

**RuBackup**

Система резервного копирования и восстановления данных

# **Резервное копирование и восстановление образов, контейнеров и томов Docker**



**RuBackup**

Версия 1.9

2022 г.

# Содержание

Введение.....	3
Принципы выполнения резервного копирования Docker.....	4
Установка клиента RuBackup.....	6
Подготовка хоста с Docker. для выполнения резервного копирования средствами RuBackup.....	7
Защитное преобразование резервных копий.....	8
Локальные листы ограничений.....	10
Использование менеджера администратора RuBackup.....	11
Настройки правил глобального расписания RuBackup.....	16
Использование клиентского менеджера RuBackup.....	18
Утилиты командной строки клиента RuBackup.....	22

## Введение

Система резервного копирования RuBackup позволяет выполнять клиентам полное резервное копирование образов (image), контейнеров (container) и томов(volume) Docker в ОС Linux.

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

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

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

Для возможности выполнения резервного копирования Docker на клиенте должны быть установлены модули RuBackup:

- rb\_module\_docker\_container;
- rb\_module\_docker\_image;
- rb\_module\_docker\_volume.

Работа вышеперечисленных модуля на клиенте возможна только в том случае, если на нем установлены необходимые пакеты Docker.

Для образов и контейнеров Docker поддерживается только полное резервное копирование, для томов Docker поддерживается полное, инкрементальное и дифференциальное резервное копирование.

Резервное копирование выполняется по заранее заданным правилам в глобальном расписании RuBackup. Так же клиенту доступно срочное резервное копирование выбранного образа, контейнера или тома Docker.

Восстановление резервной копии возможно по инициативе клиента. Для восстановления данных пользователь должен ввести пароль, позволяющий выполнить восстановление.

Резервное копирование может быть выполнено с применением сжатия на стороне клиента или на стороне сервера RuBackup. При необходимости возможно выполнить защитное преобразование резервной копии выбранным алгоритмом (см. раздел «Защитное преобразование резервных копий»).

# Принципы выполнения резервного копирования Docker

Система резервного копирования RuBackup позволяет выполнять резервное копирование образов и контейнеров Docker.

## Принципы резервного копирования образа

Для выполнения резервного копирования образа Docker используется специализированный модуль RuBackup **rb\_module\_docker\_image**.

Собственно резервное копирование модуль производит командой Docker:

```
# docker save -o archive_filename resource_id
```

где `resource_id` – ID образа.

Восстановление образа из файла производится командой

```
# docker load -I archive_filename
```

## Принципы резервного копирования контейнера

Для выполнения резервного копирования контейнера Docker используется специализированный модуль RuBackup **rb\_module\_docker\_container**.

При выполнении резервного копирования контейнера происходит создание образа на базе выбранного контейнера и выполняется резервное копирование вновь созданного образа.

Собственно резервное копирование модуль производит командами Docker:

```
# docker commit -p resource_id new_image_name:new_tag
```

где `resource_id` – ID контейнера

```
# docker save -o archive_filename resource_id
```

где `resource_id` – ID образа, который был создан предыдущей командой.

Восстановление образа из файла производится командой

```
# docker load -I archive_filename
```

### Принципы резервного копирования тома

Для выполнения резервного копирования тома Docker используется специализированный модуль RuBackup **rb\_module\_docker\_volume**.

При выполнении резервного копирования происходит запуск временного контейнера с подключённым к нему томом Docker и при помощи команды `tar` создаётся резервная копия файлов, расположенных в томе.

Собственно резервное копирование тома (например, `vol1`) выполняется командой примерно такого типа:

```
# docker run --rm --mount source=vol1,target=/rubackup-volume-data -v $(pwd)/rubackup-archive:/rubackup-archive ubuntu bash -c "cd /rubackup-volume-data && tar cvf /rubackup-archive/backup.tar ."
```

Восстановление тома происходит следующим образом:

```
# docker volume create vol1  
  
# docker run --rm --mount vol1:/recover -v ~/rubackup-archive:/backup ubuntu bash -c "cd /recover && tar xvf /backup/backup.tar"
```

В том случае, если том `vol1` существует на момент восстановления резервной копии, то восстановление произойдёт в текущий каталог, где будет создана папка с именем тома, либо в другой выбранный пользователем каталог.

## Установка клиента RuBackup

Для возможности резервного копирования Docker при помощи RuBackup на сервер должен быть установлен клиент RuBackup и соответствующие модули. Подробно процедура установки клиента описана в «Руководстве по установке серверов резервного копирования и Linux клиентов RuBackup».

