

Backup Manager

Пакет консольных программ для резервного копирования.

Написан на [Bash](#) и [Perl](#), может работать с архивами форматов tar, tar.gz, tar.bz2 и zip.

Можно запускать нескольких копий программы с различающимися настройками.

Поддерживается создание дампов [PostgreSQL](#), [MySQL](#), [Subversion](#) и [инкрементное резервное копирование](#).

Резервные копии могут храниться в [течение заданного количества дней](#), а по протоколам [FTP](#), [SSH](#) или [Rsync](#) возможна их выгрузка на удалённые хосты, также поддерживается выгрузка на [Amazon S3](#) и запись резервных копий на устройства [CD/DVD](#).

Backup Manager создаёт [архив](#) путём использования [метода резервного копирования](#) в [хранилище](#).

Пакет состоит из следующих программ:

- [backup-manager](#)
- [backup-manager-purge](#)
- [backup-manager-upload](#)

Код проекта доступен по ссылкам:

-  [sukria/Backup-Manager](#)
- [Пакеты для Debian](#)

Основой для написания раздела [Настройка](#) является перевод материала: [Backup Manager 0.7.7 User Guide © Alexis Sukrieh, ver. 1.7 - 14 Apr, 2008](#) из пакета [backup-manager-doc \(0.7.12-4\)](#).

Примеры использования

[Примеры использования Backup Manager](#)

Перевод на русский язык

[LC_MESSAGES](#) (файлы .po и .mo)¹⁾

Путь: /usr/share/locale/ru/LC_MESSAGES/

<https://github.com/sukria/Backup-Manager/blob/master/po/ru.po>

Установка

```
apt-get install backup-manager backup-manager-doc
```

 В Debian 8 нужно использовать версию [Backup-Manager 0.7.12](#)

Настройка

Актуально для версии 0.7.12.

Backup Manager настраивается через конфигурационные файлы. Вы можете запускать Backup Manager с разными конфигурационными файлами, подробнее см. [раздел о параметрах командной строки](#).

Файл с настройками по умолчанию: `/etc/backup-manager.conf`

Хранилище

Хранилище это место в файловой системе где хранятся все архивы. Это особое место для Backup Manager, оно может быть очищено в ходе выполнения резервного копирования: архивы, возраст которых превышает заданное время существования будут удалены. Если хранилище не существует, оно будет создано во время работы программы.

Изолированное хранилище на выделенном разделе это хорошая идея. Это сможет предотвратить поглощение хранилищем всего дискового пространства раздела. Плохой конфигурационный файл может привести к созданию архивов огромного размера, поэтому будьте аккуратнее.

BM_REPOSITORY_ROOT

Тип: строка, значение по умолчанию: `/var/archives`.

Абсолютный путь к папке, которая будет корнем хранилища.

Пример:

```
export BM_REPOSITORY_ROOT="/var/archives"
```

BM_REPOSITORY_SECURE

Тип: логическое, значение по умолчанию: `true`.

В целях безопасности, доступ к хранилищу может быть разрешён для определённой пары пользователь/группа. Это позволит предотвратить доступ к архивам для чтения (и записи) любым пользователям в системе. Этот режим включен по умолчанию (значение `root:root`).

Для включения данного режима нужно установить значение `BM_REPOSITORY_SECURE` в `yes`, и изменить значения `BM_REPOSITORY_USER` и `BM_REPOSITORY_GROUP` если это необходимо.

Вы также можете изменить права доступа к хранилищу и архивам также используя переменные: `BM_REPOSITORY_CHMOD` и `BM_ARCHIVE_CHMOD`.

Пример:

```
export BM_REPOSITORY_SECURE="true"
export BM_REPOSITORY_USER="root"
export BM_REPOSITORY_GROUP="root"
export BM_REPOSITORY_CHMOD="770"
export BM_ARCHIVE_CHMOD="660"
```

Шифрование

Если вы не доверяете условиям в месте хранения своих архивов, вы можете шифровать их, чтобы только вы были единственным, кто может прочитать их содержимое. Будет хорошо, если вы будете выгружать архивы в удалённые хранилища или использовать для повседневного хранения съёмные носители.

BM_ENCRYPTION_METHOD

Тип: строка, значение по умолчанию: `undefined`.

До Backup Manager шифрование определяется в одном месте в файле конфигурации. Если переменная `BM_ENCRYPTION_METHOD` не определена, то шифрование во время создания архива не происходит, если метод

шифрования определён там, то любой архив будет зашифрован через канал(pipe) этого метода шифрования.

Обратите внимание, что шифрование поддерживается для методов: `mysql`, `pipe`, `tarball` и `tarball-incremental`, но только для следующих типов файлов: `tar`, `tar.gz`, `tar.bz2`.

Для шифрования архивов поддерживается только метод `gpg`.

Backup Manager шифрует архив через канал(pipe) по порядку, чтобы не сохранять не зашифрованные данные на физических носителях. Шифрование будет выполняться с помощью командной строки, как показано в примере:

```
<command> | gpg -r "$BM_ENCRYPTION_RECIPIENT" -e > archive.gpg
```

Для расшифровки архива, зашифрованного с помощью [GPG](#), вы должны быть владельцем закрытого ключа GPG, которым было осуществлено шифрование. Для этого выполните следующее:

```
$ gpg -d <archive.gpg> > archive
```

GPG выдаст приглашение ввода пароля закрытого ключа и содержимое архива будет расшифровано, если пароль является действительным.

Для получения более подробной информации о шифровании обратитесь к документации GPG.

BM_ENCRYPTION_RECIPIENT

Тип: строка, значение по умолчанию: undefined.

Как было описано в предыдущем разделе, переменная должна содержать идентификатор получателя [GPG](#) шифрования, например: ваш GPG ID.

Примеры использования GPG ID:

```
export BM_ENCRYPTION_RECIPIENT="0x1EE5DD34"  
export BM_ENCRYPTION_RECIPIENT="Alexis Sukrieh"  
export BM_ENCRYPTION_RECIPIENT="sukria@sukria.net"
```

Архивы

Архивы создаются с помощью методов резервного копирования, они могут быть иметь практически любое имя, но формат имени будет следующей: `prefix-name-date.filetype`. Архив представляет собой файл, который содержит данные, он может быть сжат или нет, в двоичном виде или нет.

BM_ARCHIVE_STRICTPURGE

Тип: логическое, значение по умолчанию: true.

Как описано в разделе [BM_REPOSITORY_ROOT](#), каждый архив сделанный с помощью Backup Manager будут очищен, когда его срок годности истекает. В версиях до 0.7.6, любой архив, мог быть очищен.

Теперь Вы можете выбрать, чтобы очистить только архив, созданные в соответствии с настройками из файла конфигурации, а именно: архивы с префиксом [BM_ARCHIVE_PREFIX](#).

Это полезно, если вы разделяете один [BM_REPOSITORY_ROOT](#) с разными экземплярами Backup Manager, которые имеют различные правила очистки архивов (например: [BM_REPOSITORY_ROOT](#) распространяемый по [NFS](#) для работы с несколькими экземплярами Backup Manager).

