

RuBackup

Система резервного копирования и восстановления данных

Утилиты командной строки RuBackup



RuBackup

Версия 1.9

2022 г.

Содержание

Введение.....	4
Утилиты командной строки администратора RuBackup.....	5
1. rb_bandwidth.....	6
2. rb_block_devices.....	8
3. rb_client_group.....	9
4. rb_clients.....	10
5. rb_clouds.....	13
6. rb_cloud_task_queue.....	14
7. rb_copy2pool.....	15
8. rbd.....	17
9. rbfd.....	19
10. rb_global_config.....	23
11. rb_global_schedule.....	24
12. rb_init.....	29
13. rb_inventory.....	30
14. rb_local_filesystems.....	31
15. rb_log_viewer.....	33
16. rb_media_servers.....	35
17. rb_modules.....	37
18. rb_notifications.....	38
19. rb_pools.....	40
20. rb_remote_replication.....	42
21. rb_repository.....	45
22. rb_tape_cartridges.....	47
23. rb_tape_libraries.....	49
24. rb_task_queue.....	52
25. rb_tl_task_queue.....	55
26. rb_strategies.....	57

27. rb_user_groups.....	60
28. rb_users.....	61
29. rb_update.....	63
Утилиты командной строки клиента RuBackup.....	65
1. rb_archives.....	66
2. rbcrypt.....	69
3. rb_schedule.....	71
4. rb_tasks.....	72
Приложения.....	73
1 Алгоритмы хеш-функций, реализованные в RBD и RBFД.....	74
2 Алгоритмы преобразования в RBCRYPT и RBFД.....	76
3 Авторские права.....	77

Введение

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.

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

Утилиты командной строки администратора RuBackup

1. rb_bandwidth

НАЗВАНИЕ

rb_bandwidth - управление ограничениями пропускной способности для клиентов и правил глобального расписания RuBackup.

СИНТАКСИС

```
-h  
-l  
-L  
-c Имя_хоста -b Ограничение_бэкапа -g Ограничение_восстановления -  
s Время_начала -e Время_конца  
-g ID_правила -b Ограничение_бэкапа -g Ограничение_восстановления  
-s Время_начала -e Время_конца  
-x ID  
-z ID
```

ОПИСАНИЕ

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**.

ПАРАМЕТРЫ

-h

Справочное сообщение.

-l

Список ограничений для клиентов.

-L

Список ограничений для правил глобального расписания.

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

Установить новые ограничения для клиента *Имя_хоста*. Параметр **-b** определяет максимально допустимую пропускную способность при выполнении операций резервного копирования в МБ/с. Параметр **-g** определяет максимально допустимую пропускную способность при выполнении операций восстановления в МБ/с. Параметр **-s** определяет время начала действия ограничения, например, 00:01:02. Параметр **-e** 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* -g *Ограничение_восстановления* -s *Время_начала* -e *Время_конца*

Установить новые ограничения для правила глобального расписания с идентификатором *ID_правила* (узнать идентификаторы правил глобального расписания можно с помощью утилиты `gb_global_schedule`). Параметр **-b** определяет максимально допустимую пропускную способность при выполнении операций резервного копирования в МБ/с. Параметр **-g** определяет максимально допустимую пропускную способность при выполнении операций восстановления в МБ/с. Параметр **-s** определяет время начала действия ограничения, например, 00:01:02. Параметр **-e** 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*. Чтобы получить значение идентификатора используйте команду `gb_bandwidth -l`.

-z *ID*

Удалить из конфигурации ограничений правил глобального расписания ограничение с идентификатором *ID*. Чтобы получить значение идентификатора используйте команду `gb_bandwidth -L`.

2. rb_block_devices

НАЗВАНИЕ

rb_block_devices - управление блочными устройствами RuBackup.

СИНТАКСИС

```
-h  
-v  
-a Блочное_устройство -p ID_пула [ -o ] [ -d 'Описание' ]  
-r ID
```

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-h
Справочное сообщение.

-v
Список блочных устройств в конфигурации. Предоставляет информацию в расширенном формате.

-a *Блочное_устройство* -p *ID_пула* [-o] [-d '*Описание*']

Добавить блочное устройство с именем *Блочное_устройство* и, при необходимости, соответствующим описанием, в пул хранения с *ID_пула*. Параметр **-o** позволяет перезаписать файловую систему на блочном устройстве.

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

-r *ID*

Удалить из конфигурации блочное устройство с идентификатором *ID*. Для получения значения идентификатора блочного устройства используйте команду `rb_block_devices -l`.

3. rb_client_group

НАЗВАНИЕ

rb_client_group - управление группами клиентов RuBackup.

СИНТАКСИС

```
-h  
-l  
-a Имя_группы [ -d 'Описание' ]  
-г ID  
-с ID [ -п Новое_имя_группы ] [ -d 'Новое описание' ] ]
```

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-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'.

-г *ID*

Удалить из конфигурации группу с идентификатором *ID*. Для получения значения идентификатора группы используйте команду `rb_client_group -l`.

-с *ID* [-п *Новое_имя_группы*] [-d '*Новое описание*']

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

4. rb_clients

НАЗВАНИЕ

rb_clients - управление клиентами RuBackup.

СИНТАКСИС

```
-h  
-l [ -o ]  
-u [ -o ] [ -v ]  
-a Имя_хоста [ -g ID_группы ] [ -p ID_пула ]  
[ -s Емкость_хранилища ] [ -w allow_или_deny ] [ -z allow_или_deny ]  
[ -4 ipv4 ] [ -6 ipv6 ] [ -m macaddr ] [ -d 'Описание' ]  
[ -H hwid ]  
-v  
-r ID  
-x ID  
-t ID  
-c ID [ -d 'Новое описание' ] [ -g ID_группы ] [ -p ID_пула ] [ -s  
Емкость_хранилища ] [ -w allow_или_deny ] [ -z allow_или_deny ]  
-y ID
```

ОПИСАНИЕ

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 *ёмкости хранилища* (в ГБ).

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

ПАРАМЕТРЫ

-h

Справочное сообщение.

-l [-o]

Список авторизованных клиентов. Параметр `-o` выдаёт информацию только о клиентах, которые работают в данный момент времени.

-u [-o] [-v]

Список неавторизованных клиентов. Параметр `-o` выдаёт информацию только о клиентах, на которых в данный момент времени запущен клиентский процесс RuBackup. Параметр `-v` предоставляет расширенную информацию.