Клиент RuBackup представляет собой фоновое системное приложение (демон или сервис), обеспечивающее взаимодействие с серверной группировкой RuBackup. Для выполнения резервного копирования Docker клиент RuBackup должен работать от имени суперпользователя (root для Linux и Unix).

# Подготовка хоста с Docker для выполнения резервного копирования средствами RuBackup

## 1. Установка модулей Docker RuBackup для возможности выполнения резервного копирования и восстановления ресурсов Docker

В зависимости от типа операционной системы:

```
# sudo dpkg -i ./rubackup-docker.deb
```

или

```
# sudo rpm -I ./rubackup-docker.rpm
```

## 2. Каталог для создания резервных копий и хранения временных файлов

Для создания резервных копий виртуальных машин и хранения временных файлов, которые создаются при их восстановлении, требуется определённое пространство. Рекомендуется выделить для этой цели отдельный диск или устройство хранения достаточного размера и примонтировать к */docker-backup* (либо к иной удобной точке монтирования), во избежание переполнения системного диска. Необходимо определить этот каталог как значение параметра *use-local-backup-directory* в конфигурационном файле */opt/rubackup/etc/config.file* и перезагрузить клиент RuBackup.

В исключительных случаях допустимо использование возможности сервера RuBackup предоставить клиенту NFS каталог для создания резервной копии. Для этого нужно определить значение параметра *nfs-share-mountpoint*, который определяет в какую точку файловой системы будет примонтирован NFS каталог. Параметр *use-local-backup-directory* в этом случае должен быть отключён, а на сервере RuBackup произведены соответствующие настройки для определения разделяемого каталога. Более подробно см. «Руководство системного администратора RuBackup».

# Защитное преобразование резервных копий

При необходимости, сразу после выполнения резервного копирования архивы могут быть преобразованы на хосте клиента. Таким образом, важные данные будут недоступны для администратора RuBackup или других лиц, которые могли бы получить доступ к резервной копии (например, на внешнем хранилище картриджей ленточной библиотеки или на площадке провайдера облачного хранилища для ваших резервных копий).

Защитное преобразование осуществляется входящей в состав RuBackup утилитой `gbscrypt`. Ключ для защитного преобразования резервных копий располагается на хосте клиента в файле `/opt/rubackup/keys/master-key`. Защитное преобразование данных при помощи `gbscrypt` возможно с длиной ключа 256 бит (по умолчанию), а также 128, 512 или 1024 бита в зависимости от выбранного алгоритма преобразования.

Если для правила глобального расписания необходимо выбрать особый режим защитного преобразования с длиной ключа, отличной от 256 бит, и с ключом, расположенным в другом месте, то вы можете сделать это при помощи скрипта, выполняющегося после выполнения резервного копирования (определяется в правиле глобального расписания администратором RuBackup). При этом необходимо, чтобы имя преобразованного файла осталось таким же, как и ранее, иначе задача завершится с ошибкой. Провести обратное преобразование такого файла после восстановления его из архива следует вручную при помощи утилиты `gbscrypt`. При таком режиме работы нет необходимости указывать алгоритм преобразования в правиле резервного копирования, иначе архив будет повторно преобразован с использованием мастер-ключа.

Для выполнения защитного преобразования доступны алгоритмы, представленные в таблице 1.

Таблица 1 – Алгоритмы защитного преобразования, доступные в утилите `gbscrypt`.

Алгоритм	Длина ключа, бит	Примечание
Anubis	128, 256	
Aria	128, 256	



Алгоритм	Длина ключа, бит	Примечание
CAST6	128, 256	
Camellia	128, 256	
Kalyna	128, 256, 512	Украинский национальный стандарт <u>ДСТУ 7624:2014</u>
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	

## Локальные листы ограничений

В том случае, если какие-либо конкретные ресурсы клиента не должны попасть в резервную копию, их можно включить в локальный лист ограничений на клиенте. Листы ограничений располагаются в каталоге `/opt/rubackup/etc`:

```
rubackup_restriction.list.docker_container
```

```
rubackup_restriction.list.docker_image
```

```
rubackup_restriction.list.docker_volume
```

Наименование ресурса, для которого нет необходимости выполнять резервное копирование, должно быть указано в отдельной строке соответствующего файла.

Для того, чтобы листы ограничений имели силу, необходимо включить эту возможность для клиента в конфигурации RuBackup (см. Руководство системного администратора RuBackup).