Пример:

```
export BM_ARCHIVE_STRICTPURGE="true"
```

BM_ARCHIVE_NICE_LEVEL

Тип: строка, значение умолчанию: 10.

Backup Manager использует разные методы архивирования, которые могут использовать большое количество ресурсов (в основном CPU); Это может быть приемлемым, если Backup Manager запускается в ночное время на постоянно работающем сервере, но это может серьёзно замедлить работу персонального компьютера (ПК). Обычно пользователи ПК используют [AnaCron](#) для запуска Backup Manager когда это возможно, т.к. большую часть времени ПК действительно используется.

Опытные пользователи могут задать приоритет выполнения задания для создания архива при помощи этой переменной в файле конфигурации.

Для установки приоритета используйте следующие значения: -20 (наивысший приоритет) до 19 (низший приоритет), (включая 0 – как нулевой приоритет). Для получения дополнительной информации смотри документацию по nice.

Пример:

```
export BM_ARCHIVE_NICE_LEVEL="19" # рекомендовано для пользователей ПК.
```

BM_ARCHIVE_PURGEDUPS

Тип: логическое, значение умолчанию: true.

В стратегии резервного копирования важным вопросом является оптимальное использование диска, вам можете быть полезным, использовать возможность Backup Manager по удалению дубликатов файлов. Когда архив создаётся, Backup Manager просматривает предыдущие версии этого архива. Если он находит что предыдущий архив - это тот же файл, как и только что созданный, предыдущий файл заменяется символической ссылкой на новый. Это полезно если Вы не хотите иметь две копии одного архив в хранилище.

Пример:

```
export BM_ARCHIVE_PURGEDUPS="true"
```

```
host-etc.20051115.tar.gz  
host-etc.20051116.tar.gz -> /var/archives/host-etc.20051117.tar.gz  
host-etc.20051117.tar.gz
```

BM_ARCHIVE_TTL

Тип: целое число, значение умолчанию: 5.

Одной из основных концепций, определяющих работу с хранилищем является удаление устаревших архивов автоматически. Удаление старых архивов всегда выполняется при запуске Backup Manager. Во время этой фазы, все архивы старше определённого срока жизни (в днях) удаляются.

Начиная с версии 0.7.3, Backup Manager удаляет только файлы, которые создал именно он в то время как в предыдущих версиях, он удалял и другие файлы в хранилище.

Обратите внимание, что при использовании инкрементального метода для построения архивов, Backup Manager будет по разному обрабатывать полные архивы (master) и добавочные (инкрементальные). Инкрементные резервные копии будут удаляться, как и любые другие архивы (при превышении срока жизни). С другой стороны, устаревшие полные архивы (master) не будут удаляться, если есть более новые полные архивы (master) в хранилище. Тогда, даже со сроком жизни более трёх дней, полный архив (master) будет существовать более трёх дней, пока новый полный архив (master) не будет создан.

Пример:

```
export BM_ARCHIVE_TTL="5"
```

BM_REPOSITORY_RECURSIVEPURGE

Тип: логическое, значение умолчанию: false.

В большинстве случаев, все архивы хранятся в папке верхнего уровня, путь к который задан в переменной [BM_REPOSITORY_ROOT](#). Но иногда может возникнуть необходимость создания вложенных папок например для хранения архивов загруженных с других компьютеров с работающим Backup Manager. В этом случае можно настроить Backup Manager, чтобы он очищал и эти папки, установив значение `BM_REPOSITORY_RECURSIVEPURGE` равным `true`.

Обратите внимание, что переменная `BM_ARCHIVE_TTL` является глобальной, поэтому если вы хотите иметь разный срок жизни разных архивов, это не тот путь. В этом случае Вам необходимо сохранить их вне [BM_REPOSITORY_ROOT](#) и выполнять чистку запуская `backup-manager -purge` с другим конфигурационным файлом.

Пример:

```
export BM_REPOSITORY_RECURSIVEPURGE="false"
```

BM_ARCHIVE_PREFIX

Тип: строка, значение умолчанию: `$HOSTNAME`.

Это префикс имени архива.

Пример:

```
export BM_ARCHIVE_PREFIX="$HOSTNAME"
```

```
# echo $HOSTNAME
ouranos
# ls /var/archives
ouranos-20051123.md5
ouranos-usr-local-src.20051123.tar.gz
ouranos-etc.20051123.tar.gz
```

Методы резервного копирования

Основной задачей Backup Manager является создание архивов, для этого используются методы. Каждый метод может потребовать набор параметров. Здесь мы опишем каждый метод который поддерживается в версии 0.7.7.

Доступные методы:

- `tarball`
- `tarball-incremental`
- `mysql`
- `svn`
- `pipe`
- `pgsql`
- `none`

Выбираемые методы должны быть определены в переменной `BM_ARCHIVE_METHOD`. Вы можете задать несколько различных методов, которые вы хотите использовать. Помните о том, что для каждого выбранного метода нужно будет задать значения соответствующих ему переменных. Обратите внимание, что вы также можете выбрать ни один из предложенных методов, если вы не хотите создавать архивы, поставьте значение `none`.

Пример:

```
export BM_ARCHIVE_METHOD="tarball-incremental mysql"
```

Tag-архивы

Имя метода: *tarball*, префикс переменной: `BM_TARBALL`.

Если все, что вам нужно, это создать несколько Tag-архивов, вы можете использовать этот метод. Этот метод обрабатывает список папок и создаёт соответствующие архивы. Этот метод является самым простым в использовании, он создаёт Tag-архивы так же как и большинство ваших собственных скриптов. Основным недостатком этого метода заключается в том, что используется много дискового пространства: создавая каждый день архивы могут быть большими, даже если нет никаких изменений в их содержимом.

В разделе [Инкрементальные Tag-архивы](#) описан метод, который позволяет оптимизировать размер архива.

При создании полных резервных копий (если не заданы инкрементных), Backup Manager обязательно добавляет ключевое слово `master` к имени архива. Это очень полезно при использовании метода [tarball-incremental](#).

Можно задать следующие параметры: формат имени архива, метод сжатия (`gzip`, `zip`, [bzip2](#), без сжатия) и возможность разыменовывать символические ссылки при создании tar-архива.

BM_TARBALL_NAMEFORMAT

Эта переменная определяет, формат имени архива. Два возможных значения:

- `long`: абсолютный путь к папке (например: `var-log-apache` для `/var/log/apache`).
- `short`: только имя папки (например: `apache` для `/var/log/apache`).

Рекомендованное значение (по умолчанию): `long`.

BM_TARBALL_FILETYPE

Тип: перечисление (`tar`, `tar.gz`, `tar.bz2`, `tar.lz`, `zip`, `dar`), значение по умолчанию: `tar.gz`.

Эта переменная определяет тип файла создаваемого архива. В определённом смысле, это определяет, какую программу использовать (`zip`, `gzip`, `dar` or [bzip2](#)). Поддерживаются следующие значения: `tar`, `tar.gz`, `tar.bz2`, `zip` и `dar`. Обратите внимание, что в зависимости от выбранного значения вы должны убедиться, что у вас установлен соответствующий архиватор.

