД rubackup). Необходим для создания пользователя rubackup.

Аргументы`rubackup_user_passwd.`**[--disable-dump]**

При определении этой опции, если база данных rubackup уже существует при выполнении `rb_init`, дамп базы данных rubackup не будет создан. Если не указать этот параметр, то дамп базы данных будет создан

[--dump-format]

Определяет формат дампа базы данных rubackup.

-c

будет использоваться значение по-умолчанию, если не указать параметр.

Доступные значения

`p` (plain, текстовый sql скрипт), `c` (custom-архив, восстановление выполняется с помощью `pg_restore`). Дамп в custom формате занимает меньше места на диске, по сравнению с `plain` форматом

[--dump-compression-level level]

Для custom формата дампа базы данных rubackup доступен выбор степени сжатия (0-9), чем выше степень сжатия, тем меньше custom-архив занимает места на диске, и тем дольше выполняется процедура дампа базы данных. Если не указать этот параметр, будет использоваться значение по-умолчанию — `1`.

Доступные значения

число от 0 до 9

[--dump-dir-path path]

Определяет папку, в которую сохранится дамп базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `..`. При пересоздании и update БД производится бэкап устаревшей версии БД при помощи утилиты `pg_dump`, если не установлен параметр `--disable-dump`

[--ssl-mode mode]

Выбрать `ssl-mode` для определения степени защиты соединения. Доступные варианты режимов: `disable`, `allow`, `prefer`, `require`, `verify-ca`, `verify-full`

[--ssl-root-cert path]

Указать путь к корневому сертификату

[--ssl-cert path]

Указать путь к сертификату клиента

[--ssl-key path]

Указать путь к приватному ключу клиента.

-i iface

Сетевой интерфейс клиента для связи с севером.

Аргументы

Укажите имя клиентского сетевого интерфейса (напр. `Eth0`)

-f path

Определение директории для Default пула. (создает директорию, если она не была создана ранее).

Аргументы

необходимо указать путь и имя директории (напр. `/default-pool`)

[-l local-backup-directory]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Если при конфигурации не был указан флаг `-l` и аргумент к нему, то в конфигурационном файле `/opt/rubackup/etc/config.file`, в параметре `use-local-backup-directory` будет использоваться значение по-умолчанию — `/tmp`. В ином случае будет использоваться директория, которая была указана при конфигурации.

Аргумент

укажите абсолютный путь к директории (напр. `RuBackup-tmp`)

[-O if-db-exists-opt]

Опция используется при реконфигурации основного сервера (если база данных rubackup существует) и указывает, что необходимо сделать с БД. Если не указать этот параметр, будет использоваться значение по-умолчанию — `upgrade`.

Аргументы

`cancel` (отменить дальнейшую конфигурацию) | `drop` (пересоздать БД) | `upgrade` (апдейт существующей БД)

[-S secondary-server]

Указатель на расположение дополнительного сервера резервного копирования.

Аргументы

укажите FQDN дополнительного сервера (не рекомендуется использовать ip адрес)

[-p parallel-tasks]

Возможность ограничить параллельное выполнение заданий. Если не указать этот параметр, будет использоваться значение по-умолчанию — 2.

Аргумент

целое число ≤ 2

[-D dedup-memory-size]

Указывает объем памяти дедупликации для задачи в байтах (268435456 байт по умолчанию).

Аргумент

неотрицательное целое число

[-r]

Разрешает выполнять удаленную репликацию для клиента

[-R]

Разрешает выполнять централизованное восстановление для ресурсов клиента

[-g]

Создает ключи электронно-цифровой подписи (ЭЦП) и дает возможность клиенту проверять целостность резервных копий с их помощью

[-P parallelizm-num]

Определяет количеств потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое не отрицательное число не равное нулю

[-v]

Подробный вывод сервисных процессов в терминал и логи

[-m]

Опция включает мониторинг клиента

Конфигурация автономного клиента**-a archive-catalog**

Определяет клиента, как автономный.

Аргументы

следует указать директорию для хранения архивов резервных копий (прим. `/rubackup-backups`). Если на клиенте предполагается, в качестве хранилища РК, использоваться внешнее устройство или сетевую шару, то директория будет выступать в качестве точки монтирования

[-z compressor]

Выбор метода компрессии архивов резервных копий для автономного клиента (по умолчанию `pigz`).

Аргументы

`none, gzip, pigz, xz`

[-d dedicated-device-or-network-share]

Опция определяет, что мы хотим использовать в качестве хранилища внешнее устройство (USB/внешний SSD) или сетевую шару.

Аргументы

указываем путь к внешнему накопителю (прим. `/dev/sdb`) или сетевой шаре (прим. `srv://net_share`)

[-s network-share-type]

Определяет тип сетевой шары и используется вместе с опцией `-d` (используется клиентом для команды `mount -t`).

Аргументы

`nfs` или `cifs`

[-o mount-option]

При необходимости указывает опции монтирования. Используется вместе с опцией `-d`.

Аргументы

прим. `rw, sync` (опции перечисляются через запятую, без пробелов)

[-l local-backup-directory]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Если при конфигурации не был указан флаг `-l` и аргумент к нему, то в конфигурационном файле `/opt/rubackup/etc/config.file`, в параметре `use-local-backup-directory` будет использоваться значение по-умолчанию — `/tmp`. В ином случае будет использоваться директория, которая была указана при конфигурации.

Аргумент

укажите абсолютный путь к директории (напр. `RuBackup-tmp`)

[-P parallelism-num]

Определяет количество потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое не отрицательное число не равное нулю

[-v]

Подробный вывод сервисных процессов в терминал и логи

Конфигурация клиент (клиент-серверная архитектура)**-c primary-server**

Указатель на основной сервер.

Аргументы

следует указать FQDN сервера (не рекомендуется использовать ip адреса)

-i iface

Сетевой интерфейс клиента для связи с севером.

Аргументы

Укажите имя клиентского сетевого интерфейса (напр. `Eth0`)

[-S secondary-server]

Указатель на расположение дополнительного сервера резервного копирования.

Аргументы

укажите FQDN дополнительного сервера (не рекомендуется использовать ip адрес)

[-l local-backup-directory]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Если при конфигурации не был указан флаг `-l` и аргумент к нему, то в конфигурационном файле `/opt/rubackup/etc/config.file`, в параметре `use-local-backup-directory` будет использоваться значение по-умолчанию — `/tmp`. В ином случае будет использоваться директория, которая была указана при конфигурации.

Аргумент

укажите абсолютный путь к директории (напр. `RuBackup-tmp`)

[-p parallel-tasks]

Возможность ограничить параллельное выполнение заданий. Если не указать

этот параметр, будет использоваться значение по-умолчанию — 2.

Аргумент

целое число ≤ 2

`[-D dedup-memory-size]`

Указывает объем памяти дедупликации для задачи в байтах (268435456 байт по умолчанию).

Аргумент

неотрицательное целое число

`[-r]`

Разрешает выполнять удаленную репликацию для клиента

`[-R]`

Разрешает выполнять централизованное восстановление для ресурсов клиента

`[-g]`

Создает ключи электронно-цифровой подписи (ЭЦП) и дает возможность клиенту проверять целостность резервных копий с их помощью

`[-P parallelizm-num]`

Определяет количество потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое неотрицательное число не равное нулю

`[-v]`

Подробный вывод сервисных процессов в терминал и логи

Конфигурация дополнительного сервера

`-c primary-server`

Указатель на основной сервер.