# Использование менеджера администратора RuBackup

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

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

Запуск менеджера администратора RBM:

Вариант 1:

```
# sudo LD_LIBRARY_PATH=/opt/rubackup/lib /opt/rubackup/bin/rbm
```

Вариант 2:

```
# ssh -X you_rubackup_server
```

```
# sudo LD_LIBRARY_PATH=/opt/rubackup/lib /opt/rubackup/bin/rbm
```

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

Для резервного копирования Docker на хосте должен быть установлен клиент RuBackup и соответствующий модуль, обеспечивающий резервное копирование. Клиент должен быть авторизован администратором RuBackup (см. раздел «Клиенты» менеджера администратора RuBackup).

При помощи менеджера администратора RuBackup можно создать в глобальном расписании одно или несколько правил резервного копирования образов, контейнеров или томов Docker.

Для этого необходимо выполнить следующие действия:

1. Выбрать клиентский хост, на котором установлен Docker и добавить правило резервного копирования (рисунок 2).

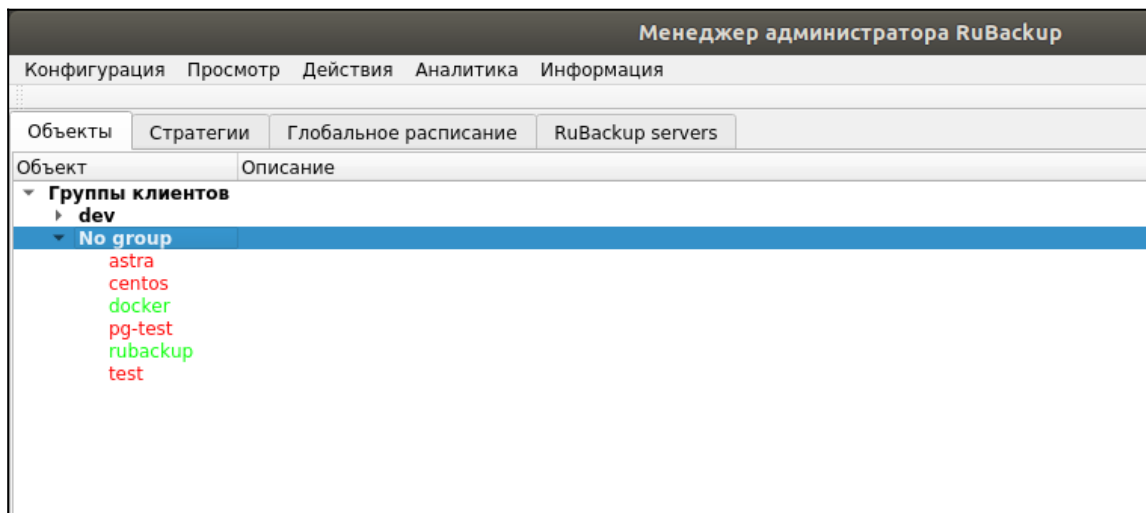


Рисунок 1

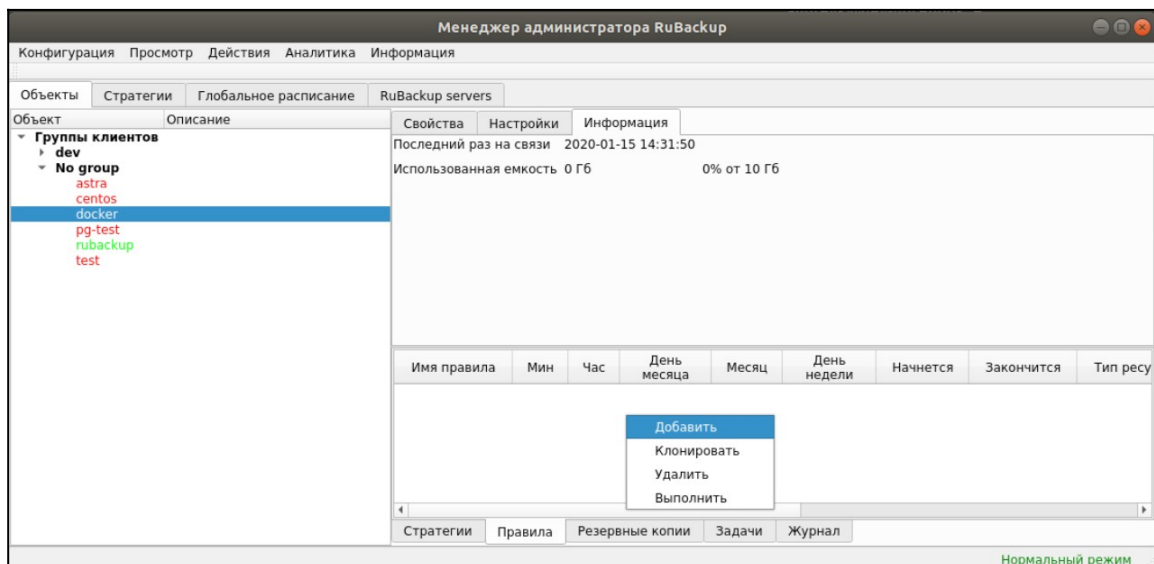


Рисунок 2

2. Выбрать тип ресурса «Docker image», «Docker container» или «Docker volume»: (рисунок 3):