Для лучшей степени сжатия используйте `tar.bz2` или `tar.lz`.

Начиная с версии 0.7.1, Backup Manager поддерживает `dar` архивы. Это архиватор обеспечивает некоторые интересные функции, такие как деление архива на части.

Начиная с версии 0.7.5, Backup Manager поддерживает `lzma` архивы.

Убедитесь, что установлен соответствующий архиватор:

- `tar.bz2` : нужен [bzip2](#).
- `tar.lz` : нужен `lzma`.
- `dar` : нужен `dar`.
- `zip` : нужен `zip`.

BM_TARBALL_SLICESIZE

Тип: строка.

Если вы хотите, чтобы размер части архива не превышал определённый размер (например 2 Гбайт), то тогда задайте значение в переменной `BM_TARBALL_SLICESIZE`. Этот параметр поддерживает только `dar`

Если вы хотите установить размер части архива 2 Гбайт, задайте следующее значение:

```
BM_TARBALL_SLICESIZE="1000M"
```

Для дополнительной информации читайте документацию на `dar`

BM_TARBALL_EXTRA_OPTIONS

Тип: строка.

Если вы хотите задать дополнительные параметры для `tar` или `dar`, задайте их в этой переменной. Оставьте значение переменной пустым, если дополнительные параметры вам не нужны.

Пример: включение полного вывода `tar`:

```
BM_TARBALL_EXTRA_OPTIONS="-v"
```

BM_TARBALL_DUMPSYMLINKS

Тип: логическое, значение по умолчанию: `true`.

Это можно использовать для разыменования символических ссылок при создании `tar`-архива (или файла `zip`). Если вы включите эту функцию, каждая символическая ссылка в файловой системе будет заменена в архиве на файл/папку на которую она указывает. Используйте эту функцию с осторожностью, это может привести к созданию огромных архивов, или даже хуже: если у вас где-то есть циклическая символическая, то это приведёт к созданию бесконечного архива!

В большинстве случаев, вы не должны использовать эту функцию.

BM_TARBALL_DIRECTORIES

Тип: список (разделитель пробел), значение по умолчанию: `/etc /home`.

Список путей (источников) для создания архива.

Начиная с версии 0.7.3, эта переменная заменяет массив [BM_TARBALL_TARGETS](#), он по-прежнему поддерживается для обратной совместимости. Вы можете использовать эту переменную, чтобы задать список путей источников для создания архива, но вы не должны использовать эту переменную, если один или более из путей, которые нужно включить в архив содержит пробелы.

Если путь содержит пробелы, например `Program Files`, вы не должны использовать эту переменную, вместе неё нужно использовать массив [BM_TARBALL_TARGETS](#).

Пример:

```
export BM_TARBALL_DIRECTORIES="/etc /home /var/log/apache"
```

Вы также можете использовать шаблоны (Bash patterns) в `BM_TARBALL_DIRECTORIES`.

Пример получения списка вложенных папок для `/home`:

```
BM_TARBALL_DIRECTORIES="/home/*"
```

BM_TARBALL_TARGETS

Тип: массив, значение по умолчанию: `/etc`.

Список путей (источников) для создания архива (начиная с версии 0.7.3 рекомендуется использовать [BM_TARBALL_DIRECTORIES](#)).

Если путь содержит пробелы, (например Program Files) вы не сможете использовать `BM_TARBALL_DIRECTORIES`.

Пример:

```
BM_TARBALL_TARGETS[0]="/etc"
BM_TARBALL_TARGETS[1]="/home/*"
BM_TARBALL_TARGETS[2]="/boot"
BM_TARBALL_TARGETS[3]="/mnt/win/Program Files"
```

Вы также можете использовать шаблоны (Bash patterns) в `BM_TARBALL_TARGETS[]`.

Пример получения списка вложенных папок для /home:

```
BM_TARBALL_TARGETS[0]="/home/*"
```

BM_TARBALL_BLACKLIST

Тип: список (разделитель пробел), значение по умолчанию: `/proc /dev /sys /tmp`.

«Черный список» полезно использовать для исключения папок и/или файлов из списка файлов, которые включаются в архив. Это действительно полезно, когда вы используете шаблоны в `BM_TARBALL_DIRECTORIES`. На самом деле, вы можете сделать резервную копию всех папок верхнего уровня файловой системы (/*) но без постоянно изменяющихся мест, таких как `/tmp`, `/dev` и `/proc`.

Вы также можете использовать эту переменную для исключения файлов по расширению, таких как, например `mp3` или `mpg`.

Пример:

```
export BM_TARBALL_BLACKLIST="/tmp /dev /proc *.mp3 *.mpg"
```

BM_TARBALL_OVER_SSH

Тип: логическое, значение по умолчанию: `false`.

Зависит от: BM_UPLOAD_SSH

Если вы хотите создавать архивы на других серверах можно использовать метод создания архивов по SSH. Это можно использовать, например для централизованного хранения архивов на отдельном сервере. Backup Manager будет создавать архивы по SSH и сохранять получившиеся архивы локально, как и другие архивы. Имя архива будет начинаться с имени удаленного сервера вместо `BM_ARCHIVE_PREFIX`.

Для того, чтобы использовать эту функцию нужно, чтобы установлены следующие переменные с префиксом `BM_UPLOAD_SSH`:

- `BM_UPLOAD_SSH_USER`: имя пользователя для подключения к удаленному серверу. Обратите внимание, что этот пользователь будет запускать `tar` удаленно, поэтому позаботьтесь, чтобы он имел право на чтение архива!
- `BM_UPLOAD_SSH_KEY`: файл ключа для установления соединения.
- `BM_UPLOAD_SSH_HOSTS`: Список хостов с которыми нужно работать по SSH.

Если вы включите эту функцию, обратите внимание, что в результате конфигурационный файл будет иметь следующие ограничения:

- Удалённое создание архивов будет работать только при использовании метода `tarball`, при использовании метода `tarball-incremental` будет то же самое.
- Вы можете использовать один файл с настройками для локального и удалённого сохранения архивов. Если хотите то можете использовать два отдельных файла.

Пример: Имеются три хоста: `host01`, `host02` и `host03`. Сервер хранения это `host01`, у него большой раздел `/var/archives`. Будут архивироваться `/etc`, `/home` и `/var/log` по SSH на `box02` и `box03`, а также сохраняться на `host01`.

```
[...]
export BM_ARCHIVE_METHOD="tarball"

export BM_TARBALL_OVERSSH="true"
export BM_TARBALL_FILETYPE="tar.bz2"
export BM_TARBALL_DIRECTORIES="/etc /home /var/log"

export BM_UPLOAD_SSH_USER="bamuser"
export BM_UPLOAD_SSH_KEY="/home/bamuser/.ssh/iddsa"
export BM_UPLOAD_SSH_HOSTS="box02 box03"
```

Конечно, для того, чтобы это работало правильно, пользователь `bamuser` должен присутствовать на `box02` и `box03`. Также ему должно быть разрешено подключаться к ним с ключом по SSH и у него должны быть права для чтения архивируемых папок.