Аргументы

следует указать FQDN сервера (не рекомендуется использовать ip адреса).

`-H dbhost`

Указатель на сервер, где располагается база данных rubackup.

Аргумент

следует указать hostname сервера, где располагается/будет располагаться

база данных rubackup (не рекомендуется использовать ip сервера). Если не указать этот параметр, будет использоваться значение по-умолчанию — `localhost`

[-N dbname]

Определяет имя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

[-U dbuser]

Определяет имя суперпользователя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

-Y dbpasswd-rubackup

Пароль для пользователя rubackup (владельца БД rubackup). Необходим для создания пользователя rubackup.

Аргументы

`rubackup_user_passwd`

-i iface

Сетевой интерфейс клиента для связи с севером.

Аргументы

Укажите имя клиентского сетевого интерфейса (напр. `Eth0`)

[-l local-backup-directory]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Если при конфигурации не был указан флаг `-l` и аргумент к нему, то в конфигурационном файле `/opt/rubackup/etc/config.file`, в параметре `use-local-backup-directory` будет использоваться значение по-умолчанию — `/tmp`. В ином случае будет использоваться директория, которая была указана при конфигурации.

Аргумент

укажите абсолютный путь к директории (напр. `RuBackup-tmp`)

[-p parallel-tasks]

Возможность ограничить параллельное выполнение заданий. Если не указать этот параметр, будет использоваться значение по-умолчанию — `2`.

Аргумент

целое число ≤ 2

`[-D dedup-memory-size]`

Указывает объем памяти дедупликации для задачи в байтах (268435456 байт по умолчанию).

Аргумент

неотрицательное целое число

`[-r]`

Разрешает выполнять удаленную репликацию для клиента

`[-R]`

Разрешает выполнять централизованное восстановление для ресурсов клиента

`[-g]`

Создает ключи электронно-цифровой подписи (ЭЦП) и дает возможность клиенту проверять целостность резервных копий с их помощью

`[-P parallelizm-num]`

Определяет количество потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое неотрицательное число не равное нулю

`[-v]`

Подробный вывод сервисных процессов в терминал и логи

Конфигурация медиасервера

`-c primary-server`

Указатель на основной сервер.

Аргументы

следует указать FQDN сервера (не рекомендуется использовать ip адреса)

`-H dbhost`

Указатель на сервер, где располагается база данных rubackup.

Аргумент

следует указать hostname сервера, где располагается/будет располагаться база данных rubackup (не рекомендуется использовать ip сервера). Если не указать этот параметр, будет использоваться значение по-умолчанию — `localhost`

[-N dbname]

Определяет имя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

[-U dbuser]

Определяет имя суперпользователя базы данных rubackup. Если не указать этот параметр, будет использоваться значение по-умолчанию — `rubackup`

-Y dbpasswd-rubackup

Пароль для пользователя rubackup (владельца БД rubackup). Необходим для создания пользователя rubackup.

Аргументы

`rubackup_user_passwd`

-i iface

Сетевой интерфейс клиента для связи с севером.

Аргументы

Укажите имя клиентского сетевого интерфейса (напр. `Eth0`)

[-l local-backup-directory]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Если при конфигурации не был указан флаг `-l` и аргумент к нему, то в конфигурационном файле `/opt/rubackup/etc/config.file`, в параметре `use-local-backup-directory` будет использоваться значение по-умолчанию — `/tmp`. В ином случае будет использоваться директория, которая была указана при конфигурации.

Аргумент

укажите абсолютный путь к директории (напр. `RuBackup-tmp`)

[-S secondary-server]

Указатель на расположение дополнительного сервера резервного копирования.

Аргументы

укажите FQDN дополнительного сервера (не рекомендуется использовать ip адрес)

[-p parallel-tasks]

Возможность ограничить параллельное выполнение заданий. Если не указать этот параметр, будет использоваться значение по-умолчанию — `2`.

Аргумент

целое число ≤ 2

`[-D dedup-memory-size]`

Указывает объем памяти дедупликации для задачи в байтах (268435456 байт по умолчанию).

Аргумент

неотрицательное целое число

`[-r]`

Разрешает выполнять удаленную репликацию для клиента

`[-R]`

Разрешает выполнять централизованное восстановление для ресурсов клиента

`[-g]`

Создает ключи электронно-цифровой подписи (ЭЦП) и дает возможность клиенту проверять целостность резервных копий с их помощью

`[-P parallelizm-num]`

Определяет количество потоков для сетевых подпрограмм. Тип: опционально (устанавливает ограничение в 8 потоков, если опция не выбрана).

Аргументы

целое неотрицательное число не равное нулю

`[-v]`

Подробный вывод сервисных процессов в терминал и логи

`[-m]`

Опция включает мониторинг клиента

rb_interoperation

rb_interoperation — утилита администратора RuBackup для управления задачами импорта или экспорта резервных копий между независимыми системами резервного копирования. Вы можете управлять списком систем, для которых существует возможность импорта или экспорта. Добавлять, просматривать, редактировать, удалять, останавливать и запускать правила экспорта или импорта. Также вы сможете проверять очередь задач и удалять выполненные задачи или завершившиеся с ошибкой. У вас будет возможность создать задачу на экспорт резервной копии из репозитория.

-h

Справочное сообщение

-S

Вывод списка систем, для которых разрешены операции импорта или экспорта

-a fqdn -t export|import -H destination_system_HWID [-d description]

Добавить устройство FQDN в список систем, для которых разрешены операции импорта или экспорта резервных копий

-r ID

Удалить устройство из списка систем, для которых разрешены операции импорта или экспорта

-E

Вывод списка правил экспорта

-e export_rule_name -D destination_system_ID [-g global_schedule_rule_ID|-s strategy_rule_ID] [-P delay_period] [-d description]

Добавить новое правило экспорта в список

-x ID

Удалить правило экспорта

-I

Вывод списка правил импорта

-i import_rule_name -D destination_system_ID -c client_id -p pool_ID, -P store_period [-d description] [-H destination_client_HWID] [-m module]

Добавить новое правило импорта в список

-X ID

Удалить правило импорта

-T

Вывод очереди задач экспорта/импорта

-o

Удалить устаревшие задачи из очереди экспорта/импорта

-w

Удалить задачи, завершившиеся ошибкой, из очереди экспорта/импорта

-M

Вывод списка резервных копий, импортированных из репозитория

-W ID -t export|import

Остановка правила экспорта или импорта

-R ID -t export|import

Запуск правила экспорта или импорта

-C resource id -D destination_system_ID

Создать задачу на экспорт резервной копии из репозитория

rb_inventory

rb_inventory утилита администратора RuBackup для внесения в базу данных RuBackup информации о резервных копиях, которые были сделаны вне текущей конфигурации RuBackup, например, в другой серверной группировке RuBackup. **rb_inventory** анализирует каталог с резервными копиями и вносит информацию о них в базу данных RuBackup в зависимости от содержимого информационных файлов резервных копий.