-a *Имя_хоста* [-g *ID_группы*] [-p *ID_пула*] [-s *Емкость_хранилища*] [-w *allow_или_deny*] [-z *allow_или_deny*] [-4 *ipv4*] [-6 *ipv6*] [-m *macaddr*] [-d '*Описание*'] [-h *hwid*]

Добавить в конфигурацию RuBackup нового клиента. Параметр `-g` позволяет соотнести клиента с клиентской группой. В том случае, если на клиенте будут выполняться локальные правила резервного копирования, то эти резервные копии будут размещаться в пуле, определённом параметром `-p`. Локальные правила резервного копирования для клиента находятся в файле `/opt/rubackup/etc/rubackup.lsf`. Если необходимо разрешить клиенту исполнять правила локального расписания, то нужно использовать параметр `-w allow`, в противном случае используйте `-w deny`. Параметр `-s` определяет максимальный объём всех резервных копий клиента в системе резервного копирования. Параметр `-4` определяет IP адрес v4, `-6` - IP адрес v6, `-m` - MAC адрес.

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`. Локальный 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. Чтобы разрешить клиенту использовать локальный лист запретов, используйте параметр `-z allow`, в противном случае используйте `-z deny`.

-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 '*Новое описание*'] [-g *ID_группы*] [-p *ID_пула*] [-s *Емкость_хранилища*] [-w *allow_или_deny*] [-z *allow_или_deny*]

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*.

5. rb_clouds

НАЗВАНИЕ

rb_clouds - управление облаками S3 для использования их в качестве хранилищ для пулов типа “Облако”.

СИНТАКСИС

```
-h  
-v  
-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 ]  
-r ID
```

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-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 ]
```

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

Внимание! С каждым пулом может быть ассоциировано только одно облако.

-r *ID*
Удалить облако из конфигурации.

6. rb_cloud_task_queue

НАЗВАНИЕ

rb_cloud_task_queue - просмотр информации о текущих задачах, связанных с облачными операциями.

СИНТАКСИС

-h
-l
-e

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-h
Справочное сообщение.
-l
Получить информацию о всех задачах.
-e
Удалить ошибочные задачи.

7. rb_copy2pool

НАЗВАНИЕ

rb_copy2pool - управление репликацией резервных копий при их создании.

СИНТАКСИС

```
-h  
-l  
-L  
-a ID_правила -p ID_пула  
-A ID_стратегии -p ID_пула  
-r ID  
-R ID
```

ОПИСАНИЕ

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 сразу 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.

ПАРАМЕТРЫ

```
-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

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

-A ID_стратегии -p ID_пула

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

-g ID

Удалить запись из пулов репликации правил резервного копирования.

-R ID

Удалить запись из пулов репликации стратегий резервного копирования.

8. rbd

НАЗВАНИЕ

rbd - дифференциация информации, содержащейся в файле, и создание разностных копий.

СИНТАКСИС

```
-h
-i inputfile1 [ -o hashfile1 ] [ -o outputfile1 ] [ -a
hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
-i inputfile2 -c hashfile1 [ -p inputfile2patch ] [ -o hashfile2 ]
[ -a hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
-i inputfile1 -x inputfile2patch -n restoredfile [ -a
hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
```

ОПИСАНИЕ

rbd - утилита администратора RuBackup для дифференциации информации, которая содержится в файле. Утилита создаёт хеш-таблицу файла, на основании которой может быть впоследствии создана разностная копия изменённой версии файла (патч-файл). При помощи утилиты можно воссоздать файл из его старой и разностной копий.

В утилите **rbd** реализованы алгоритмы хеш-функций blake2b, sha, skein и streebog (GOST_R_34_11_2012) с длиной хеша 256 и 512 бит. Подробную информацию о реализованных алгоритмах см. *Приложение 1 «Алгоритмы хеш-функций, реализованные в RBD и Rbfd»*.

ПАРАМЕТРЫ

-h
Справочное сообщение.

```
-i inputfile1 [ -o hashfile1 ] [ -o outputfile1 ] [ -a
hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
```

Для входного файла *inputfile1* создать хеш-таблицу. Параметр **-o** позволяет задать имя файла, в который будет записана хеш-таблица (по умолчанию имя файла будет иметь вид *inputfile1.hashalgorithm_blocksize.ht*, например, *test1.tgz.sha_512.ht*). Параметр **-o** позволяет задать имя файла, в который будет записано содержимое файла *inputfile1*, таким образом, будет сделана его копия. Дополнительные параметры позволяют задать хеш-функцию, размер блока данных и длину хеша.

```
-i inputfile2 -c hashfile1 [ -p inputfile2patch ] [ -o hashfile2 ]  
[ -a hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
```

Для изменённого входного файла *inputfile2* создать хеш-таблицу, и на основе данных, содержащихся в хэш-таблице первого файла *hashfile1*, создать разностную копию. В результате команды будут созданы два файла: хеш-таблица *hashfile2* и разностная копия *inputfile2patch*. Параметр *-p* позволяет задать имя файла, в который будет записана разностная копия (по умолчанию имя файла будет иметь вид *inputfile2.pt*). Параметр *-o* позволяет задать имя файла, в который будет записана хеш-таблица (по умолчанию имя файла будет иметь вид *inputfile2.hashalgorithm_blocksize.ht*, например, *test2.tgz.sha_512.ht*). Дополнительные параметры позволяют задать хеш-функцию, размер блока данных и длину хеша.

```
-i inputfile1 -x inputfile2patch -n restoredfile [ -a  
hashalgorithm ] [ -b blocksize ] [ -2 | -5 ] [ -v ]
```

На основе входного файла *inputfile1* и разностной копии *inputfile2patch* воссоздать файл, идентичный изменённому файлу (*inputfile2*), под именем *restoredfile*. Дополнительные параметры позволяют задать хеш-функцию, размер блока данных и длину хеша.

```
[ -a hashalgorithm ]
```

Хеш-функция. Можно выбрать следующие варианты: *streebog* (или *GOST_R_34_11_2012*), *sha*, *skein* или *blake2b*. Значение по умолчанию *streebog*.

```
[ -b blocksize ]
```

Задать размер блока данных (байт). Размер по умолчанию 16384 байта, максимальный размер блока данных 1048576 байт.

```
[ -2 | -5 ]
```

Указать длину хеша (бит). Возможные варианты 256 бит или 512 бит (по умолчанию 512 бит).

```
[ -v ]
```

Вывести расширенную информацию о работе утилиты.

9. rbfd

НАЗВАНИЕ

rbfd - создание и восстановление полных и инкрементальных резервных копий блочных устройств, файлов и каталогов в любых файловых системах.