Инкрементальные Tar-архивы

Метод: *tarball-incremental*, префикс переменной: `BM_TARBALLINC`.

Если вы хотите работать с архивами, не трата дисковое пространство, вы должны использовать этот метод. Концепция этого метода проста: Вы выбираете частоту создания полного архива, все архивы которые будут созданы между двумя полными копиями будут содержать только те файлы, которые изменились со времени создания предыдущим архива.

Например, предположим, что вы хотите сделать резервную копию папки `/home` с помощью этого метода. Ваша папка `/home` содержит две папки: `/home/foo` и `/home/bar`. Вы выбираете еженедельную частоту и определяете, что днём полной копии будет понедельник. Очевидно, что вы будете иметь полный архив с `/home` в понедельник. Затем, если файл изменён внутри `/home/foo` и если `/home/bar` остаётся неизменным, архив вторника будет содержать только изменённые файлы `/home/foo`. Использование этого метода позволит сэкономить много дисковое пространство.

Для создания инкрементных архивов Backup Manager вызывает `tar` с параметром `–listed-incremental`. При этом для каждого архива создаётся специальный файл который содержит статистические данные о архивируемых файлах и папках, это нужно чтобы при следующем архивировании знать изменился этот файл или нет. Когда Backup Manager запускается в первый раз, этот файл не существует, поэтому впервые создаваемые архивы всегда будут создавать полную копию данных. Если файлы которые были ранее заархивированы будут удалены то в следующих инкрементальных копиях этих файлов тоже не будет.

Начиная с версии 0.7.3, это можно сразу заметить, если резервная копия является мастером или это первый инкрементный архив: Backup Manager добавляет в имя архива ключевое слово `master` после дате архива. При очистке хранилища, полные архивы (`master`) не удаляются, как инкрементных. Backup Manager всегда оставляет полный архив (`master`), которое старше, чем инкрементные(добавочные) архивы.

Этот метод использует все переменные для Tar-архивов и добавляет ещё две. Одна, чтобы определить частоту, а другая, чтобы выбрать день для создания полной резервной копии.

BM_TARBALLINC_MASTERDATETYPE

Тип: перечисление(*weekly*, *monthly*), значение по умолчанию: *weekly*.

Период создания архива:

- *weekly* (еженедельно), день недели от 0 до 6;
- *monthly* (ежемесячно), день месяца в диапазоне от 1 до 31.

BM_TARBALLINC_MASTERDATEVALUE

Тип: целое число, значение по умолчанию: 1.

Количество дней при создании полных резервных копий. Обратите внимание, что его значение напрямую зависит от

BM_TARBALLINC_MASTERDATETYPE . Например, 1 означает «понедельник», если вы выбираете weekly (еженедельно), или это означает, «первый день месяца», если вы выбираете monthly ежемесячно.

MySQL

Метод: *mysql*, префикс переменных: BM_MYSQL

Этот метод дает возможность архивировать базы данных [MySQL](#), архивы создаются при помощи с Mysqldump (текстовые файлы SQL) и поэтому могут быть хорошо сжаты.

До версии 0.7.6, Backup Manager использовал метод передачи пароля клиенту MySQL через командную строку. Как поясняет документация MySQL, такой метод является не безопасным, т.к. пароль в открытом виде доступен в течение короткого промежутка времени в файловой системе /proc (или с помощью команды ps).

Чтобы устранить эту уязвимость, клиент пароля MySQL больше не передаётся через командную строку, а сохраняется в файле, расположенном в домашней папке пользователя, запустившего Backup Manager: ~/.my.cnf.

Если этот файл не существует, то Backup Manager создаст его и сохранит в него пароль из переменной [BM_MYSQL_ADMINPASS](#).

BM_MYSQL_DATABASES

Тип: список (разделитель пробел), значение по умолчанию: ALL.

Это список баз данных, которые нужно заархивировать. Вы можете использовать ключевое слово ALL, чтобы сделать резервную копию всех базы данных без необходимости их перечислять.

Пример:

```
export BM_MYSQL_DATABASES="mysql mybase wordpress dotclear phpbb2"
```

BM_MYSQL_SAFEDUMPS

Тип: логическое, значение по умолчанию: true.

Лучший способ для создания дампов MySQL это использовать Mysqldump с ключем --opt. В этой переменной можно задать эти параметры.

Пример:

```
export BM_MYSQL_SAFEDUMPS="true"
```

BM_MYSQL_ADMINLOGIN

Тип: строка, значение по умолчанию: root.

Имя пользователя для подключения к серверу баз данных MySQL. Убедитесь, у этого пользователя есть права на чтение всех баз данных, которые указаны в переменной BM_MYSQL_DATABASES.

Пример:

```
export BM_MYSQL_ADMINLOGIN="root"
```

BM_MYSQL_ADMINPASS

Тип: строка, значение по умолчанию: undefined.

Пароль для доступа к серверу MySQL.

Если вы уже создали файл `~/ .my.cnf` то можно не задавать значение для этой переменной.

Если вы не знаете, что такое `~/ .my.cnf`, тогда установите пароль, а затем Backup Manager сам создаст этот файл.

Пример:

```
export BM_MYSQL_ADMINPASS="MySecretPass"
```

BM_MYSQL_HOST

Тип: строка, значение по умолчанию: localhost.

Имя хоста с сервером MySQL.

Пример:

```
export BM_MYSQL_HOST="localhost"
```

BM_MYSQL_PORT

Тип: строка, значение по умолчанию: 3306.

Порт на котором работает сервер MySQL, определённый в переменной `BM_MYSQL_HOST`.

Пример:

```
export BM_MYSQL_PORT="3306"
```

BM_MYSQL_FILETYPE

Тип: перечисление(`gzip`, `bzip2`), значение по умолчанию: `bzip2`.

Дамп который создаёт Mysqldump это текстовый файл, который может быть сжат. Если вы хотите сжимать файлы, выберите архиватор. Оставьте эту переменную пустой, если вы хотите иметь не сжатые дампы.

Пример:

```
export BM_MYSQL_FILETYPE="bzip2"
```

PostgreSQL

Метод: `pgsql`, префикс переменных: `BM_PGSQL`.

Этот метод позволяет архивировать базы данных [PostgreSQL](#).

BM_PGSQL_DATABASES

Тип: список (разделитель пробел), значение по умолчанию: `ALL`.

Это список баз данных, которые нужно заархивировать. Вы можете использовать ключевое слово `ALL`, чтобы сделать резервную копию всех базы данных без необходимости их перечислять, все БД записываются в один архив.

Пример:

```
export BM_PGSQL_DATABASES="__ALL__"
```

BM_PGSQL_ADMINLOGIN

Тип: строка, значение по умолчанию: root.

Имя пользователя для подключения к серверу баз данных PostgreSQL. Убедитесь, у этого пользователя есть права на чтение всех баз данных, которые указаны в переменной [BM_PGSQL_DATABASES](#)

Пример:

```
export BM_PGSQL_ADMINLOGIN="root"
```

BM_PGSQL_ADMINPASS

Тип: строка.