-h

Справочное сообщение

-i Каталог

Провести инвентаризацию резервных копий, находящихся в указанном Каталоге

[-v]

Вывести расширенную информацию о работе утилиты

[-f]

Игнорировать незначительные ошибки, которые может содержать информационный файл резервной копии, или игнорировать недостаточную информацию о резервной копии в информационном файле

-t Volume_tag_картриджа

Выполнить инвентаризацию резервных копий, находящихся на указанном ленточном носителе

rb_local_filesystems

rb_local_filesystems — утилита администратора RuBackup для управления хранилищами резервных копий типа Файловая система (*File system*). Хранилища такого типа должны быть ассоциированы с пулом того же типа.

Резервные копии, по мере создания, будут автоматически располагаться в файловых системах пула, который назначен для той или иной резервной копии. Если с пулом ассоциированы несколько файловых систем, то они могут заполняться последовательно или равномерно, это определяется глобальным параметром `data_spred_into_pool` (возможны значения `sequentially` или `simultaneously`). При помощи **rb_local_filesystems** вы можете просматривать общий список хранилищ резервных копий типа файловая система в RuBackup, добавлять файловые системы в пул, удалять их из пула или изменять их описание. С пулом файловых систем должна ассоциироваться хотя бы одна файловая система. Утилита **rb_local_filesystems** не создаёт путь и не проверяет его физическое наличие на медиасервере, которому принадлежит пул.

Системный администратор RuBackup должен проверить точное наименование пути хранения резервных копий при использовании утилиты **rb_local_filesystems**. При удалении файловой системы из списка пулов содержащиеся в них резервные копии останутся нетронутыми.

-h

Справочное сообщение

-l

Список хранилищ резервных копий типа файловая система

-a Путь -p ID_пула [-d 'Описание']

Добавить в конфигурацию RuBackup новое хранилище резервных копий типа файловая система

-a Путь

абсолютный путь в файловой системе медиасервера, которому принадлежит пул с идентификатором ID_пула

-d

можно задать описание хранилища

-r ID

Удалить хранилище с идентификатором ID из конфигурации RuBackup

-c ID -d 'Новое описание'

Изменить описание хранилища

rb_log_viewer

rb_log_viewer — утилита администратора RuBackup для просмотра журнальных сообщений RuBackup.

-h

Справочное сообщение

-l

Вывод системного журнала RuBackup. Системный журнал обычно располагается в файле `/opt/rubackup/log/RuBackup.log`, но его местоположение может быть переопределено в конфигурационном файле (значение параметра `logfile`). Максимальный размер системного журнала составляет 1 МБ. При достижении этого размера происходит переключение на новый журнальный файл, а старый переименовывается с добавлением даты и времени переключения к названию файла

-c

Журнал клиентов

-m

Журнал медиасерверов

-g

Журнал глобального расписания

-t

Журнал главной очереди задач

-r

Журнал репозитория

-x

Очистить все журналы RuBackup (кроме системного журнала)

rb_logger_settings

rb_logger_settings — утилита администратора RuBackup для управления параметрами логирования.

-h

Справочное сообщение

-L

Вывод глобальных настроек

-U client/server

Обновление глобальных настроек



Если клиент и сервер находятся на одном хосте, то применяются настройки сервера.

-l

Уровень логирования

Значения:

- Error – только сообщения об ошибках;
- Warning – сообщения об ошибках и критические предупреждения;
- Info – сообщения об ошибках, критические предупреждения и информационные сообщения;
- Debug – сообщения об ошибках, критические предупреждения, информационные сообщения и отладочная информация; Debug1, Debug2, Debug3, Debug4 – отладочная информация. От основной информации (Debug1) до расширенной формы (Debug4).

-s

Максимальный размер файла журнала

-r

Максимальное количество журналов одного типа

-p client/server -i ID

Вывод персональных настроек

-i

ID клиента/сервера

-a client/server -i ID -l -s -r

Добавление персональных настроек. Необходимо задать все из указанных параметров



Если клиент и сервер находятся на одном хосте, то применяются настройки сервера.

-d client/server -i ID

Удаление персональных настроек. После удаления персональных настроек к серверу/клиенту применяются общие настройки

-u client/server -i ID

Обновление персональных настроек

-D client/server

Удаление всех персональных настроек. После удаления персональных настроек к серверу/клиенту применяются общие настройки

-v

Вывод расширенной информации

rb_media_servers

rb_media_servers — утилита администратора RuBackup для управления медиасerverами RuBackup. Вы можете просматривать список медиасерверов, добавлять их, удалять или изменять их описания. медиасервер предназначен для взаимодействия с клиентами при создании, восстановлении и передаче резервных копий. В простейшем случае основной сервер RuBackup также является медиасервером. В случае включения в серверную группировку резервного сервера RuBackup, он также является медиасервером. В серверной группировке RuBackup вспомогательные медиасерверы могут быть использованы для горизонтального масштабирования системы резервного копирования и равномерного распределения нагрузки.

Удалить медиасервер из конфигурации RuBackup возможно только если заранее удалены все пулы устройств хранения резервных копий, ассоциированные с ним. При первоначальной настройке системы резервного копирования нет необходимости специально добавлять медиасерверы в серверную группировку с помощью параметра **-a**. Достаточно запустить медиасервер с правильными адресами основного и резервного (при наличии) серверов в конфигурационном файле.

При первом запуске медиасервер будет зарегистрирован в системе как неавторизованный медиасервер, при этом процесс медиасервера будет остановлен. Необходимо авторизовать новый медиасервер с помощью вызова **rb_media_servers -t** и вновь запустить его процесс. медиасервер должен иметь хотя бы один пул устройств хранения резервных копий (см. [rb_pools](#)), в противном случае на хосте медиасервера будет выводиться соответствующее предупреждение, а медиасервер будет невозможно использовать для хранения резервных копий.

-h

Справочное сообщение.

-l [-o]

Список авторизованных медиасерверов

-o

выдаёт информацию только о серверах, которые работают в данный момент времени

-u [-v]

Список неавторизованных медиасерверов

-v

предоставляет расширенную информацию

-a Имя_хоста [-4 ipv4] [-6 ipv6] [-m macaddr] [-d 'Описание']

Добавить в конфигурацию RuBackup новый медиасервер

-v

Выводит список авторизованных медиа-серверов в расширенном формате

-r ID

Удалить авторизованный медиасервер с идентификатором ID из конфигурации RuBackup. Если с медиасервером ассоциированы пулы устройств хранения резервных копий, его невозможно удалить из конфигурации

-l

получение значения идентификатора

-x ID

Удалить неавторизованный медиасервер с идентификатором ID из списка неавторизованных серверов

-u

получение значения идентификатора

-t ID

Авторизовать неавторизованный медиасервер с идентификатором ID

-u

получение значения идентификатора

-c ID -d 'Новое описание'

Изменить описание медиасервера. При необходимости изменить другие параметры следует удалить медиасервер и создать новый медиасервер с правильными параметрами

rb_modules

rb_modules — утилита администратора RuBackup для управления модулями RuBackup. При вызове без аргументов утилита выводит список установленных модулей.