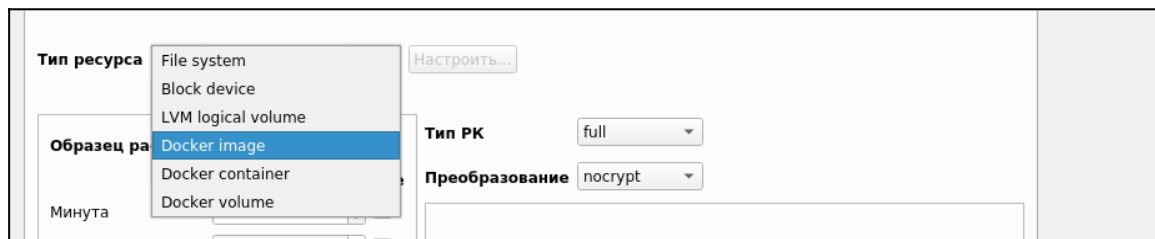


Рисунок 3

3. Выбрать ресурс, для которого будет выполняться правило (рисунок 4):

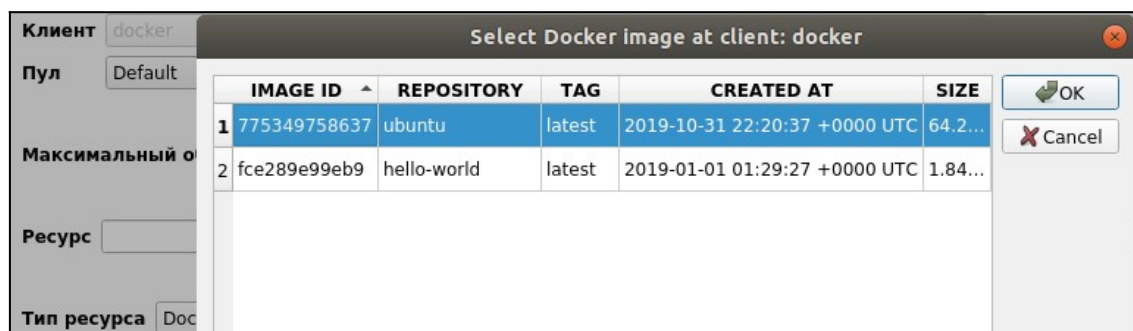
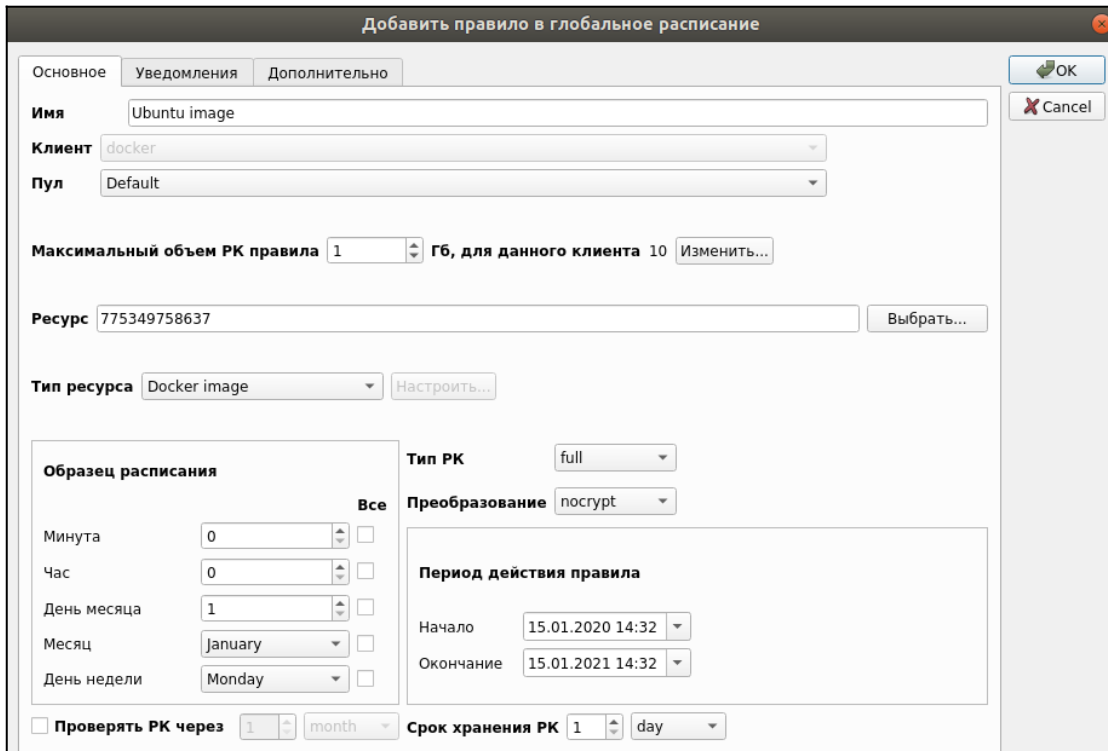