СИНТАКСИС

-h

```
-c resource [ -v | -V ] [ -C crypto_algorithm -k key_file ] [ -a hash_algorithm ] [ -2 | -5 ] [ -b block_size ] [ -t threads ] [ -y ] -A archive_file -m meta_file [ -S reference_meta_file ] [ -x exclude_file_or_catalog ] [ -X exclude_all_from_file ] [ -w ]
```

```
-c resource [ -v | -V ] [ -C crypto_algorithm -k key_file ] [ -a hash_algorithm ] [ -2 | -5 ] [ -b block_size ] [ -t threads ] [ -y ] -A archive_name [ -S reference_meta_file ] [ -x exclude_file_or_catalog ] [ -X exclude_all_from_file ] [ -w ]
```

```
-r archive_file [ -d destination_path ] [ -f filename ] [ -C crypto_algorithm -k key_file ] [ -v | -V ] [ -z ]
```

```
-r archive_meta_file [ -d destination_path ] [ -f filename ] [ -t threads ] [ -C crypto_algorithm -k key_file ] [ -v | -V ] [ -z ]
```

```
-l archive_meta_file [ -v | -V ]
```

ОПИСАНИЕ

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 метаданными снимка состояния.

В режиме Catalog утилита создаёт файл с мета-данными и каталог, содержащий уникальные блоки данных резервной копии в виде файлов.

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 файл, содержащий список файлов. В этом случае файл должен начинаться со строки:

```
RBFD file list:
```

В утилите **rbfd** реализованы алгоритмы хеш-функций blake2b, sha, skein и streebog (GOST_R_34_11_2012) с длиной хеша 256 и 512 бит. Подробную информацию о реализованных алгоритмах см. *Приложение 1 «Алгоритмы хеш-функций, реализованные в RBD и RBFD»*.

С помощью опции **-l** можно получить список файлов резервной копии не только для **.meta** файла, но и для архива **.rbfd**, созданного в режиме.

В том случае, если в режиме **single_file_mode** не требуется создания мета файла, то нужно использовать опцию **-m -**

Имеется возможность добавлять файлы в архив

```
--update single_archive --add file0,file1,dir0,dir1 [ -m meta_file ]
```

Есть возможность подавать на вход несколько ресурсов, например:

```
rbfd -c file0,file1,dir0,dir1 -A 1.rbfd -m -
```

Утилита **rbfd** поддерживает алгоритмы защитного преобразования. Возможны следующие варианты: anubis, aria, cast6, camellia, kalyna, kuznyechik (или GOST_R_34_12_2015), mars, rijndael (или AES), serpent, simon, sm4, speck, threefish, twofish. Подробную информацию о реализованных алгоритмах см. *Приложение 2 «Алгоритмы преобразования в RBCRYPT и RBFD»*.

ПАРАМЕТРЫ

-h

Справочное сообщение.

```
-c resource [ -v | -V ] [ -C crypto_algorithm -k key_file ] [ -a hash_algorithm ] [ -2 | -5 ] [ -b block_size ] [ -t threads ] [ -y ] -A archive_file -m meta_file [ -S reference_meta_file ] [ -x exclude_file_or_catalog ] [ -X exclude_all_from_file ] [ -w ]
```

Для ресурса *resource* создать резервную копию с именем *archive_file*. В результате будет создан файл с именем *archive_file* и файл *archive_file.meta* с метаданными (режим Single archive).

Параметр `-S` позволяет создать инкрементальную копию на основе файла `reference_meta_file`. Параметр `-y` позволяет обработать имена файлов как абсолютные пути (по умолчанию будут использованы относительные пути файлов, если это применимо). Дополнительные параметры позволяют задать алгоритм защитного преобразования, хеш-функцию, размер блока данных, длину хеша, количество потоков выполнения операции, а также исключить файл или каталог.

```
-c resource [ -v | -V ] [ -C crypto_algorithm -k key_file ] [ -a  
hash_algorithm ] [ -2 | -5 ] [ -b block_size ] [ -t threads ] [ -y  
] -A archive_name [ -S reference_meta_file ] [ -x  
exclude_file_or_catalog ] [ -X exclude_all_from_file ] [ -w ]
```

Для ресурса `resource` создать резервную копию с именем `archive_file`. В результате будет создан каталог с именем `archive_name`, содержащий уникальные блоки данных, и файл `archive_name.meta` с метаданными (режим Catalog).

Параметр `-S` позволяет создать инкрементальную копию на основе файла `reference_meta_file`. Параметр `-y` позволяет обработать имена файлов как абсолютные пути (по умолчанию будут использованы относительные пути файлов, если это применимо). Дополнительные параметры позволяют задать алгоритм защитного преобразования, хеш-функцию, размер блока данных, длину хеша, количество потоков выполнения операции, а также исключить файл или каталог.

```
-r archive_file [ -d destination_path ] [ -f filename ] [ -C  
crypto_algorithm -k key_file ] [ -v | -V ] [ -z ]
```

Восстановить резервную копию из файла `archive_file` (режим Single archive). При наличии цепочки инкрементальных резервных копий, восстановление следует осуществлять последовательно, начиная с полной, в один и тот же каталог. Параметр `-d` позволяет задать папку, в которую будет произведено восстановление. Параметр `-f` позволяет восстановить отдельный файл из архива. Параметр `-z` удаляет в каталоге назначения файлы, которых нет в архиве.

```
-r archive_meta_file [ -d destination_path ] [ -f filename ] [ -t  
threads ] [ -C crypto_algorithm -k key_file ] [ -v | -V ] [ -z ]
```

Восстановить резервную копию из каталога, содержащего уникальные блоки данных (режим Catalog). При наличии цепочки инкрементальных резервных копий, восстановление следует осуществлять последовательно, начиная с полной, в один и тот же каталог. Параметр `-d` позволяет задать папку, в которую будет произведено восстановление. Параметр `-f` позволяет восстановить отдельный файл из архива. Параметр `-z` удаляет в каталоге назначения файлы, которых нет в архиве.

```
-l archive_meta_file [ -v | -V ]
```

Вывести список файлов резервной копии для `.meta` файла и для архива `.rbfd`.

[**-C** *crypto algorithm* **-k** *key_file*]

Алгоритм защитного преобразования. Возможны следующие варианты: anubis, aria, cast6, camellia, kalyna, kuznyechik (или GOST_R_34_12_2015), mars, rijndael (или AES), serpent, simon, sm4, speck, threefish, twofish. Значение по умолчанию kuznyechik.

[**-a** *hash_algorithm*]

Хеш-функция. Можно выбрать следующие варианты: streebog (или GOST_R_34_11_2012), sha, skein или blake2b. Значение по умолчанию streebog.

[**-b** *blocksize*]

Задать размер блока данных (байт). Размер по умолчанию 16384 байта, максимальный размер блока данных 1048576 байт.

[**-2** | **-5**]

Указать длину хеша (бит). Возможные варианты 256 бит или 512 бит (по умолчанию 512 бит).