-h

Справочное сообщение

-l

Список установленных модулей RuBackup

-i module.sql

Инициализировать новый модуль

rb_notifications

rb_notifications — утилита администратора RuBackup для управления очередью уведомлений RuBackup. В очереди уведомлений содержатся все актуальные уведомления групп пользователей RuBackup о происходящих в системе событиях. Уведомления могут быть настроены в правилах глобального расписания и в стратегиях.

Уведомления могут быть настроены для следующих событий:

- успешное выполнение резервной копии;
- резервная копия завершилась с ошибкой;
- уведомление о проверке резервной копии;
- завершение действия правила или стратегии;
- заканчивается свободное место в пуле правила или стратегии.

В соответствии с глобальными настройками RuBackup очередь может автоматически очищаться от завершённых, ошибочных и прерванных задач. В очереди задачи могут находиться со следующими статусами:

- New  новое уведомление
- Sent  уведомление отправлено
- Error  уведомление не выполнено по каким-то причинам (см. журнальный файл)

-h

Справочное сообщение

-v

Список актуальных задач в очереди. Параметр предоставляет расширенную информацию. Наиболее полная информация о состоянии и свойствах задачи может быть получена в оконном менеджере администратора системы резервного копирования RBM

-o

Удалить из очереди все уведомления со статусом Sent

-e

Удалить из очереди все уведомления со статусом Error

-k ID

Перевести уведомление с идентификатором ID в статус Error



Используйте *rb_notifications -l* чтобы получить значение идентификатора

rb_pools

rb_pools — утилита администратора RuBackup для управления пулами RuBackup. Вы можете просматривать список пулов, добавлять, удалять и изменять их. Каждый пул принадлежит какому-либо медиасерверу. Пулы RuBackup используются для группирования устройств хранения резервных копий.

Пулы бывают пяти типов:

- файловая система (File system). В такой пул могут быть добавлены каталоги файловой системы медиасервера;
- ленточная библиотека (Tape library). В такой пул могут быть добавлены картриджи ленточной библиотеки;
- облако (Cloud). В такой пул могут быть добавлены облачные хранилища.
- пул определенный клиентом (Client defined). Такой пул определяет сам клиент.
- Блочное устройство (block device). Такой пул предоставляет доступ к ресурсам в виде блоков фиксированного размера. Для такого типа пула невозможно определить параметры `-z compression_type` и `compression location`.

Чтобы включить устройство хранения в пул необходимо воспользоваться соответствующей командой: `rb_local_filesystems`, `rb_tape_cartridges`, `rb_clouds`, `rb_client_defined_storages` или `rb_block_devices`.

-h

Справочное сообщение

-v

Выводит список пулов. Предоставляет расширенную информацию о пулах

-a Имя пула **-t** 'Block device' [**-m** Имя хоста] [**-z** Тип_сжатия] [**-R** 'Период хранения'] [**-b** Размер блочного устройства] [**-A** Хэш-алгоритм] [**-d** 'Описание'] [**-2|5**]

-a

имя пула

-t Block device

определяет тип пула, возможные варианты: `File system`, `Tape library`, `Cloud` и `Client defined`

-m

определяет имя хоста медиасервера

-z

можно определить будут ли резервные копии в этом пуле подвергаться сжа-

тию (возможные значения параметра: `None, fast, optimal, best`)

-R

устанавливает период хранения данных в днях. Это обозначает, что задается период хранения резервных копий с момента создания в данном пуле. В некоторых случаях эта настройка может быть проигнорирована

Таблица 5. Значения сроков хранения могут быть записаны в формате периода хранения, например: 3 дня, 1 месяц, 2 недели, 1 год или `pull` для удаления значения.

Значение	Пояснение
Y	Годы
M	Месяцы
W	Недели
D	Дни

-b

размер блока, который определяет объем информации, хранящейся в одном блоке, Варианты размеров блока указаны в байтах и включают следующие значения: `8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576`. (Более подробно см. «Руководство системного администратора RuBackup»)

-A

указывает хэш-алгоритм. Для блочных устройств (`block_devices`) алгоритмы хэширования могут быть такими: `sha1, sha2, skein, blake2b`

-d

добавляет описание пула

-2|5

заданная длина кэша (256 и 512, соответственно)

-n

новое имя пула

-r ID

Удалить пул с идентификатором ID из конфигурации RuBackup. Для получения значения идентификатора используйте вызов `rb_pools -l`. В том случае, если в пул включены устройства хранения, его невозможно удалить из конфигурации

-c ID [-n Имя_пула] [-z Тип_сжатия] [-s Сторона_сжатия] [-d 'Новое описание']

Изменить параметры пула

rb_remote_replication

rb_remote_replication — утилита администратора RuBackup для управления непрерывной удалённой репликацией в RuBackup. Непрерывная удалённая репликация состоит из отдельных правил, которые могут выполняться по определённым условиям для определённого ресурса. При помощи **rb_remote_replication** можно просматривать список правил непрерывной удалённой репликации, экспортировать настройки правила в файл и импортировать правило из файла, удалять правила, останавливать функционирование правила или запускать его в работу.

Система резервного копирования инициирует новые задачи на основании временных параметров запуска правил непрерывной удалённой репликации. В качестве параметров старта правила вы можете определить минуту, час, день месяца, месяц и день недели.

Правило может быть приостановлено (статус `wait`) и запущено в работу (статус `run`). Приостановленное правило не будет инициировать создание задач, но, при необходимости, можно выполнить срочную задачу, используя параметр `-x`.

-h

Справочное сообщение

-m

Список модулей, поддерживающих непрерывную удалённую репликацию

-l

Список правил непрерывной удалённой репликации и их базовые параметры

-s

Список правил непрерывной удалённой репликации и их временные параметры

-n

Список правил непрерывной удалённой репликации и их параметры уведомления

-a

Список правил непрерывной удалённой репликации и их расширенные параметры

-X ID

Список дополнительных параметров для правила непрерывной удалённой репликации

-i Файл_импорта

Импорт правила из заранее подготовленного файла

-e ID [-f Файл_экспорта]

Экспорт правила с идентификатором ID в вывод stdout или в файл

-d ID

Удалить правило непрерывной удалённой репликации с идентификатором ID

-w ID

Приостановить исполнение правила с идентификатором ID

-r ID

Возобновить исполнение правила с идентификатором ID

-x ID

Немедленно выполнить правило непрерывной удалённой репликации с идентификатором ID

-C ID -E Параметр:Значение,Параметр1:Значение1, ...

Изменить дополнительные параметры для правила с идентификатором ID. При этом возможно изменить сразу несколько параметров в виде Параметр:Значение, перечислив их через запятую

rb_repository

rb_repository — утилита администратора RuBackup для доступа к записям репозитория RuBackup. Позволяет просматривать список резервных копий, удалять и перемещать резервные копии, проверять их целостность и выполнять их репликацию (копирование) в другие пулы. Для выполнения этих действий утилита **rb_repository** создаёт соответствующую задачу в главной очереди задач и заканчивает своё выполнение до того момента, как задача будет выполнена. Чтобы проверить исполнение поставленной задачи, необходимо использовать утилиту **rb_task_queue**.

Перемещение резервных копий невозможно из пула типа ленточная библиотека, поскольку картриджи ленточной библиотеки могут располагаться вне ленточной библиотеки.