Рисунок 4

4. Установить прочие настройки: тип резервного копирования (Full), максимальный объем для резервных копий данного правила (1Гб), срок хранения (1 день), через какой промежуток времени требуется выполнить проверку резервной копии или не проверять её вовсе (рисунок 5).
5. На вкладке «Дополнительно» можно установить разрешение для клиента удалять резервные копии, установить автоматическое удаление устаревших резервных копий или определить условие их перемещения в другой пул (рисунок 6).



Добавить правило в глобальное расписание

Основное Уведомления Дополнительно

Имя

Клиент

Пул

Максимальный объем РК правила  Гб, для данного клиента 10

Ресурс

Тип ресурса

Образец расписания  Все

Минута

Час

День месяца

Месяц

День недели

Тип РК

Преобразование

Период действия правила

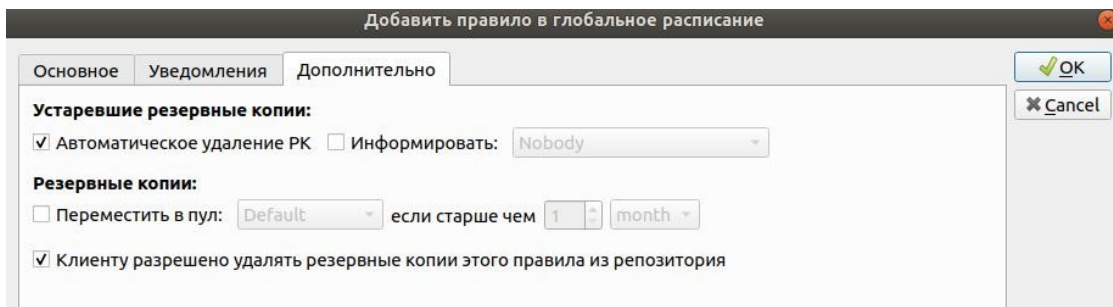
Начало

Окончание

Проверять РК через

Срок хранения РК

Рисунок 5



Добавить правило в глобальное расписание

Основное Уведомления Дополнительно

**Устаревшие резервные копии:**

Автоматическое удаление РК  Информировать:

**Резервные копии:**

Переместить в пул:  если старше чем

Клиенту разрешено удалять резервные копии этого правила из репозитория

Рисунок 6

Вновь созданное правило будет обладать статусом «wait», это означает что оно не будет порождать задач на выполнение резервного копирования до той поры, пока администратор RuBackup не запустит его и оно изменит свой статус на «run». При необходимости работу правила можно будет приостановить или запустить в любой момент времени по желанию администратора. Так же администратор может инициировать немедленное создание задачи при статусе правила «wait».

Правило глобального расписания имеет срок жизни, определяемый при его создании, а так же предусматривает следующие возможности:

- 1) Выполнить скрипт на клиенте перед началом резервного копирования.
- 2) Выполнить скрипт на клиенте после успешного окончания резервного копирования.

3) Выполнить скрипт на клиенте после неудачного завершения резервного копирования.

4) В дополнительных настройках правила резервного копирования возможно задать:

– для контейнера Docker: с каким именем образа и с каким tag восстанавливать резервную копию;

– для тома Docker: с каким именем тома восстанавливать резервную копию.

5) Выполнить преобразование резервной копии на клиенте.

6) Периодически выполнять проверку целостности резервной копии.