[**-t** *threads*]

Задать количество потоков. Этот параметр работает только с при создании и восстановлении полной резервной копии. При работе с инкрементальной резервной копией количество потоков равно 1.

[**-v** | **-V**]

Вывести расширенную информацию о работе утилиты. Параметр -V выводит полную информацию об атрибутах файлов.

[**-w**]

Рассчитать коэффициент дедупликации.

[**-z**]

При восстановлении, удалить в каталоге назначения файлы, которых нет в архиве.

10. rb_global_config

НАЗВАНИЕ

rb_global_config - управление глобальной конфигурацией RuBackup.

СИНТАКСИС

-h
-l
-o *ПАРАМЕТР*
-s *ПАРАМЕТР=ЗНАЧЕНИЕ*

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-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.**

11. rb_global_schedule

НАЗВАНИЕ

rb_global_schedule - управление глобальным расписанием RuBackup.

СИНТАКСИС

```
-h  
-l  
-s  
-n  
-a  
-X ID  
-i Файл_импорта  
-e ID -f Файл_экспорта  
-d ID  
-w ID  
-r ID  
-x ID  
-C ID -E Параметр:Значение,Параметр1:Значение1,...
```

ОПИСАНИЕ

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 инициирует 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 инициировать 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, а также необходимость и условия перемещения резервных копий правила, их удаления или верификации.

Создать новое правило с помощью утилиты **rb_global_schedule** можно, используя заранее подготовленный файл, описывающий все параметры правила (см.пример):

#Начало примера

#Название правила

Name: New rule name

#Имя клиента, hostname.

Client: antares

#Имя пула, в котором будут располагаться резервные копии

Pool: Default

#Суммарный разрешенный размер всех резервных копий правила в ГБ

Storage capacity: 50

*#Условия старта. * означает все значения*

#минуты(0-59). В данном случае правило создаст задачу в 0 минут

#при соблюдении остальных условий старта

Minute: 0

#часы(0-23). В данном случае правило создаст задачу в 0 часов

#при соблюдении остальных условий старта

Hour: 0

#день месяца(1-31). В данном случае правило будет создавать задачу каждый день при соблюдении остальных условий старта

Day of month: *

#месяц (1-12). В данном случае правило будет создавать задачу каждый месяц при соблюдении остальных условий старта

Month: *

#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

Day of week: *

#Начало действия правила

Validity start period: 2019-06-17 18:15:00

#Окончание действия правила

Validity end period: 2020-06-17 18:15:00

#Продолжительность хранения резервных копий (days, weeks, months, years)

Storage duration: 2 days

#Проверять или не проверять резервные копии после их создания (true, false)

```
Verify flag: true
#Интервал проверки резервных копий (days, weeks, months, years).
Verify interval: 1 day
#Тип ресурса (см.актуальный список типов ресурсов
#и их названия в вашей лицензии)
Resource type: File system
#Ресурс (файл, директория, виртуальная машина, база данных и пр.
#в зависимости от типа ресурса)
Resource: /home/andreyk/ST/
#Тип резервной копии (full, incremental, differential)
Backup type: full
#Алгоритм защитного преобразования резервных копий.
#См.актуальный перечень алгоритмов в gbscrypt.
Crypto: kuznyechik
#Удалять или не удалять резервные копии,
# срок хранения которых истек (true, false)
Auto delete obsoleted archive: true
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы о наличии устаревшей резервной копии или
#об ее удалении
Who will be informed when archive is obsoleted: RuBackup
administrators
#Перемещать ли резервную копию автоматически в другой пул
#по достижении определенного срока ее хранения (true, false)
Auto moving archive: true
#Пул, в которой будут автоматически перемещены резервные копии
#по достижении определенного срока хранения (см.gb_pools)
Move to pool: TL pool
#Срок, по истечении которого, резервные копии правила могут быть
#перемещены в другой пул
Move archive if older than: 1 day
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы об успешном выполнении задачи
#резервного копирования данного правила
Notify normal: RuBackup administrators
#Отправить копию по адресу в случае успешного выполнении
#задачи резервного копирования
Notify normal CC: you@mail.ru
#Выполнить скрипт
Normal execution script:
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы об ошибочном выполнении задачи
#резервного копирования данного правила
Notify exception: Nobody
#Отправить копию по адресу в случае ошибочного выполнении
```

```
#задачи резервного копирования
Notify exception CC:
#Выполнить скрипт в случае ошибочного выполнения
#задачи резервного копирования
Exception execution script:
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы о результатах проверки резервной копии
Notify verify: Nobody
#Отправить копию
Notify verify CC:
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы об окончании срока действия правила
Notify if end of rule validity: Nobody
#Отправить копию
Notify if end of rule validity CC:
#Группа пользователей (см.gb_user_groups), которые будут
#проинформированы об окончании квоты, выделенной
#для хранения резервных копий данного правила
Notify if end of storage capacity: Nobody
#Отправить копию
Notify if end of storage capacity CC:
#Разрешено ли клиенту создавать задачу для
#удаления своих резервных копий (true, false)
Allow the client to remove archives: false
#Окончание примера
```

Примеры файлов для импорта правил можно найти в каталоге /opt/rubackup/гс.

ПАРАМЕТРЫ

-h

Справочное сообщение.

-l

Список правил глобального расписания и их базовые параметры.

-s

Список правил глобального расписания и их временные параметры.

-a

Список правил глобального расписания и их расширенные параметры.

-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 правила с идентификатором *ID* 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.

12. rb_init

НАЗВАНИЕ

rb_init - первоначальное конфигурирование клиента или сервера RuBackup.

ИСПОЛЬЗОВАНИЕ

rb_init - утилита администратора RuBackup для первоначального конфигурирования 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 конфигурирование 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.

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*».

13. rb_init (неинтерактивный режим)

НАЗВАНИЕ

rb_init — первоначальное конфигурирование клиента или сервера RuBackup в неинтерактивном режиме.