Пароль для доступа к серверу PostgreSQL.

Пример:

```
export BM_PGSQL_ADMINPASS=""
```

BM_PGSQL_HOST

Тип: строка, значение по умолчанию: localhost.

Имя хоста с сервером PostgreSQL.

Пример:

```
export BM_PGSQL_HOST="localhost"
```

BM_PGSQL_PORT

Тип: строка, значение по умолчанию: 5432.

Порт на котором работает сервер PostgreSQL, определённый в переменной [BM_PGSQL_HOST](#).

Пример:

```
export BM_PGSQL_PORT="5432"
```

BM_PGSQL_FILETYPE

Тип: перечисление(*gzip*, *bzip2*), значение по умолчанию: *bzip2*.

Если вы хотите сжимать файлы, выберите архиватор. Оставьте эту переменную пустой, если вы хотите иметь не сжатые дампы.

Пример:

```
export BM_PGSQL_FILETYPE="bzip2"
```

BM_PGSQL_EXTRA_OPTION

Тип: строка.

Если вы хотите задать дополнительные параметры для pg_dump задайте их в этой переменной. Оставьте значение переменной пустым, если дополнительные параметры вам не нужны.

Пример:

```
export BM_PGSQL_EXTRA_OPTIONS=""
```

Subversion

С помощью этого метода вы можете делать резервные копии хранилищ Subversion. Архив будет создан с помощью svnadmin и будет содержать данные XML (текстовые файлы). Как и метод при работе с MySQL, вы можете сжимать эти файлы.

BM_SVN_REPOSITORIES

Тип: список (разделитель пробел)

Список абсолютных путей к хранилищам SVN для архивирования.

Пример:

```
export BM_SVN_REPOSITORIES="/srv/svnroot/repo1 /srv/svnroot/repo2"
```

BM_SVN_COMPRESSWITH

Тип: перечисление(gzip, bzip2), значение по умолчанию: bzip2.

Если вы хотите сжимать XML файлы, выберите архиватор. Оставьте эту переменную пустой, если вы хотите иметь не сжатые файлы.

Пример:

```
export BM_SVN_COMPRESSWITH="gzip"
```

Pipe

Даже если большинство задач решаются стандартными методами, всегда может возникнуть ситуация для которой нет стандартного решения. Для подобных задач Backup Manager предоставляет возможность произвольного (нестандартного) варианта резервного копирования.

Этот метод называется pipe, он более сложен в использовании, но может использоваться для практически любых задач резервного копирования. Настройка этого метода проста и определяется следующими параметрами:

- Имя (архива)
- Команда (работает с stdout)
- Тип файла (txt, sql, dump, ...)
- Архиватор (gzip, bzip2)

Эти параметры являются массивами, так что вы можете создать столько методов, сколько вам нужно.

Для каждого pipe метода, Backup Manager выполнит переданные команды, и перенаправить содержимое, отправленное на стандартный вывод этой команды в файл с именем с именем метода и его типом файла. Если метод использует архиватор, файл будет сжат.

Пример

Пример архивирования удалённой БД MySQL по SSH:

```
BM_PIPE_COMMAND[0]="ssh host -c \"mysqldump -ufoo -pbar base\""  
BM_PIPE_NAME[0]="base"  
BM_PIPE_FILETYPE[0]="sql"  
BM_PIPE_COMPRESS[0]="gzip"
```

А это второй набор параметров, например, для создания Tar-архива по SSH:

```
BM_PIPE_COMMAND[1]="ssh host -c \"tar -c -z /home/user\""  
BM_PIPE_NAME[1]="host.home.user"  
BM_PIPE_FILETYPE[1]="tar.gz"  
BM_PIPE_COMPRESS[1]=""
```

Обратите внимание, что для каждого следующего набора параметров используется следующий индекс массива.

Методы выгрузки

Одним из самых важных условий при выполнении резервного копирования является хранение исходных данных и резервных копий в разных местах. Чем больше разных физических хранилищ у вас есть, тем лучше. В Backup Manager для этого существуют методы выгрузки.

Существуют различные методы выгрузки, каждый из них имеет свои особенности. В Backup Manager 0.7.7 можно использовать:

- [FTP](#)
- [SSH](#)
- [SSH GPG](#)
- [Rsync](#)
- [Amazon S3](#)

Таким же образом, как и методы резервного копирования, вы можете использовать как любые доступные методы выгрузки по вашему усмотрению.

Список методов выгрузки задаётся в переменной `BM_UPLOAD_METHOD`.

Обратите внимание, что [FTP](#), [SSH](#) и [Amazon S3](#) предназначены для выгрузки архивов, которые созданы с при использовании по меньшей мере, одного метода резервного копирования.

Метод [Rsync](#) выгружает данные в удалённое хранилище или любое другое место вашей файловой системы.

Глобальные переменные

Следующие переменные являются глобальными для методов выгрузки:

BM_UPLOAD_METHOD

Тип: список (разделитель пробел)

Список методов выгрузки.

Если вы не хотите использовать ни один из методов выгрузки, просто задайте значение none.

Пример:

```
export BM_UPLOAD_METHOD="scp"
```

BM_UPLOAD_HOSTS

Тип: список (разделитель пробел)

Каждый из хостов, определённых в этом списке используется всеми методами выгрузки. Например, если вы хотите выполнить выгрузку ваших архивов по [SSH](#) и [Rsync](#) выгрузку, поместите оба хоста в этот список.

Пример:

```
export BM_UPLOAD_HOSTS="mirror1.lan.mysite.net mirror2.lan.mysite.net"
```

BM_UPLOAD_DESTINATION

Тип: строка

Абсолютный путь к папке, находящейся на удалённых хостах, в которую будут выгружены файлы.

Если вы установили Backup Manager на удалённом хосте, будет хорошо использовать отдельную папку в хранилище. Затем, во время фазы чистки данных на этом хосте выгруженные туда файлы ваши тоже будут очищены.

Можно также определить отдельную папку для вашего хоста:

```
BM_UPLOAD_DESTINATION="/var/archives/$HOSTNAME"
```

Пример:

Допустим, вы хотите, чтобы все ваши загрузки хранились на хосте `mirror2.lan.mysite.net`, в папке `/var/archives/uploads`:

```
export BM_UPLOAD_HOSTS="mirror2.lan.mysite.net"
export BM_UPLOAD_DESTINATION="/var/archives/uploads"
```

SSH

Имя метода: `ssh`, цель: выгружать архивы на удалённые хосты через [SSH](#). Этот метод зависит от способа резервного копирования.

Если вы хотите выгружать свои архивы в удалённые хранилища, вы можете использовать метод SSH. Этот метод хорош, если вы будете использовать безопасный туннель между точками выгрузки.

Для вызова `scp` будет выполнен от имени пользователя заданного в переменной [BM_UPLOAD_SSH_USER](#), таким образом, вы должны заранее убедиться с том, что этот пользователь может иметь доступ к хранилищу.

BM_UPLOAD_SSH_USER

Тип: строка

Пользователь для соединения по SSH. Убедитесь, что этот пользователь может получить доступ к хранилищу.