7) Хранить резервные копии определённый срок, а после его окончания удалять их из хранилища резервных копий и из записей репозитория, либо просто уведомлять пользователей системы резервного копирования об окончании срока хранения.

8) Через определённый срок после создания резервной копии автоматически переместить её на другой пул хранения резервных копий, например на картридж ленточной библиотеки.

9) Уведомлять пользователей системы резервного копирования о результатах выполнения тех или иных операций, связанных с правилом глобального расписания.

При создании задачи RuBackup она появляется в главной очереди задач. Отслеживать исполнение правил может как администратор, с помощью RBM, так клиент при помощи RBC.

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

# Настройки правил глобального расписания RuBackup

Для выполнения резервного копирования ресурсов Docker необходимо при помощи менеджера администратора RuBackup создать правило в глобальном расписании, в котором указать соответствующий тип ресурса. При создании правила в глобальном расписании администратор RuBackup будет видеть список всех ресурсов Docker на клиенте и может выбрать требуемый (для этого необходимо, чтобы на клиенте работал клиентский фоновый процесс).

При создании правила резервного копирования можно определить следующие параметры:

- тип резервного копирования (полный, дифференциальный или инкрементальный). Для образов и контейнеров выполняется только полный тип резервного копирования, для томов — полный, дифференциальный или инкрементальный;

- разрешенный максимальный объем для всех резервных копий правила;

- необходимость преобразования резервной копии тем или иным алгоритмом. Преобразование будет выполняться на стороне клиента;

- шаблон времени и даты создания задачи резервного копирования;

- флаг и период автоматической проверки резервной копии;

- срок хранения резервных копий создаваемого правила;

- пул хранения, в котором будут размещены резервные копии;

- необходимость автоматического удаления резервной копии, срок хранения которой истёк;

- перемещение резервной копии в другой пул, при достижении определённого срока с момента её создания;



- возможность для клиента удалять резервные копии из репозитория;
- настройки системы уведомления RuBackup для создаваемого правила;

Уведомления могут происходить в следующих случаях:

- нормальное исполнение процедуры резервного копирования;
- исполнение процедуры резервного копирования с ошибками;
- проверка резервной копии;
- окончание периода действия создаваемого правила;
- окончание выделенного объёма для хранения резервных копий правила;
- окончание срока хранения резервной копии;
- дополнительные настройки правила для выполнения резервного копирования ресурсов Docker (только для контейнеров и томов).

# Использование клиентского менеджера RuBackup

Принцип взаимодействия клиентского менеджера с системой резервного копирования состоит в том, что пользователь может сформировать ту или иную команду (желаемое действие) и отправить его серверу резервного копирования RuBackup. Взаимодействие пользователя с сервером резервного копирования производится через клиента (фоновый процесс) резервного копирования. Клиентский менеджер отправляет команду пользователя клиенту, клиент отправляет её серверу. В том случае, если действие допустимо, то сервер RuBackup отдаст обратную команду клиенту и/или перенаправит её медиасерверу RuBackup для дальнейшей обработки. Это означает, что клиентский менеджер обычно не ожидает завершения того или иного действия, но ожидает ответа от клиента, что задание принято. Это позволяет инициировать параллельные запросы клиента к серверу резервного копирования, но требует от пользователя самостоятельно контролировать чтобы не было «встречных» операций, когда происходит восстановление данных, и в этот же момент эти же данные требуются для создания новой резервной копии. После того, как вы отдали ту или иную команду при помощи клиентского менеджера, вы можете просто закрыть приложение, все действия будут выполнены системой резервного копирования (однако стоит дождаться сообщения что задание принято к исполнению и проконтролировать это во вкладке «Задачи»).

Графический интерфейс клиентского менеджера поддерживает русский и английский языки.

Запуск клиентского менеджера:

Вариант 1:

```
# sudo LD_LIBRARY_PATH=/opt/rubackup/lib/opt/rubackup/bin/rbc
```

Вариант 2:

```
# ssh -X you_rubackup_client
```

```
# sudo LD_LIBRARY_PATH=/opt/rubackup/lib /opt/rubackup/bin/rbc
```

При первом запуске клиентского менеджера необходимо задать пароль, при помощи которого впоследствии можно будет запросить восстановление резервной копии. Без ввода пароля получить резервную копию для клиента из хранилища невозможно. Хэш пароля восстановления хранится в базе данных

RuBackup сервера. При необходимости можно изменить пароль при помощи клиентского менеджера (меню «**Конфигурация**» → «**Изменить пароль**»).

На главной странице клиентского менеджера расположены переключающиеся вкладки, позволяющие управлять резервными копиями, расписанием резервного копирования и просматривать текущие задачи клиента.