Проверка резервных копий может осуществляться с проверкой электронной подписи, если соответствующие настройки были произведены перед выполнением резервной копии на клиенте (см. руководство «Электронные подписи резервных копий»).

-h

Справочное сообщение

-l [-v]

Выводит список резервных копий в репозитории RuBackup. Предоставляет расширенную информацию о резервных копиях. Чтобы получить полную информацию обо всех свойствах резервной копии рекомендуется использовать оконный менеджер администратора системы резервного копирования RBM

-v

Выводит детализированный список резервных копий в репозитории RuBackup

-c ID -p pool_ID

Реплицировать (копировать) резервную копию в другой пул

-m ID -p pool_ID

Переместить резервную копию в другой пул

-r ID

Удалить резервную копию из репозитория. При выполнении этой команды резервная копия будет физически удалена из устройств хранения резервных копий RuBackup

-V ID

Проверить резервную копию. Результат проверки будет отражён как статус

резервной копии

-S ID [-D YYYY-MM-DD] [-T HH:MM] [-C]

Установить время и дату хранения резервной копии

-C

Очистить время и дату хранения резервной копии

-x ID -H HWID -d restore_path [-y]

Восстановить резервную копию на клиенте с развертыванием

-y

Установить согласие со всеми вопросами

-X ID -H HWID -d restore_path [-y]

Восстановить резервную копию на клиенте без развертывания

-y

Установить согласие со всеми вопросами

-U time_interval [-R]

Задать срок хранения РК срочных бэкапов от времени создания РК, если это поле еще не задано

time_interval

имеет формат PostgreSQL interval

-R и -U

означает, что срок хранения срочных РК нужно обновить для всех РК а не только для тех у которых это поле не задано

rb_secret_storage

rb_secret_storage — утилита администратора RuBackup для управления доступом к аутентификационной информации СУБД PostgreSQL в хранилище секретов HashiCorp Vault.

-h

Справочное сообщение

-s

Вывод списка всех созданных в RuBackup хранилищ секретов

-t

Вывод списка типов хранилищ секретов

-v name -T storage_type [-d description] [-e cacert]

Добавление нового хранилища секретов

-v name

название хранилища секретов

-T storage_type

тип хранилища секретов

-d description

описание хранилища

-e cacert

полный путь к SSL/TLS сертификату для доступа к хранилищу

-r ID

Удаление хранилища секретов, созданного в СРК RuBackup, с идентификатором ID

-m -u username -p password

Просмотр списка методов получения секретов из хранилища секретов, доступных пользователю СРК RuBackup

-u username

логин Пользователя или Администратора СРК

-p password

пароль Пользователя или Администратора СРК

```
-c name -S secret_storage_id -k token -M method -u username -p password [-d description]
```

Добавление метода получения секретов из хранилища секретов в СРК RuBackup

-c name

название метода

-S secret_storage_id

идентификатор хранилища

-k token

строка, при помощи которой происходит авторизация в хранилище секретов при отправки запросов

-M method

метод доступа к хранилищу

-u username

логин Администратора СРК

-p password

пароль Администратора СРК

-d description

описание хранилища, необязательный параметр

```
-R ID -u username -p password
```

Удаление метода получения секретов СРК RuBackup из хранилища секретов

-R ID

идентификатор метода

-u username

логин пользователя СРК

-p password

пароль пользователя СРК

```
-n ID -k token -u username -p password
```

Получение нового токена для хранилища по его ID

-n ID

идентификатор хранилища секретов

-k token

строка, при помощи которой происходит авторизация в хранилище секретов при отправке запросов

-u username

логин администратора СРК

-p password

пароль администратора СРК

rb_security

rb_security — утилита для работы с журналом событий информационной безопасности.

При необходимости, можно выполнить:

- включение/выключение режима сбора данных. По умолчанию осуществляется включение журналирования всех значимых таблиц (кроме очередей задач, временных таблиц и таблиц с хэш)
- журналирование очередей задач (опция `--enable-for-tasks`)
- отображение событий
- получение информации о состоянии функционала (вкл./выкл.)
- экспорт событий в форматы txt, json, cef
- очистку журнала

-h

Справочное сообщение

-s [--connection-string]

Использует строку подключения без пароля для работы с базой данных вместо информации о конфигурационном файле* (см. <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNSTRING>)

-s

является служебным и необходим для внутренней работы сервера RuBackup* Пользователям не рекомендуется использовать данный параметр

--check-enabled

Возвращает строку "true", если аудит был включен, в противном случае "false"

--check-tasks-enabled

Возвращает строку "true", если аудит для задач был включен, в противном случае строку "false"

-e [--enable]

Включение режима сбора данных в журнал событий ИБ* По умолчанию осуществляется включение журналирования всех значимых таблиц, кроме очередей задач и временных таблиц* При необходимости можно включить журналирование очередей задач (см. опцию `-E [ --enable-for-tasks ]`)

-E [--enable-for-tasks]

Дополнительно включить аудит безопасности для всех задач в очередях (используется совместно с опцией `-e [ --enable ]`)

-d [--disable]

Отключить аудит безопасности

-x [--export] arg

Экспорт журнала событий ИБ в файл требуемого формата, возможные значения: txt, json, cef

-f [--export-filename]

если параметр не задан, то экспорт выполняется в каталог по умолчанию `/opt/rubackup/log/security_log/` с форматом имени файла вида `rubackup_audit.export.cef.<ts>`, где `<ts>` — локальное время системы в формате `%Y-%m-%d_%H-%M-%S``.

-f [--export-filename] arg

Определение полного пути экспортируемого файла

Пример экспорта журнала событий ИБ:

```
sudo      rb_security      --export      cef      --export-filename
/opt/rubackup/log/security_log/info_sec_log.cef
```

Журнал событий ИБ будет экспортирован в файл формата `.cef` в указанный каталог и будет содержать все события информационной безопасности, начиная с момента создания БД (или её последней очистки) и до момента отсоединения от неё.

Формат записи в журнале событий ИБ в экспортируемом файле `.CEF`:

- версия CEF
- наименование поставщика продукта
- наименование продукта
- версия продукта
- тип события
- user — имя пользователя, совершившего действие
- made a change — тип запроса к базе данных
- on schema — используемая схема базы данных
- table — наименование таблицы, в которой произошли изменения
- дата и время совершения события

- степень важности события
- `inet_server_addr` — IP-адрес сервера RuBackup
- `inet_client_addr` — IP-адрес узла, с которого было инициировано событие
- `original_data` — данные до изменения
- `new_data` — данные после изменения
- `query` — описание события



некоторые атрибуты в последнем столбце могут отсутствовать в зависимости от сценария изменений

-c [--clean]

Очистить записи таблицы базы данных (предварительно экспортировав ее в формате "txt")*

В результате очистки журнала событий ИБ:

- произведен экспорт всех записей о событиях информационной безопасности в файл формата .txt по указанному пути
- удалены все записи о событиях в журнале событий ИБ
- в очищенный журнал событий ИБ добавлена запись об очистке журнала ИБ

-l [--list]

Вывод в консоль всех записей журнала событий ИБ с момента создания БД или с момента последней очистки журнала в формате:

- ID – номер события в журнале
- User name – имя пользователя, инициировавшего событие
- Action name – тип события
- Timestamp – дата и время по UTC произошедшего события
- Query – описание события
- Original data (при указании опции `-v [--verbose]`) – данные до изменений, если таковые имеются, например, при запросе обновления данных (UPDATE).