Пример:

```
export BM_UPLOAD_SSH_USER="bmngr"
```

BM_UPLOAD_SSH_KEY

Тип: строка

Ключ для пользователя, который задан в переменной [BM_UPLOAD_SSH_USER](#).

Пример:

```
export BM_UPLOAD_SSH_KEY="/home/bmgr/.ssh/iddsa"
```

BM_UPLOAD_SSH_PORT

Тип: целое число.

Номер порта для соединения по SSH.

Пример:

```
export BM_UPLOAD_SSH_PORT="1352"
```

BM_UPLOAD_SSH_HOSTS

Тип: список (разделитель пробел)

Список хостов, только для выгрузки по SSH. Обратите внимание, что если вы поместите некоторые хосты в [BM_UPLOAD_HOSTS](#), они будут использоваться.

Пример:

```
export BM_UPLOAD_SSH_HOSTS="mirror3.lan.mysite.net"
```

BM_UPLOAD_SSH_PURGE

Тип: логическое

Если установить значение true, удалённые архивы будут очищены до того, как новые будут загружены. Backup Manager использует одинаковые правила очистки при локальной и удалённой работе. Если определена переменная [BM_UPLOAD_SSH_TTL](#), то будет использовано её значение, иначе значение из [BM_ARCHIVE_TTL](#).

Пример:

```
export BM_UPLOAD_SSH_PURGE="true"
export BM_UPLOAD_SSH_TTL="10"
```

BM_UPLOAD_SSH_DESTINATION

Тип: строка

Путь только для выгрузки по SSH, эта переменная переопределяет [BM_UPLOAD_DESTINATION](#).

Пример:

```
export BM_UPLOAD_SSH_DESTINATION="/var/archives/scp-uploads"
```

SSH GPG

Выгрузка по SSH с шифрованием

Метод: `ssh-gpg`, цель: шифрование архивов, используя метод шифрования с открытым ключом и выгрузка зашифрованных архивов по SSH в хранилища, к которым нет полного доверия. Этот метод зависит от способа

резервного копирования.

Выгрузку по SSH также можно комбинировать с шифрованием с открытым ключом средствами [GPG](#). Архивы будут зашифрованы с использованием открытого ключа перед отправкой их по сети, и таким образом файлы на удаленном сервере будут защищены.

Этот метод может быть использован для защиты ваших данных на серверах к которым нет доверия. Однако, так как зашифрованные файлы не подписаны, это не защитит вас от манипуляций с ними. Поэтому создание хешей md5 является необходимым.

Этот метод использует все переменные метода выгрузки по SSH. Необходимо определить только одну дополнительную переменную.

BM_UPLOAD_SSHGPG_RECIPIENT

Тип: строка

Этот параметр определяет получателя, для которого шифруется архив. Действующая спецификация является коротким или длинным идентификатором ключа, или описательное имя, как объяснено в документации на [GPG](#). Открытый ключ для этой идентичности должна быть в кольце для ключей пользователя, запустившего GPG, который является таким же, как указано `BM_UPLOAD_SSH_USER`. Для проверки выполните команду `gpg --list-ключи ID` от имени этого пользователя, где ID является идентификатором. Если GPG отображает только один ключ, то все в порядке. Для получения дополнительной информации обратитесь к документации на [GPG](#).

Пример:

```
export BM_UPLOAD_SSHGPG_RECIPIENT="email@address.com"
export BM_UPLOAD_SSHGPG_RECIPIENT="ECE009856"
```

FTP

Если безопасность не имеет большого значения в вашей локальной сети (между двумя точками сети) вы можете использовать метод выгрузки [FTP](#). Одним из главных плюсов этого метода является то, вы можете выполнять очистку отдельно на источнике и хранилище. Вы можете смело использовать этот метод для загрузки файлов на хост, для этого Вам просто нужно иметь учётную запись пользователя FTP сервера.

BM_UPLOAD_FTP_SECURE

Тип: логическое, значение по умолчанию: false.

Если значение переменной `true`, все при выгрузке по FTP будет использован SSL.

Пример:

```
export BM_UPLOAD_FTP_SECURE="true"
```

BM_UPLOAD_FTP_PASSIVE

Тип: логическое, значение по умолчанию: true.

Если значение переменной `true`, то будет использован [пассивный канал передачи данных FTP](#).

Пример:

```
export BM_UPLOAD_FTP_PASSIVE="true"
```

BM_UPLOAD_FTP_TTL

Тип: целое число, значение по умолчанию: `$BM_ARCHIVE_TTL`.

Например, можно настроить Backup Manager для локального создания архивов и выгрузки их по FTP на удалённый хост, а затем очистить их локально (не на удалённом хосте). Делать это возможно при установке нулевого значения локального TTL (`BM_ARCHIVE_TTL`) и ненулевого значения `BM_UPLOAD_FTP_TTL`.

Пример: # in your main conffile - /etc/backup-manager.conf

```
export BM_ARCHIVE_TTL="0"
export BM_UPLOADFTP_TTL="5"
export BM_POST_BACKUP_COMMAND="/usr/sbin/backup-manager --purge"
```

Содержимое задания Cron:

```
/usr/sbin/backup-manager
/usr/sbin/backup-manager --purge
```

⚠ Не задавайте `BM_POST_BACKUP_COMMAND` в основном файле настройки иначе вы получите бесконечный цикл.

BM_UPLOAD_FTP_USER

Тип: строка.

Имя пользователя для соединения по FTP.

Пример:

```
export BM_UPLOAD_FTP_USER="bmngr"
```

BM_UPLOAD_FTP_PASSWORD

Тип: строка.

Пароль пользователя для соединения по FTP (в открытом виде).

Пример:

```
export BM_UPLOAD_FTP_PASSWORD="secret"
```

BM_UPLOAD_FTP_HOSTS

Тип: список (разделитель пробел).

Поместите здесь список хостов, чтобы использовать их только для выгрузки по FTP. Обратите внимание, что если вы поместите некоторые хосты в переменную `BM_UPLOAD_HOSTS`, они тоже будут использоваться для выгрузки по FTP.

Пример:

```
export BMUPLOADFTPHOSTS="mirror4.lan.mysite.net"
```

BM_UPLOAD_FTP_DESTINATION

Тип: строка.

Здесь должен быть указан локальный путь на удалённом сервере для выгрузки по FTP, эта переменная перекрывает значение в `BM_UPLOAD_DESTINATION`.

Пример:

```
export BM_UPLOAD_FTP_DESTINATION="/var/archives/ftp-uploads"
```

BM_UPLOAD_FTP_PURGE

Тип: логическое, значение по умолчанию: true.

Вы можете задать очистку устаревших архивов перед выгрузкой новых. Эта очистка осуществляется через FTP и использует переменную `BM_ARCHIVE_TTL` ведёт себя таким же образом, как локальная очистка (хотя очистка через FTP не является рекурсивной).

Пример:

```
export BM_UPLOAD_FTP_PURGE="true"
```

BM_UPLOAD_FTP_TIMEOUT

Тип: целое число, значение по умолчанию: 120.