### **Вкладка «Резервные копии»**

В таблице вкладки «Резервные копии» содержится информация обо всех резервных копиях клиента, которые хранятся в репозитории RuBackup (рисунок 7). Дифференциальные резервные копии ссылаются на полные резервные копии, инкрементальные резервные копии ссылаются на полные резервные копии или предыдущие инкрементальные, так что при необходимости восстановить данные можно одной командой инициировать восстановление всей цепочки резервных копий.

RuBackup менеджер клиента											
Конфигурация Вид Действия Информация											
Резервные копии			Глобальное расписание		Задачи	Локальное расписание		Ограничения			
	Id	Task ID	Reference ID	Resource type	Resource	Backup type	Pool	Archive size	Snapshot size	Created	Cr
1	451	1676		Docker container	d7c81eb3918f	full	Default	25806378	3337	2019-12-16 14:56:26+03	00:00
2	452	1678		Docker container	d7c81eb3918f	full	Default	25805388	3348	2019-12-16 15:15:57+03	00:00
3	506	1836		Docker container	4dab779bafc7	full	Default	25807028	3526	2019-12-19 22:15:03+03	00:00
4	507	1837		Docker image	775349758637	full	Default	25806191	3553	2019-12-19 22:17:03+03	00:00
5	511	1842		Docker container	4dab779bafc7	full	Default	25807084	3517	2019-12-19 22:42:04+03	00:00
6	513	1845		Block device	/dev/sde1	full	Default	21043379	65472	2019-12-19 22:51:24+03	00:00
7	514	1846		Block device	/dev/sde1	full	Default	21043384	65472	2019-12-19 22:56:56+03	00:00
8	637	2511		Docker volume	vol1	full	Default	551	91	2019-12-30 14:06:05+03	00:00

Рисунок 7

Во вкладке «Резервные копии» пользователю доступны следующие действия:

#### **Удалить выбранную резервную копию.**

Это действие возможно в том случае, если в правиле глобального расписания есть соответствующее разрешение. Кроме того, при необходимости выполнить удаление резервной копии потребуются вести пароль клиента.

#### **Восстановить цепочку резервных копий.**

Это действие запускает процесс восстановления цепочки резервных копий на системе клиента.

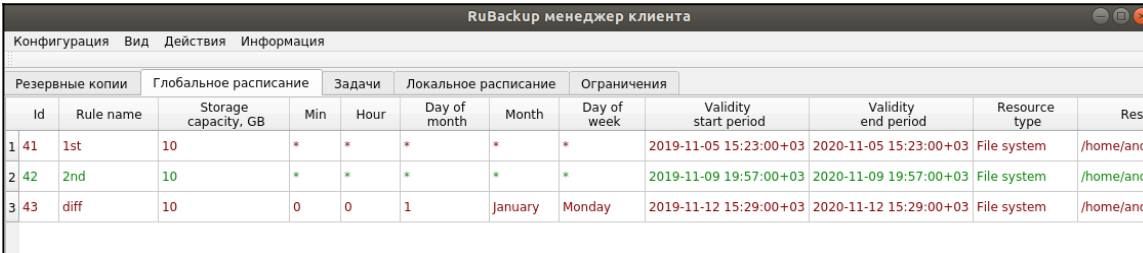
Клиентский менеджер не ожидает окончания восстановления всех резервных копий, пользователь должен проконтролировать во вкладке «Задачи» что все созданные задачи на восстановление данных завершились успешно (статус задач «Done»). Для успешного выполнения этого действия требуется наличие достаточного свободного места в каталоге, предназначенном для создания и временного хранения резервных копий (см.опцию use-local-backup-directory).

## Проверить резервную копию.

Это действие инициирует создание задачи проверки резервной копии. В том случае, если резервная копия была подписана цифровой подписью, то будут проверены размер файлов резервной копии, md5 сумма и проверена сама резервная копия. Если резервная копия не была подписана цифровой подписью, то будут проверены размер файлов резервной копии и md5 сумма.

## Вкладка «Глобальное расписание»

В таблице вкладки «Глобальное расписание» содержится информация обо всех правилах в глобальном расписании RuBackup для этого клиента (рисунок 8).