-v [--verbose]

Расширенный режим вывода

rb_strategies

rb_strategies — утилита администратора RuBackup для управления стратегиями RuBackup.

Стратегия резервного копирования может быть включена (статус `run`) и выключена (статус `wait`). Если стратегия работает, она будет создавать задачи резервного копирования в соответствии с расписаниями для всех ресурсов и клиентов, которые ее касаются.

В стратегии резервного копирования задачи создания разностных копий не могут стартовать в одно и то же время с созданием полной копии. Даже если такое задано расписанием, сервер резервного копирования будет игнорировать создание задач разностного копирования, если в то же время была создана задача или группа задач на выполнение полного резервного копирования ресурсов и клиентов, задействованных в стратегии.

-h

Справочное сообщение

-v

Параметр `-v` предоставляет информацию о стратегиях в расширенном формате

-l

Выводит список стратегий с их идентификаторами (ID)

-F

Выводит список полных временных параметров резервного копирования

-I

Выводит список временных параметров инкрементального резервного копирования

-D

Выводит список временных параметров дифференциального резервного копирования.

-N

Выводит список параметров уведомления

-A

Выводит список дополнительных опций

-L strategy_ID

Выводит список всех правил стратегий

-X rule_ID

Выводит список дополнительных параметров правил стратегий

-C rule_ID -E

Изменяет дополнительные параметры правил стратегий

-W ID

Изменяет статус на ожидание

-R ID

Изменяет статус на запуск

-i import_file

Импортирует стратегию из файла

-e ID -f export_file

Экспортирует стратегию в файл

-d ID

Удаляет стратегию

-a ID -c client_hostname -t resource_type -r resource [-n normal_execution_script] [-x exception_execution_script] [-p priority]

Добавляет правило в стратегию

-z ID

Удаляет правило из стратегии

rb_tape_cartridges

rb_tape_cartridges — утилита администратора RuBackup для управления картриджами ленточных библиотек в конфигурации RuBackup. Вы можете просматривать список картриджей, добавлять, удалять или изменять их. Каждый картридж принадлежит какому-либо пулу типа ленточная библиотека. Нет необходимости использовать параметр `-a` для добавления нового картриджа в коллекцию картриджей RuBackup; хорошим приёмом будет воспользоваться операцией импорта картриджей (см. [rb_tape_libraries](#)).

Картридж может быть загружен в ленточную библиотеку или находится вне неё. Удалить картридж из коллекции RuBackup можно только в том случае, если он выгружен из ленточной библиотеки при помощи операции экспорта в **RBM** или **rb_tape_libraries**. При удалении картриджа из коллекции резервные копии на нём остаются в неприкосновенности.

Для хранения резервных копий на картридже ленточной библиотеки на нём должна быть создана файловая система LTFS. При необходимости получить доступ к резервным копиям на картридже вне системы резервного копирования, это можно сделать просто примонтировав файловую систему LTFS к нужной точке монтирования хоста. При необходимости удалить данные с картриджа рекомендуется использовать команду `mt erase` (см. `mt`). Изменять описание или пул картриджа допускается только в случае, когда RuBackup находится в сервисном режиме (см. [rb_global_config](#) параметр `service_mode`, значение `yes` для сервисного режима, `no` для обычного режима).

Если необходимо добавить в коллекцию картридж с расположенными на нём резервными копиями, которые сделаны в какой-либо другой конфигурации RuBackup, и присутствующие на картридже резервные копии не учтены в репозитории RuBackup, используйте операцию импорта картриджей (см. [rb_tape_libraries](#)).

-h

Справочное сообщение

-l

Список картриджей в коллекции RuBackup

-a volume_tag -t Тип -p ID_пула [-d 'Описание']

Добавить в коллекцию картриджей RuBackup новый картридж. Использовать этот способ добавления картриджей целесообразно в случае необходимости при помощи скрипта добавить в коллекцию большое количество картриджей, у которых совершенно точно известны их `volume_tag`. В других случаях рекомендуется использовать операцию импорта картриджа (см. [rb_tape_libraries](#)).

-r ID

Удалить картридж из коллекции

-c ID [-p ID_пула] [-d 'Новое описание']

Изменить пул, к которому принадлежит картридж, или описание картриджа

-F volume_tag

Заморозить картридж. Картридж становится доступным для чтения и недоступным для записи. Срок хранения резервной копии не истекает, пока администратор не разморозит картридж вручную. Если у картриджа стоит данный статус, то с него не будут удаляться просроченные резервные копии.

-f volume_tag

Разморозить картридж.

-Z volume_tag

Приостановить работу картриджа. Картридж становится доступным для чтения и недоступным для записи. РК могут удаляться, срок хранения РК продолжает отсчитываться.

-z volume_tag

Возобновить работу картриджа после приостановки.

rb_tape_libraries

rb_tape_libraries — утилита администратора RuBackup для управления ленточными библиотеками в системе резервного копирования RuBackup. С помощью **rb_tape_libraries** вы можете просматривать информацию о ленточных библиотеках в серверной группировке RuBackup, синхронизировать ленточную библиотеку с информацией о ней в базе данных RuBackup, импортировать, экспортировать и перемещать картриджи в ленточной библиотеке, а также производить LTFS форматирование картриджей, находящихся в слотах ленточной библиотеки.

Для внесения в конфигурацию RuBackup новой библиотеки, изменения конфигурации библиотеки или удаления библиотеки из конфигурации необходимо воспользоваться оконным менеджером администратора системы резервного копирования RBM.

Перед использованием **rb_tape_libraries** рекомендуется перевести RuBackup в сервисный режим (см. [rb_global_config](#), параметр *service_mode*, значение `yes`) и дождаться окончания всех задач, находящихся в главной очереди задач (см. [rb_task_queue](#)). После окончания использования утилиты необходимо перевести RuBackup в обычный режим (см. [rb_global_config](#), параметр *service_mode*, значение `no`).

-h

Справочное сообщение

-l

Список ленточных библиотек в серверной группировке RuBackup

-L ID_библиотеки -s

Список слотов в указанной ленточной библиотеке

-L ID_библиотеки -m

Список роботов в указанной ленточной библиотеке

-L ID_библиотеки -d

Список всех ленточных приводов в указанной ленточной библиотеке

-L ID_библиотеки -t

Список всех картриджей, загруженных в ленточные приводы, в указанной ленточной библиотеке

-M ID_робота -e MAIL_SLOT_ID

Выгрузить (экспортировать) картридж из mail-слота ленточной библиотеки. После этой операции картридж останется в коллекции картриджей RuBackup, но mail-слот будет обозначен как пустой. См. документацию на ленточную биб-

лиотеку как действовать в ситуации экспорта картриджа для его физического извлечения из слота

-M ID_робота -S

Провести процедуру синхронизации ленточной библиотеки и информации в базе данных RuBackup. Эта операция может быть необходима в случае, если слоты библиотеки были загружены без использования операции импорта или картриджи были извлечены из библиотеки без использования операции экспорта

-S

(синхронизация) проверяет только расположение картриджей в ленточной библиотеке

-c

используйте параметр чтобы проверить наличие файловой системы LTFS на картридже

-M ID_робота -i MAIL_SLOT_ID -p ID_пула

Загрузить (импортировать) картридж в библиотеку, используя mail-слот. Перед выполнением этой операции необходимо разместить картридж в принимающем слоте ленточной библиотеки (см. документацию на ленточную библиотеку)

-n volume_tag

Провести LTFS форматирование картриджа. Без этой операции невозможно использование картриджа в RuBackup. Картридж должен находиться в слоте ленточной библиотеки перед началом операции. Если картридж находится в ленточном приводе, его необходимо выгрузить в слот ленточной библиотеки перед началом форматирования.