Задержка при выгрузке по FTP.

Эта переменная будет использоваться если при выгрузке по FTP не используется SSL (`BM_UPLOAD_FTP_SECURE` имеет значение false).

Пример:

```
export BM_UPLOAD_FTP_TIMEOUT="120"
```

Amazon S3

Amazon Simple Storage Service (S3) — веб-служба предоставляющая возможность для хранения и получения любого объёма данных, в любое время из любой точки сети, так называемый файловый хостинг.

Подробнее см. <http://aws.amazon.com>

Используя метод выгрузки S3 ваши архивы будут храниться на S3 Amazon. Вы должны выделить «bucket» исключительного для использования с Backup Manager. Каждый из созданных вами архивов будет загружен в S3 и хранится в этом «bucket» в имени ключа, который соответствует имени архива.

Как и в случае использования других способов резервного копирования Backup Manager не поможет вам в восстановлении файлов из архивов. Вы должны извлекать архивы из S3 с помощью других механизмов, таких как например, S3Shell от Amazon.

Обратите внимание, что при использовании этого метода выгрузки, значения переменной `BM_UPLOAD_HOSTS` будут проигнорированы и для выгрузки будет использован только хост `s3.amazonaws.com`.

BM_UPLOAD_S3_DESTINATION

Тип: строка.

Этот параметр необходим для метода выгрузки S3. Это указывает «bucket», используемый для хранения резервных копий данных. Если «bucket» не существует, он будет создан в качестве частного «bucket». Эта переменная перекрывает значение в `BM_UPLOAD_DESTINATION`. Обратите внимание, что Amazon требует, чтобы имена «bucket» были глобально уникальным.

Пример:

```
export BM_UPLOAD_S3_DESTINATION="mybackupbucket"
```

BM_UPLOAD_S3_ACCESS_KEY

Тип: строка.

Ключ доступа к S3. После того, как вы зарегистрировались Amazon предоставит вам ключ доступа. Вы должны использовать этот ключ, чтобы получить доступ к хранилищу на S3.

Пример:

```
export BM_UPLOADS3ACCESSKEY="a9sabkz0342dasv"
```

BM_UPLOAD_S3_SECRET_KEY

Тип: строка.

Ключ безопасности для доступа к S3. После того, как вы зарегистрировались Amazon предоставит вам ключ доступа. Вы должны использовать этот ключ, чтобы получить доступ к хранилищу на S3.

Пример:

```
export BM_UPLOAD_S3_SECRET_KEY="lkj2341askj123sa"
```

BM_UPLOAD_S3_PURGE

Тип: логическое, значение по умолчанию: *true*.

Вы можете задать очистку устаревших архивов перед выгрузкой новых. Эта очистка осуществляется через S3 и использует переменную [BM_ARCHIVE_TTL](#) ведёт себя таким же образом, как локальная очистка (хотя очистка S3 не является рекурсивной).

Пример:

```
export BM_UPLOAD_S3_PURGE="true"
```

Rsync

Если вы не хотите создавать архивы, то вы можете выгружать нужные объекты локальной файловой системы в на удалённые хосты. Для этого удобно использовать метод выгрузки `rsync`, который использует работе одноимённую программу: [Rsync](#).

Для использования метода выгрузки `rsync` нужны пользователь с доступом по SSH и его ключ, заданные в переменных [BM_UPLOAD_SSH_USER](#) и [BM_UPLOAD_SSH_KEY](#).

BM_UPLOAD_RSYNC_DIRECTORIES

Тип: список (разделитель пробел).

Список локальных папок, которые вы хотите выгружать методом `rsync`.

Пример:

```
export BM_UPLOAD_RSYNC_DIRECTORIES="/data/photos /data/videos /data/mp3"
```

BM_UPLOAD_RSYNC_HOSTS

Тип: список (разделитель пробел).

Поместите здесь список хостов, чтобы использовать их только для выгрузки методом rsync. Обратите внимание, что если вы поместите некоторые хосты в переменную `BM_UPLOAD_HOSTS`, они тоже будут использоваться для выгрузки методом rsync.

Пример:

```
export BM_UPLOAD_RSYNC_HOSTS="mirror5.lan.mysite.net"
```

BM_UPLOAD_RSYNC_DESTINATION

Тип: строка.

Здесь должен быть указан локальный путь на удалённом сервере для выгрузки методом rsync, эта переменная перекрывает значение в `BM_UPLOAD_DESTINATION`.

Пример:

```
export BM_UPLOAD_RSYNC_DESTINATION="/var/archives/rsync-snapshots"
```

BM_UPLOAD_RSYNC_DUMPSYMLINKS

Тип: логическое, значение по умолчанию: `false`.

Вы можете разыменовывать символические ссылки в ваших rsync-снимках.

 Очень осторожно используйте эту функцию.

Пример:

```
export BM_UPLOAD_RSYNC_DUMPSYMLINKS="false"
```

Запись на внешние носители

Ещё один способ хранения ваших архивов в безопасном месте это использование внешних носителей.

В версии 0.7.7, в качестве внешних носителей поддерживаются только CD и DVD. Префикс переменных: `BM_BURNING`.

CD/DVD

Запись на носители CDR/DVD.

В версии 0.7.7, Backup Manager поддерживает четыре вида носителей: CDR, CDRW, DVD+R(W) и DVD-R(W).

BM_BURNING_METHOD

Метод записи задаётся в переменной `BM_BURNING_METHOD`, он соответствует типу носителя:

- CDR

- CDRW
- DVD
- DVD-RW
- none

При запуске в пакетном режиме, при использовании любого из этих методов программа будет пытаться разместить все файлы архивов на один носитель, если не на носителе для это не хватает места, то будут выполнена попытка разместить только архивы на день, если и это не представляется возможным, тогда ничего записано не будет.

В интерактивном режиме (когда Backup Manager запускается из терминала), все файлы архивов будут записаны на необходимое для этого количество носителей. Каждый новый носитель информации будет запрошен по мере необходимости.

При использовании методов CDRW и DVD-RW нужно сначала очистить носитель чтобы записать на него новую информацию, это позволяет многократно использовать одни и те же носители.

При использовании методов CDR and DVD предварительная чистка носителя не нужна (носитель DVD+RW не нуждается в предварительной очистке, на таких носителях возможна добавлять данные «на лету»).

Для записи DVD используется dvd+rw-tools. До версии 6.1 ,были проблемы при использовании с [Cron](#), убедитесь, что у вас ваша версия dvd+rw-tools новее 6.1, если вы хотите записывать DVD носители с помощью Backup Manager.

Для отключения этого метода нужно присвоить переменной BM_BURNING_METHOD значение none.

Все методы записи использую одни и те же параметры для настройки, что позволяет легко изменить тип носителя.

BM_BURNING_DEVICE

Тип: строка, значение по умолчанию: /dev/cdrom.

Имя устройства для записи.

Пример:

```
export BM_BURNING_DEVICE="/dev/cdrom"
```

BM_BURNING_DEVFORCED

Тип: строка.