СИНТАКСИС

Общие параметры

- h
- y
- n node type
- [-P parallelizm]
- [-v]

Конфигурация основного сервера

- H dbhost
- X dbpasswd-postgres
- Y dbpasswd-rubackup
- i iface
- f path
- l local backup directory
- [-O if-db-exists-opt]
- [-S secondary server]
- [-p parallel tasks]
- [-D dedup memory size]
- [-r remote-replication]
- [-R centrflized-recovery]
- [-g digital-signature]
- [-P parallelizm]
- [-v verbose]

Конфигурация автономного клиента

- a archive catalog
- [-z compressor]
- [-d dedicated device or network share]
- [-s network share type]
- [-o mount option]
- [-l local backup directory]
- [-P parallelizm]
- [-v verbose]

Конфигурация клиент (клиент-серверная архитектура)

- c primary server
- i iface

```
[ -S secondary server ]  
[ -l local backup directory ]  
[ -p parallel tasks ]  
[ -D dedup memory size ]  
[ -r remote-replication ]  
[ -R centrflized-recovery ]  
[ -g digital-signature ]  
[ -P parallelizm ]  
[ -v verbose ]
```

Конфигурация дополнительного сервера

```
-c primary server  
-H dbhost  
-Y dbpasswd-rubackup  
-i iface  
-l local backup directory  
[ -p parallel tasks ]  
[ -D dedup memory size ]  
[ -r remote-replication ]  
[ -R centrflized-recovery ]  
[ -g digital-signature ]  
[ -P parallelizm ]  
[ -v verbose ]
```

Конфигурация медиасервера

```
-c primary server  
-H dbhost  
-Y dbpasswd-rubackup  
-i iface  
-l local backup directory  
[ -S secondary-server ]  
[ -p parallel tasks ]  
[ -D dedup memory size ]  
[ -r remote-replication ]  
[ -R centrflized-recovery ]  
[ -g digital-signature ]  
[ -P parallelizm ]  
[ -v verbose ]
```

ОПИСАНИЕ

Неинтерактивный режим необходим для сценариев массового развертывания, например, при использовании ansible. В этом режиме администратор имеет возможность конфигурировать RuBackup в bash/shell однострочной командой и, как следствие, использовать эту команду в скриптах для автоматизации процесса.

ПАРАМЕТРЫ

-h

Справочное сообщение.

-y

Вы должны указать эту опцию для входа в неинтерактивный режим (автоматическое принятие условий лицензионного соглашения).

-n

Определяет тип конфигурируемого узла.

Аргументы: `primary-server` | `secondary-server` | `media-server` | `client` (для любого типа).

[-P parallelizm]

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.

-a [--autonomous]

Определяет клиента, как автономный.

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

-c [--primary-fqdn]

Указатель на основной сервер.

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

-d [--dedicated-device]

Опция определяет, что мы хотим использовать в качестве хранилища внешнее устройств (usb/внешний ssd) или сетевую шару.

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

-s [--net-share-type]

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

Аргументы: `nfs` | `cifs`

`-o` [`--mount-options`]

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

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

`-D` [`--dedup-memory`]

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

Аргумент: неотрицательное целое число.

`-f` [`--required-filesystem`]

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

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

`-g` [`--dig-signature`]

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

`-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).

`-i` [`--client-iface`]

Сетевой интерфейс клиента для связи с сервером.

Аргументы: Укажите имя клиентского сетевого интерфейса (напр. `eth0`).

`-l` [`--local-backup-dir`]

Определение директории для временных операций бэкапа. Если локальный каталог на клиенте не используется, то будет использоваться сетевая шара. Для конфигурации сервера эта опция обязательна.

Аргумент: укажите абсолютный путь к директории (напр. `Rubackup-tmp`).

`-O` [`--if-db-exists-opt`]

Опция используется при при реконфигурации основного сервера (если база данных `rubackup` существует) и указывает, что необходимо сделать с БД.

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

При пересоздании и `update` БД автоматически производится бэкап устаревшей версии БД при помощи утилиты `pg_dump`!

-p [`--parallel-tasks`]

Возможность ограничить параллельное выполнение заданий (значение 2 по умолчанию).

Аргумент: целое число ≤ 2 .

-r [`--remote-replica`]

Разрешает выполнять удаленную репликацию для клиента.

-R [`--centr-recovery`]

Разрешает выполнять централизованное восстановление для ресурсов клиента.

-S [`--secondary-fqdn`]

Указатель на расположение дополнительного сервера резервного копирования.

Аргументы: укажите FQDN дополнительного сервера (не рекомендуется использовать ip адрес)

-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>`.

-z [`--compressor`]

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

Аргументы: `none` | `gzip` | `pigz` | `xz`.

14. rb_inventory

НАЗВАНИЕ

rb_inventory - инвентаризация резервных копий RuBackup.

СИНТАКСИС

-h
-i *Каталог* [-с *Файл_конфигурации*] [-у] [-v] [-f]

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-h
Справочное сообщение.

-i *Каталог*
Провести инвентаризацию резервных копий, находящихся в указанном *Каталоге*.

[-с *Файл_конфигурации*]
Использовать нестандартный *Файл_конфигурации* для инвентаризации резервных копий.

[-у]
Заранее соглашаться с импортом всех резервных копий, которые не были обнаружены в базе данных.

[-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.

15. rb_local_filesystems

НАЗВАНИЕ

rb_local_filesystems - управление хранилищами резервных копий типа Файловая система (File system) медиасерверов RuBackup.

СИНТАКСИС

```
-h  
-l  
-a Путь -p ID_пула [ -d 'Описание' ]  
-г ID  
-с ID -d 'Новое описание'
```

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

```
-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. Параметр *Путь* — это абсолютный путь в файловой системе медиасервера, которому принадлежит пул с идентификатором *ID_пула*. При помощи параметра *-d* можно задать описание хранилища.

-г ID

Удалить хранилище с идентификатором *ID* из конфигурации RuBackup.

-с ID -d 'Новое описание'

Изменить описание хранилища.

16. rb_log_viewer

НАЗВАНИЕ

rb_log_viewer - просмотр журналов RuBackup.

СИНТАКСИС

-h
-l
-c
-m
-g
-t
-r
-x

ОПИСАНИЕ

rb_log_viewer - утилита администратора RuBackup для просмотра журнальных сообщений RuBackup.

ПАРАМЕТРЫ

-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.

- г Журнал репозитория.
- х Очистить все журналы RuBackup (кроме системного журнала).

17. rb_media_servers

НАЗВАНИЕ

rb_media_servers - управление медиасerverами RuBackup.

СИНТАКСИС