-c volume_tag

Проверить наличие файловой системы LTFS на картридже ленточной библиотеки

rb_task_queue

rb_task_queue — утилита администратора RuBackup для управления главной очередью задач RuBackup. В очереди задач содержатся все актуальные задачи на создание, восстановление, удаление, перемещение и проверку резервных копий. В соответствии с глобальными настройками RuBackup очередь может автоматически очищаться от завершённых, ошибочных и прерванных задач.

Задачи системы резервного копирования могут быть следующих типов:

- Backup global  задача резервного копирования, инициированная стратегией или глобальным расписанием RuBackup.
- Backup local  задача резервного копирования, инициированная локальным расписанием клиента (если это разрешено клиенту).
- Restore  задача восстановления резервной копии.
- Verify  задача проверки резервной копии. Проверка может выполняться с использованием цифровой подписи, если клиент настроен соответствующим образом (см. руководство "Электронная подпись резервных копий").
- Delete  задача удаления резервной копии из репозитория и с устройств хранения резервных копий.
- Move  задача перемещения резервной копии из одного пула устройств хранения резервных копий в другой пул. Задача перемещения не может быть выполнена для резервных копий, которые находятся на картриджах ленточных библиотек.

В очереди задачи могут находиться со следующими статусами:

- New  новая задача.
- Assigned - задача назначена определённому медиасерверу.
- At_Client  задача передана клиенту.
- Execution  происходит исполнение задачи, например, создаётся резервная копия.
- Done  задача успешно выполнена.
- Broken  задача была прервана (при создании резервной копии, вероятно, вызвали ошибку пользовательские скрипты, запускающиеся перед или после создания резервной копии. Точную информацию можно найти в журнальном файле клиента).
- Suspended  задача приостановлена из-за недоступности устройства хранения резервных копий.
- Error  задача завершилась с ошибкой.

- Restarted  задача перезапущена после приостановки.
- Transmission  передача резервной копии.
- Start_Transfer  система резервного копирования готова к передаче резервной копии от клиента медиасерверу или обратно.

-h

Справочное сообщение

-v

Список актуальных задач в очереди. Параметр -v предоставляет расширенную информацию. Наиболее полная информация о состоянии и свойствах задачи может быть получена с помощью оконного менеджера администратора системы резервного копирования RBM

-o

Удалить из очереди все задачи со статусом Done

-e

Удалить из очереди все задачи со статусом Error

-k ID

Перевести задачу с идентификатором ID в статус Error.



Используйте `rb_task_queue -l` чтобы получить значение идентификатора

rb_tl_task_queue

rb_tl_task_queue — утилита администратора RuBackup для управления очередью ленточных библиотек RuBackup. Нормальное положение картриджа в ленточной библиотеке при отсутствии необходимости записать на него резервную копию в данный момент времени - выгружен в свой слот ленточной библиотеки. При возникновении необходимости записать или прочитать резервную копию на картридже ленточной библиотеки он будет автоматически загружен в первый свободный магнитофон и выгружен в свой слот по окончании операции. Не рекомендуется использовать параметр **-k** в каком-либо случае, кроме явной блокировки очереди какой-либо задачей. При возникновении подобной ситуации необходимо обратиться в службу технической поддержки RuBackup.

В соответствии с глобальными настройками RuBackup очередь ленточных библиотек автоматически очищается от завершённых задач. В очереди могут быть задачи следующих типов:

- Load  загрузить картридж в ленточный привод;
- Unload  выгрузить картридж из ленточного привода.

Возможные статусы задач:

- New  только что поставленная задача;
- Wait  ожидает возможности переместить картридж по назначению (должен быть свободен ленточный привод или слот);
- Execution  происходит перемещение картриджа;
- Ready  перемещение произошло, и задача из очереди задач может быть перезапущена;
- Done  задача из очереди задач завершена и может быть удалена из очереди за ненадобностью;
- Error  перемещение прошло неудачно;
- Out of library  картридж находится вне библиотеки.

-h

Справочное сообщение

-l

Список актуальных задач в очереди с их идентификаторами (ID)

-k ID

Убить задачу. Перевести задачу с идентификатором ID в статус Error.



Возможность убить задачу предусмотрена на крайний случай, когда

задача явно зависла. Это, по сути, нештатная ситуация, т.к. ленточная библиотека должна быть приведена в состояние, когда после выполнения задач её приводы пустые, а картриджи находятся в своих слотах. После применения подобной команды возможно потребуется выполнение выгрузки картриджа в ручном режиме при помощи `mtx` и последующая синхронизация библиотеки в RBM. Не рекомендуется использовать параметр `-k` в каком-либо случае, кроме явной блокировки очереди какой-либо задачей. При возникновении подобной ситуации необходимо обратиться в службу технической поддержки RuBackup.

rb_update

rb_update — утилита администратора RuBackup для управления обновлениями баз данных RuBackup. Создает sql инструкции, позволяющие сделать update базы данных RuBackup.

-h

Справочное сообщение

-v

Включение информационного режима и предоставление дополнительной информации о выполняемых операциях

-R

Принудительное обновление существующей базы данных

-O

Указать путь к выходному файлу

-H

Указать адрес хоста базы данных

-P

Указать порт соединения базы данных

-D

Указать имя существующей базы данных

--ssl-mode mode

Выбрать ssl-mode для определения степени защиты соединения

--ssl-root-cert path

Указать путь к корневому сертификату

--ssl-cert path

Указать путь к сертификату клиента

--ssl-key path

Указать путь к приватному ключу клиента

-U

Указать имя пользователя базы данных

-W

Принудительная установка пароля

-I

Указать путь к каталогу с новыми sql-скриптами

rb_user_groups

rb_user_groups — утилита администратора RuBackup для управления группами пользователей RuBackup. Вы можете просматривать группы пользователей, добавлять и удалять их, а также изменять их название и описание.

-h

Справочное сообщение

-l

Список пользовательских групп

-a Имя_группы [-d 'Описание']

Добавить в конфигурацию RuBackup группу с именем Имя_группы и с заданным описанием. При необходимости добавить значения с пробелами используйте одинарные кавычки, например, 'Новая группа'

-r ID

Удалить из конфигурации группу с идентификатором ID

-s ID [-n Новое_имя_группы] [-d 'Новое описание']

Изменить название группы с идентификатором ID и, при необходимости, изменить описание группы



Используйте *rb_user_groups -l* чтобы получить значение идентификатора

rb_users

rb_users — утилита администратора RuBackup для управления пользователями RuBackup. Вы можете просматривать список пользователей, добавлять, удалять и изменять их. Пользователи RuBackup используются в работе системы уведомления о событиях RuBackup. Каждый пользователь должен иметь работающий адрес электронной почты. В списке пользователей не допускаются пользователи с одинаковым именем или одинаковыми адресами электронной почты.