RuBackup менеджер клиента											
Конфигурация Вид Действия Информация											
Резервные копии		Глобальное расписание			Задачи		Локальное расписание		Ограничения		
Id	Rule name	Storage capacity, GB	Min	Hour	Day of month	Month	Day of week	Validity start period	Validity end period	Resource type	Reso
1 41	1st	10	*	*	*	*	*	2019-11-05 15:23:00+03	2020-11-05 15:23:00+03	File system	/home/andr
2 42	2nd	10	*	*	*	*	*	2019-11-09 19:57:00+03	2020-11-09 19:57:00+03	File system	/home/andr
3 43	diff	10	0	0	1	January	Monday	2019-11-12 15:29:00+03	2020-11-12 15:29:00+03	File system	/home/andr

Рисунок 8

Во вкладке «Глобальное расписание» пользователю доступны следующие действия:

### Запросить новое правило.

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

### Запросить удалить правило из глобального расписания.

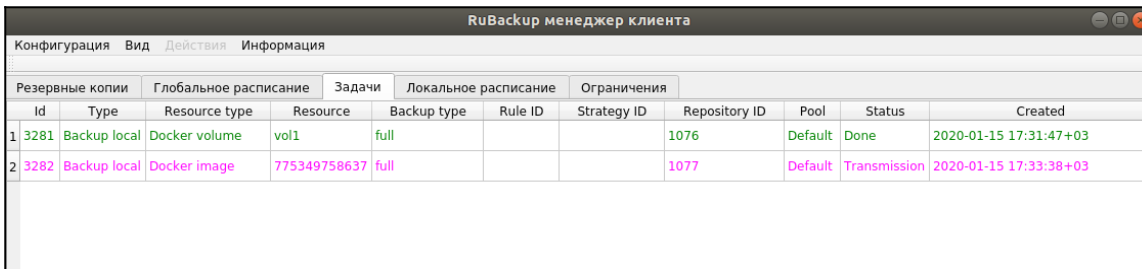
Это действие формирует запрос к администратору RuBackup об удалении выбранного пользователем правила из глобального расписания RuBackup. Запрос на удаление правила требует одобрения администратора RuBackup, одобрение может быть сделано в оконном менеджере администратора RuBackup.

## Вкладка «Задачи»

В таблице вкладки «Задачи» содержится информация обо всех задачах в главной очереди заданий RuBackup для этого клиента (рисунок 9).

В зависимости от настроек резервного сервера RuBackup выполненные задачи и задачи, завершившиеся неудачно, через какое-то время могут быть автоматически удалены из главной очереди задач. Информация о выполнении заданий фиксируется в специальном журнале задач сервера RuBackup, при необходимости статус любой задачи, даже удалённой из очереди, можно уточнить у администратора RuBackup. Так же информация о выполнении задач клиента заносится в локальный журнальный файл на клиенте. В

клиентском менеджере можно открыть окно отслеживания журнального файла (меню «Информация» → «Журнальный файл»).



RuBackup менеджер клиента											
Конфигурация Вид Действия Информация											
Резервные копии		Глобальное расписание		Задачи		Локальное расписание		Ограничения			
Id	Type	Resource type	Resource	Backup type	Rule ID	Strategy ID	Repository ID	Pool	Status	Created	
1	3281	Backup local	Docker volume	vol1	full		1076	Default	Done	2020-01-15 17:31:47+03	
2	3282	Backup local	Docker image	775349758637	full		1077	Default	Transmission	2020-01-15 17:33:38+03	

Рисунок 9

### Вкладка «Локальное расписание»

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

### Вкладка «Ограничения»

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

# Утилиты командной строки клиента

## RuBackup

Для управления RuBackup со стороны клиента, помимо клиентского оконного менеджера, можно воспользоваться утилитами командной строки:

### rb\_archive

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

```
root@docker:~#
root@docker:~# rb_archives
Id | Ref ID | Resource | Resource type | Backup type | Created | Crypto | Signed | Status
-----+-----+-----+-----+-----+-----+-----+-----+-----
1072 | | ubuntu | Docker image | full | 2020-01-15 16:48:18+03 | nocrypt | True | Not Verified
root@docker:~#
root@docker:~#
```

### rb\_schedule

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

```
root@docker:~# rb_schedule
Id | Name | Resource type | Resource | Backup type | Status
---+---+-----+-----+-----+-----
95 | Ubuntu image | Docker image | 775349758637 | full | wait
root@docker:~#
```

### rb\_tasks

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

```
root@docker:~# rb_schedule
Id | Name | Resource type | Resource | Backup type | Status
---+---+-----+-----+-----+-----
95 | Ubuntu image | Docker image | 775349758637 | full | wait
root@docker:~#
```

Ознакомиться с функциями утилит командной строки можно при помощи команды `man` или в руководстве «Утилиты командной строки RuBackup».