```
-h  
-u [ -v ]  
-v  
-a Имя_хоста [ -4 ipv4 ] [ -6 ipv6 ] [ -m macaddr ] [ -d  
'Описание' ]  
-r ID  
-x ID  
-t ID  
-c ID -d 'Новое описание'
```

ОПИСАНИЕ

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

Удалить медиасerver 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 медиасerverы в серверную группировку с помощью параметра -a. Достаточно запустить медиасerver с правильными адресами основного и резервного (при наличии) серверов в конфигурационном файле.

При первом запуске медиасerver будет зарегистрирован в системе как неавторизованный медиасerver, при этом процесс медиасerverа будет остановлен. Необходимо авторизовать новый медиасerver с помощью вызова `rb_media_servers -t` и вновь запустить его процесс.

медиасerver должен иметь хотя бы один пул устройств хранения резервных копий (см. **rb_pools**), в противном случае на хосте медиасerverа

будет выводиться соответствующее предупреждение, а медиасервер будет невозможно использовать для хранения резервных копий.

ПАРАМЕТРЫ

-h

Справочное сообщение.

-l [-o]

Список авторизованных медиасерверов. Параметр **-o** выдаёт информацию только о серверах, которые работают в данный момент времени.

-u [-v]

Список неавторизованных медиасерверов. Параметр **-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. Для получения значения идентификатора используйте вызов `gb_media_servers -l`. Если с медиасервером ассоциированы пулы устройств хранения резервных копий, его невозможно удалить из конфигурации.

-x ID

Удалить неавторизованный медиасервер с идентификатором *ID* из списка неавторизованных серверов. Для получения значения идентификатора используйте вызов `gb_media_servers -u`.

-t ID

Авторизовать неавторизованный медиасервер с идентификатором *ID*. Для получения значения идентификатора используйте вызов `gb_media_servers -u`.

-c ID -d 'Новое описание'

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

18. rb_modules

НАЗВАНИЕ

rb_modules - управление модулями RuBackup.

СИНТАКСИС

-h
-l
-i *module.sql*

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-h
Справочное сообщение.

-l
Список установленных модулей RuBackup.

-i *module.sql*
Инициализировать новый модуль.

19. rb_notifications

НАЗВАНИЕ

rb_notifications - контроль работы очереди уведомлений RuBackup.

СИНТАКСИС

-h
-v
-o
-e
-k *ID*

ОПИСАНИЕ

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).

ПАРАМЕТРЫ

-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 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. Для получения значения идентификатора используйте вызов `rb_notifications -l`.

20. rb_pools

НАЗВАНИЕ

rb_pools - управление пулами RuBackup.

СИНТАКСИС

```
-h  
-v  
-a Имя_пула [ -t Тип_пула ] [ -m Имя_хоста ] [ -z Тип_сжатия ] [ -s Сторона_сжатия ] [ -d 'Описание' ]  
-g ID  
-c ID [ -n Имя_пула ] [ -z Тип_сжатия ] [ -s Сторона_сжатия ] [ -d 'Новое описание' ]
```

ОПИСАНИЕ

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 трех 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.

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

ПАРАМЕТРЫ

- h**
Справочное сообщение.
- v**
Выводит список пулов. Предоставляет расширенную информацию о пулах.

-a *Имя_пула* [-t *Тип_пула*] [-m *Имя_хоста*] [-z *Тип_сжатия*] [-s *compression_location*] [-d '*Описание*']

Добавить в конфигурацию RuBackup новый пул. При помощи параметра -t можно определить тип пула. Возможные варианты: 'File system', 'Tape library' и 'Cloud'. При помощи параметра -z можно определить будут ли резервные копии в этом пуле подвергаться сжатию (возможные значения параметра: None, gzip, xz или pigz). При помощи параметра -s можно определить, где будет выполняться процедура сжатия: на сервере или клиенте (возможные значения Server или Client).

-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.

21. rb_remote_replication

НАЗВАНИЕ

rb_remote_replication - управление правилами непрерывной удалённой репликации в RuBackup.

СИНТАКСИС

-h
-m
-l
-s
-n
-a
-X *ID*
-i *Файл_импорта*
-e *ID* [-f *Файл_экспорта*]
-d *ID*
-w *ID*
-r *ID*
-x *ID*
-C *ID* -E *Параметр:Значение,Параметр1:Значение1,...*

ОПИСАНИЕ

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`.

ПАРАМЕТРЫ

- 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*.

-С ID -Е 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.

22. rb_repository

НАЗВАНИЕ

rb_repository — управление репозиторием резервных копий RuBackup.

СИНТАКСИС

-h
-v
-c *ID* -p *ID_пула*
-m *ID* -p *ID_пула*
-r *ID*
-V *ID*

ОПИСАНИЕ

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»).

ПАРАМЕТРЫ

-h
Справочное сообщение.

-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.

-г ID

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

-с ID -р ID_пула

Реплицировать (копировать) резервную копию в другой пул.

-м ID -р ID_пула

Переместить резервную копию в другой пул.

-v ID

Проверить резервную копию. Результат проверки будет отражён как статус резервной копии.

23. rb_tape_cartridges

НАЗВАНИЕ

rb_tape_cartridges - управление картриджами ленточных библиотек в конфигурации RuBackup.

СИНТАКСИС

```
-h  
-l  
-a volume_tag -t Tun -p ID_пула [ -d 'Описание' ]  
-r ID  
-c ID [ -p ID_пула ] [ -d 'Новое описание' ]
```

ОПИСАНИЕ

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**).

ПАРАМЕТРЫ

-h

Справочное сообщение.

-l

Список картриджей в коллекции RuBackup.

-a *volume_tag* -t *Tun* -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.

-s *ID* [-p *ID_пула*] [-d '*Новое описание*']

Изменить пул, к которому принадлежит картридж, или описание картриджа.

24. rb_tape_libraries

НАЗВАНИЕ

rb_tape_libraries - управление ленточными библиотеками в конфигурации RuBackup.

СИНТАКСИС

```
-h  
-l  
-L ID_библиотеки -s  
-L ID_библиотеки -m  
-L ID_библиотеки -d  
-L ID_библиотеки -t  
-M ID_робота -e MAIL_SLOT_ID  
-M ID_робота -S  
-M ID_робота -i MAIL_SLOT_ID -p ID_пула  
-n volume_tag  
-c volume_tag
```

ОПИСАНИЕ

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`).

ПАРАМЕТРЫ

-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 библиотеку 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.

Параметр **-S** (синхронизация) проверяет только расположение картриджей в ленточной библиотеке. Чтобы проверить наличие файловой системы LTFS на картридже используйте параметр **-c**.

-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.

25. rb_task_queue

НАЗВАНИЕ

rb_task_queue - контроль работы главной очереди задач RuBackup.

СИНТАКСИС

-h
-v
-o
-e
-k *ID*

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-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. Для получения значения идентификатора используйте команду `gb_task_queue -l`.

26. rb_tl_task_queue

НАЗВАНИЕ

rb_tl_task_queue - контроль работы очереди ленточных библиотек RuBackup.

СИНТАКСИС

-h
-l
-k *ID*

ОПИСАНИЕ

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 рестартована;
- 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.

ПАРАМЕТРЫ

- 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.

Внимание! 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 потребуются 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.

27. rb_strategies

НАЗВАНИЕ

rb_strategies – контроль работы механизма выполнения одновременных действий над группами ресурсов.

СИНТАКСИС