-h

Справочное сообщение

-l [-v]

Список пользователей.

-v

предоставляет расширенную информацию о пользователях

-A username -e e-mail [-g ID_группы] [-p 'пароль'] [-f 'Полное имя'] [-o 'адрес':'тел':'офис'] (add new user with password)

-A username -e e-mail [-g ID_группы] -W [-f 'Полное имя'] [-o 'адрес':'тел':'офис'] (add new user and enter the password securely)

Добавить в конфигурацию RuBackup нового пользователя с указанным адресом электронной почты

-g

возможно определить, в какую пользовательскую группу будет добавлен пользователь

-f

можно указать полное имя пользователя

-p

можно указать адрес, телефон и месторасположение офиса пользователя. Если значения содержат пробелы, то их необходимо заключить в одинарные кавычки, например, 'Тверская ул. д.12' : '+70001002020' : 'офис 123'

-W

дает возможность ввести пароль пользователя после его создания

-R ID

Удалить из конфигурации пользователя с идентификатором ID



Используйте `rb_users -l` для получения значения идентификатора блоч-

ного устройства

```
-C ID [-n username] [-e e-mail] [-g ID_группы] [-f 'Полное имя'] [-o 'адрес': 'тел': 'офис'] -p password -W
```

Изменить параметры пользователя с идентификатором ID

```
-D ID
```

Лишает пользователя привилегий

```
-G ID -s (Grant supervisor privilege to the user. User id is required)
```

```
-G ID -S ID_медиа сервера (Grant maintainer privilege to the user. User id and media server id are required)
```

```
-G ID -a ID_группы_клиентов (Grant administrator privilege to the user. User id and client group id are required)
```

Добавление привилегий для существующего пользователя

```
-s
```

наделяет пользователя полномочиями супервизора

```
-S
```

наделяет пользователя полномочиями сопровождающего

```
-a
```

наделяет пользователя полномочиями администратора

rbfd

rbfd — утилита администратора RuBackup для создания и восстановления полных и инкрементальных резервных копий ресурсов в любых файловых системах. Ресурсом может быть файл, каталог или блочное устройство.

Утилита поддерживает три режима создания и восстановления резервных копий.

В режиме *Single archive* утилита создаёт файл-архив и файл с мета-данными снимка состояния.

В режиме *Data flow* утилита создаёт файл с мета-данными и переносит дедуплицированные блоки с клиента на сервер RuBackup, помещая их в дедуплицированное хранилище. Этот режим работы является служебным. Его явный вызов не поддерживается.

При помощи утилиты **rbfd** можно восстановить из копии или цепочки копий отдельные файлы или всю папку целиком.

В качестве ресурса для создания резервной копии вы можете использовать текстовый файл, содержащий список файлов. В этом случае файл должен начинаться со строки **RBFD file list**, например:

```
RBFD file list:
/home/user/dir/exclude
/home/user/dir/exclude/file-1.txt
/dev/sdb :sdb
```

При восстановлении блочного устройства путь указывается отличный от исходного (то, что после двоеточия в RBFD file list).

В утилите **rbfd** реализованы алгоритмы хеш-функций:

```
blake2b, sha, skein, streebog и (GOST_R_34_11_2012)
```

Подробную информацию о реализованных алгоритмах см. [Алгоритмы хеш-функций](#)

-l

используйте параметр **-l** чтобы получить список файлов резервной копии не только для **.meta** файла, но и для архива **.rbfd**.

-m

используйте параметр **-m** в том случае, если в режиме **single_file_mode** не требу-

ется создания мета файла.

```
-u archive_file --add resource0,resource1 [ -m meta_file ]
```

используйте параметр для добавления файл в уже созданный архив

Есть возможность подавать на вход несколько ресурсов списком через запятую (без пробелов), например:

```
rbfd -c file0,file1,dir0,dir1 -n 1.rbfd
```

Утилита **rbfd** поддерживает алгоритмы защитного преобразования.

Возможны следующие варианты:

```
anubis, aria, cast6, camellia, kalyna, kuznyechik, mars, rijndael (или AES),  
serpent, simon, sm4, speck, threefish, twofish
```

Подробную информацию о реализованных алгоритмах см. [Алгоритмы защитного преобразования](#).

Мастер-ключ

В ходе инсталляции будет создан мастер-ключ для защитного преобразования резервных копий и ключи для электронной подписи, если электронную подпись предполагается использовать.



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



После создания ключи рекомендуется скопировать на внешний носитель, а так же 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:

```
brestadmt@srv:~$ hexdump /opt/rubackup/keys/master-key  
00000000 e973 053d 10a1 c0c1 40e8 d332 9463 a7ee  
00000100 8965 f275 d5e4 a04a d07d a625 d4e8 755f  
00000200
```

Алгоритмы защитного преобразования

В кластерной серверной группировке при восстановлении зашифрованной резервной копии на другом клиенте группировки необходимо использовать тот же мастер-ключ клиента, с помощью которого делалась данная копия.

Таблица 6. Алгоритмы защитного преобразования, доступные в утилите `rbfd`

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

Алгоритмы хеш-функций

Таблица 7. Алгоритмы хеш-функций, реализованные в утилите `rbfd`

Наименование алгоритма	Длина хэш, бит	Примечание
streebog или GOST_R_34_11_2012	256, 512	ГОСТ Р 34.11-2012 «Информационная технология Криптографическая защита информации. Функция хеширования» — действующий российский криптографический стандарт, определяющий алгоритм и процедуру вычисления хеш-функции. Разработан Центром защиты информации и специальной связи ФСБ России с участием ОАО «ИнфоТеКС» и введен в действие 1 января 2013 года.

Наименование алгоритма	Длина хэш, бит	Примечание
sha	256, 512	Хеш-функции SHA-2 разработаны Агентством национальной безопасности США и опубликованы Национальным институтом стандартов и технологий в федеральном стандарте обработки информации FIPS PUB 180-2 в августе 2002 года
skein	256, 512	Skein — алгоритм хеширования переменной разрядности, разработанный группой авторов во главе с Брюсом Шнайером. Хеш-функция Skein выполнена как универсальный криптографический примитив, на основе блочного шифра Threefish, работающего в режиме UBI-хеширования. Основные требования, предъявлявшиеся при разработке — оптимизация под минимальное использование памяти, криптографически безопасное хеширование небольших сообщений, устойчивость ко всем существующим атакам на хеш-функции, оптимизация под 64-разрядные процессоры и активное использование обращений к таблицам
blake2b	256, 512	BLAKE2 — криптографическая хеш-функция, улучшенная версия BLAKE — одного из пяти финалистов конкурса на хеш-функцию SHA-3 (главным образом улучшено быстродействие), представлена 21 декабря 2012 года. Разработчики: Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, и Christian Winnerlein. Была создана как альтернатива широко использовавшимся в прошлом MD5 и SHA-1, в которых были найдены уязвимости