```
-h  
-v  
-F  
-I  
-D  
-N  
-A  
-L strategy_ID  
-X rule_ID  
-C rule_ID -E  
-W ID  
-R ID  
-i import_file  
-e ID -f  
-d ID  
-a ID -c  
-z ID
```

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

-h

Справочное сообщение.

-v

Выводит список стратегий с их идентификаторами (ID). Параметр -v предоставляет информацию о стратегиях в расширенном формате.

-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.

28. rb_user_groups

НАЗВАНИЕ

rb_user_groups - управление группами пользователей RuBackup.

СИНТАКСИС

```
-h  
-l  
-a Имя_группы [ -d 'Описание' ]  
-г ID  
-с ID [ -п Новое_имя_группы ] [ -d 'Новое описание' ]
```

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

```
-h  
Справочное сообщение.  
-l  
Список пользовательских групп.  
-а 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'.  
-г ID  
Удалить из конфигурации группу с идентификатором ID. Для получения значения идентификатора группы используйте вызов rb_user_groups -l.  
-с ID [ -п Новое_имя_группы ] [ -d 'Новое описание' ]  
Изменить название группы с идентификатором ID и, при необходимости, изменить описание группы.
```

29. rb_users

НАЗВАНИЕ

rb_users - управление пользователями RuBackup.

СИНТАКСИС

```
-h
-l [ -v ]
-a username -e e-mail [ -g ID_группы ] [ -f 'Полное имя' ] [ -p
'адрес': 'тел': 'офис' ]
-g ID
-c ID [ -n username ] [ -e e-mail ] [ -g ID_группы ] [ -f 'Полное
имя' ] [ -p 'адрес': 'тел': 'офис' ]
```

ОПИСАНИЕ

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.

ПАРАМЕТРЫ

```
-h
Справочное сообщение.

-l [ -v ]
Список пользователей. Параметр -v предоставляет расширенную
информацию о пользователях.

-a username -e e-mail [ -g ID_группы ] [ -f 'Полное имя' ] [ -p
'адрес': 'тел': 'офис' ]
```

Добавить в конфигурацию RuBackup нового пользователя с указанным адресом электронной почты. При помощи параметра -g возможно определить, в какую пользовательскую группу будет добавлен пользователь. При помощи параметра -f можно указать полное имя пользователя. При помощи параметра -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'.

-г *ID*

Удалить из конфигурации пользователя с идентификатором *ID*. Для получения значения идентификатора пользователя используйте вызов `gb_users -l`.

-с *ID* [-n *username*] [-e *e-mail*] [-g *ID_группы*] [-f '*Полное имя*'] [-р '*адрес*': '*тел*': '*офис*']

Изменить параметры пользователя с идентификатором *ID*.

30. rb_update

НАЗВАНИЕ

rb_update - утилита сравнения существующей базы данных и sql скрипта создания новой базы данных.

СИНТАКСИС

```
-h  
-l [ -v ]  
-a username -e e-mail [ -g ID_группы ] [ -f 'Полное имя' ] [ -p  
'адрес':'тел':'офис' ]  
-г ID  
-с ID [ -n username ] [ -e e-mail ] [ -g ID_группы ] [ -f 'Полное  
имя' ] [ -p 'адрес':'тел':'офис' ]
```

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-h
Справочное сообщение.

-v
Включение информационного режима. Параметр -v предоставляет дополнительную информацию о выполняемых операциях.

- R**
Принудительное обновление существующей базы данных
- O**
Указать путь к выходному файлу. Использовать '-' для вывода в стандартный поток вывода
- H**
Указать адрес хоста базы данных
- P**
Указать порт соединения базы данных
- D**
Указать имя существующей базы данных
- U**
Указать имя пользователя базы данных
- P**
Указать пароль через аргумент
- W**
Принудительная установка пароля
- I**
Указать путь к каталогу с новыми sql-скриптами.

Утилиты командной строки клиента

RuBackup

1. rb_archives

НАЗВАНИЕ

rb_archives - управление резервными копиями со стороны клиента RuBackup.

СИНТАКСИС

```
-h  
-L [ -v ]  
-l Модуль [ -v ]  
-c Ресурс -m Модуль [ -a Криптоалгоритм ] [ -e Параметр:Значение, Параметр1:Значение1,... ]  
-г ID  
-у ID  
-С ID -р ID_пула  
-М ID -р ID_пула  
-х ID [ -d Каталог ]  
-Х ID [ -d Каталог ]  
-f ID [ -v ]  
-А  
-Р  
-w
```

ОПИСАНИЕ

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. При помощи параметра -р можно впоследствии изменить пароль клиента. Если вы забыли пароль клиента, то для его сброса следует обратиться к администратору системы резервного копирования RuBackup (см. **rb_clients**).

ПАРАМЕТРЫ

```
-h  
Справочное сообщение.
```

-L

Список системных имен всех модулей резервного копирования, которые поддерживаются на данном клиенте. Параметр **-v** предоставляет расширенную информацию о наименовании модулей.

-l Модуль

Список резервных копий клиента для выбранного *модуля* (требуется задать системное имя модуля). Параметр **-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.а.

-с Ресурс -m Модуль [-a Криптоалгоритм] [-е Параметр:Значение, Параметр1:Значение1, ...]

Создать срочную резервную копию определённого *ресурса*. Если ресурс располагается в файловой системе, следует использовать абсолютный путь. Параметр **-m** определяет системное имя *модуля*, для которого необходимо выполнить срочное резервное копирование. Параметр **-a** позволяет выбрать *алгоритм защитного преобразования* резервной копии (см. **rbcrypt** для полного перечня возможных *алгоритмов*). Параметр **-e** позволяет задать значения нескольких параметров сразу в виде *Параметр:Значение*, перечислив их через запятую. Эта команда всегда выполняет полное резервное копирование.

-г 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.

-у ID

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

-x ID [-d Каталог]

Восстановить резервную копию или цепочку резервных копий, если резервная копия с идентификатором *ID* ссылается на другие резервные копии, в текущий каталог или в *Каталог*, определённый параметром **-d**.

-X ID [-d Каталог]

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, если

резервная копия с идентификатором *ID* ссылается на другие резервные копии, в текущий каталог или в *Каталог*, определённый параметром *-d*.

-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.

-P

Вывести список доступных пулов RuBackup.

-w

Изменить пароль клиента.

2. rbcrypt

НАЗВАНИЕ

rbcrypt - защитное преобразование файлов при помощи секретного ключа.

СИНТАКСИС

```
-h  
-l  
-с Входной_файл -к Ключ_файл [ -о Выходной_файл ] [ -а  
Криптоалгоритм ] [ -b Размер_блока ] [ -1 | -2 | -5 | -0 ] [ -г ]  
[ -v ]  
-x Входной_файл.сг -к Ключ_файл [ -о Выходной_файл ] [ -а  
Криптоалгоритм ] [ -b Размер_блока ] [ -1 | -2 | -5 | -0 ] [ -г ]  
[ -v ]
```

ОПИСАНИЕ

rbcrypt - утилита клиента RuBackup для защитного преобразования файлов на стороне клиента RuBackup. Для защитного преобразования используется секретный ключ, задаваемый пользователем. Секретный ключ для защитного преобразования должен находиться в файле. По умолчанию длина ключа составляет 256 бит или 32 байта. Соответственно, файл ключа должен иметь длину 32 байта. Длина ключа должна соответствовать выбранному алгоритму преобразования (см. Приложение 2 «Алгоритмы преобразования в RBCRYPT и RBFD»).

Поскольку секретный ключ может содержать неотображаемые на экране символы, рекомендуется распечатать его при помощи утилиты hexdump, например:

```
$ hexdump master-key  
00000000 79d1 4749 7335 e387 9f74 c67e 55a7 20ff  
00000010 6284 54as 83a3 2053 4818 e183 1528 a343  
00000020
```

Важно! Секретный ключ необходимо хранить в месте, доступном только тем, кто должен иметь возможность произвести обратное преобразование файла. Утеря ключа делает невозможным обратное преобразование файла.

ПАРАМЕТРЫ

```
-h  
Справочное сообщение.
```

-l

Список доступных алгоритмов защитного преобразования.

-с *Входной_файл* **-к** *Ключ_файл* [**-о** *Выходной_файл*] [**-а** *Криптоалгоритм*] [**-b** *Размер_блока*] [**-1** | **-2** | **-5** | **-0**] [**-г**] [**-v**]

Защитное преобразование файла *Входной_файл* ключом, который содержится в файле *Ключ_файл*. Параметр **-о** позволяет задать имя файла, в который будут записаны преобразованные данные (по умолчанию имя файла будет иметь вид *Входной_файл.cr*). Дополнительные параметры позволяют выбрать алгоритм преобразования, размер блока данных и длину ключа.

-x *Входной_файл.cr* **-к** *Ключ_файл* [**-о** *Выходной_файл*] [**-а** *Криптоалгоритм*] [**-b** *Размер_блока*] [**-1** | **-2** | **-5** | **-0**] [**-г**] [**-v**]

Обратное преобразование зашифрованного файла *Входной_файл.cr* ключом, который содержится в файле *Ключ_файл*. Параметр **-о** позволяет задать имя файла, в который будут записаны преобразованные данные (по умолчанию имя файла будет иметь вид *Входной_файл* без расширения *cr*). Дополнительные параметры позволяют выбрать алгоритм преобразования, размер блока данных и длину ключа.

[**-а** *Криптоалгоритм*]

Задать алгоритм преобразования. Возможны следующие варианты: anubis, aria, cast6, camellia, kalyna, kuznyechik (или GOST_R_34_12_2015), mars, rijndael (или AES), serpent, simon, sm4, speck, threefish, twofish. Значение по умолчанию kuznyechik. Дополнительную информацию см. Приложение 2 «Алгоритмы преобразования, реализованные в RBCRYPT и RBFD».

[**-b** *Размер_блока*]

Задать размер блока данных (байт). Размер по умолчанию 16384 байта, максимальный размер блока данных 1048576 байт.

[**-1** | **-2** | **-5** | **-0**]

Указать длину ключа (бит). Возможные варианты 128, 256, 512 или 1024 бит (по умолчанию 256 бит). Если алгоритм не поддерживает выбранную длину ключа, то будет использован ключ, максимально длинный из поддерживаемых. Дополнительную информацию см. Приложение 2 «Алгоритмы преобразования, реализованные в RBCRYPT».

[**-г**]

Удалить входящий файл.

[**-v**]

Вывести расширенную информацию о работе утилиты.

3. `rb_schedule`

НАЗВАНИЕ

`rb_schedule` - список правил глобального расписания клиента RuBackup.

СИНТАКСИС

-h

-v

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-h

Справочное сообщение.

-v

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

4. **rb_tasks**

НАЗВАНИЕ

rb_tasks - список актуальных задач клиента RuBackup.

СИНТАКСИС

-h
-v

ОПИСАНИЕ

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

ПАРАМЕТРЫ

-h
Справочное сообщение.

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

Приложения

1 Алгоритмы хеш-функций, реализованные в RBD и RBFD

ОПИСАНИЕ

В таблице 1 представлена информация об алгоритмах хеш-функций, которые реализованы в утилитах командной строки **rbd** и **rbfd**.

Таблица 1.

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 введён в действие 1 января 2013 года.
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,	BLAKE2 — криптографическая хеш-функция,

	512	<p>улучшенная версия BLAKE — одного из пяти финалистов конкурса на хеш-функцию SHA-3 (главным образом улучшено быстродействие), представлена 21 декабря 2012 года.</p> <p>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.</p>
--	-----	---

2 Алгоритмы преобразования в RBCRYPT и RBFD

ОПИСАНИЕ

В таблице 2 представлена информация об алгоритмах преобразования, которые реализованы в утилите командной строки **rbcrypt** и **rbfd**.

Таблица 2

Алгоритм	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	

3 Авторские права

ОБЩИЕ ПОЛОЖЕНИЯ

© 2019-2021, ООО «РУБЭКАП». Все права защищены

Ни одна часть этого документа не может быть воспроизведена или передана каким-либо образом, электронным, механическим, методом фотокопирования, записи или как-то ещё без письменного разрешения ООО «РУБЭКАП».

RBCRYPT

Утилита RBCRYPT разработана Andrey Kuznetsov © 2019-2021 с использованием библиотеки `srpcrypto` (<http://cppcrypto.sourceforge.net/>).

Copyright (c) 2015-2016, kerukuro. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012, Intel Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of the Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2014, Project Nayuki.

(MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

- The Software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the Software or the use or other dealings in the Software.

Copyright (c) 2013, Alexey Degtyarev. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2007 Robert W. Waite <winstonwaite@gmail.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2009 Colin Percival, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file was originally written by Colin Percival as part of the Tarsnap online backup system.

In addition to these licenses, cppcrypto contains public-domain or non-copyrighted code written by: Wei Dai, Jean-Philippe Aumasson, Samuel Neves, Shawn Kirst, Peter Schwabe, Günther A. Roland, Martin Schläffer, Krystian Matusiewicz, Daniel J. Bernstein, vampire77, Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, Hongjun Wu, Maxim Locktyukhin, Ronen Zohar, Romain Dolbeau, Andrew Moon, Vladimir Sedach.