Backup Manager использует cdbrecord для записи CD. Если при запуске cdbrecord -scanbus вы не увидите ваше устройство для записи, это означает, что вам придётся перевести устройство в режим ATA. Для настройки Backup Manager на это режим укажите путь к устройству в этой переменной, тогда команда вызова cdbrecord будет приведена к виду: cdbrecord ... DEV = \$ BM_BURNING_DEVFORCED ...

Оставьте эту переменную пустой, если вы видите свое устройство в выводе команды cdbrecord -scanbus.

Пример:

```
export BM_BURNING_DEVFORCED="/dev/cdrom"
```

BM_BURNING_ISO_FLAGS

Тип: строка, значение по умолчанию: -R -J

Backup Manager использует образы дисков Joliet. Флаги, определённые в этой переменной будут добавлены к командным mkisofs которая используется для формирования образов.

По умолчанию: используются флаги: -R -J для создания образа диска Joliet, если вы хотите создавать другие образы то укажите нужные флаги. Для этого обратитесь к документации по mkisofs для получения подробной информации.

Не изменяйте значение этой переменной, если вы не знаете, что вы делаете.

Пример:

```
export BM_BURNING_ISO_FLAGS="-R -J"
```

BM_BURNING_MAXSIZE

Тип: целое число, значение по умолчанию: 700.

В этой переменной определяется максимальная ёмкость (в мегабайтах) устройства записи.

Список значений:

- CDR/CDRW: 650, 700, 800
- DVD: 4700

Backup Manager будет пытаться записать все файлы архивов на один носитель, если не на носителе для это не хватает места, то будут выполнена попытка записать только архивы на день.

Пример для записи CD:

```
export BM_BURNING_METHOD="CDRW"  
export BM_BURNING_MAXSIZE="700"
```

Пример для записи DVD:

```
export BM_BURNING_METHOD="DVD"  
export BM_BURNING_MAXSIZE="4700"
```

BM_BURNING_CHKMD5

Тип: логическое, значение по умолчанию: true.

Если значение переменной true то будет выполнена проверка контрольных сумм MD5 всех фалов после их записи на носитель, чтобы убедиться, что все в порядке.

Обратите внимание, что вы можете выбрать, чтобы выполнить эту проверку с помощью параметра командной строки `-md5check`.

Пример:

```
export BM_BURNING_CHKMD5="true"
```

Дополнительные возможности

Предусмотрено несколько дополнительных функций, они будут рассмотрены в этом разделе.

BM_TEMP_DIR

Тип: строка, значение по умолчанию: /tmp/backup-manager.

Путь к папке в которой Backup Manager будет создавать временные файлы.

Пример:

```
export BM_ARCHIVE_CHMOD="/tmp/backup-manager"
```

Использование syslog

Если вы хотите чтобы Backup Manager записывал протокол своих действий в [syslog](#), вы должны задать установить true в переменную BM_LOGGER. Также вы можете выбрать категории протокола с помощью переменной BM_LOGGER_FACILITY.

BM_LOGGER

Тип: логическое, значение по умолчанию: true. какие системного журнала для использования благодаря

Если значение переменной true то Backup Manager будет записывать протокол своих действий в [syslog](#).

Пример:

```
exports BM_LOGGER="true"
```

BM_LOGGER_FACILITY

Тип: строка, значение по умолчанию: user.

С помощью этой переменной вы можете выбрать категории протокола [syslog](#).

Пример:

```
exports BM_LOGGER_FACILITY="cron"
```

Запуск внешних программ

Backup Manager предоставляет два различных перехватчика (hooks) : pre-command и post-command.

Можно, например монтировать внешнее хранилище по NFS до начала резервного копирования и отключите его после его завершения.

BM_PRE_BACKUP_COMMAND

Тип: строка

В этой переменной можно задать путь к программе (или команде оболочки) которая будет выполнена до запуска процедуры резервного копирования. Если команда не выполниться (выход с не нулевым значением, или с кодом false на стандартный вывод), тогда процедура резервного копирования будет остановлена. Если предварительно команда будет выполнена успешно, то будет выполнена процедура резервного копирования.

Пример использования простой консольной команды:

```
export BM_PRE_BACKUP_COMMAND="mount -t nfs mirror.lan.net:/exports/backups /var/archives"
```

Пример вызова скрипта с параметром:

```
export BM_PRE_BACKUP_COMMAND="/usr/local/bin/backup-prepare.pl $TODAY"
```

BM_POST_BACKUP_COMMAND

Тип: строка.

В этой переменной можно задать путь к программе (или команде оболочки) которая будет выполнена после запуска

процедуры резервного копирования. Если команда не выполнится (выход с не нулевым значением, или с кодом `false` на стандартный вывод), Backup Manager будет завершаться с кодом ошибки (который будет записан в [syslog](#), если эта возможность включена).

Пример использования простой консольной команды:

```
export BM_POST_BACKUP_COMMAND="umount /var/archives"
```

Пример вызова скрипта с параметром:

```
export BM_POST_BACKUP_COMMAND="/usr/local/bin/backup-cleanup.pl $TODAY"
```

Использование

Теперь, когда вы знаете, в деталях, как писать ваши конфигурационные файлы, давайте рассмотрим, как использовать Backup Manager.



```
cat /etc/backup-manager.conf |grep -E 'export|declare' >backup-manager_new.conf
```

backup-manager

Скрипт на [Bash](#).

Начиная с версии 0.7.7, Backup Manager должен быть запущен от пользователя `root`.

`/usr/sbin/backup-manager`

```
$ backup-manager
backup-manager must be run as root.
```

Если вы хотите запустить его из командной строки, сначала вы должны авторизоваться, как `root`.

--help|-h

Вывести эту справочную информацию

--version

Вывести номер версии.

```
# backup-manager --version
Backup Manager 0.6
```

--verbose|-v

Выводить информацию о ходе работы на устройство стандартного вывода (`stdout`).

```
# backup-manager -v
Getting lock for backup-manager 10605 with /etc/backup-manager.conf: ok
Cleaning /var/archives
Entering directory /var/archives/lost+found.
[...]
```

--debug|-d

Выводить отладочную информацию на устройство стандартного вывода (stdout).

--no-warnings

Отключить предупреждения.

--upload|-u

Выгрузить файлы текущего дня.

--burn|-b

```
--burn|-b [<DATE>]
```

Записать файлы текущего дня. Или за дату.

Формат даты: ГГГГММДД

Записать все архивы за 12 марта 2006:

```
# backup-manager --bourn 20060312
```

--md5check|-m

Проверить контрольные суммы (md5).

--purge|-p

Очистить старые архивы.

Для подробной информации см. [BM_ARCHIVE_TTL](#)).

--conffile|-c file

Использовать другой файл с параметрами.

По умолчанию программа берёт настройки из файла `/etc/backup-manager.conf`.

Если нужно можно указать другой файл:

```
# backup-manager -c <ФАЙЛ>
```

Примеры:

```
backup-manager -c /etc/backup-manager/backup-nfs.conf &  
backup-manager -c /etc/backup-manager/backup-homedirs.conf &  
backup-manager -c /etc/backup-manager/backup-rsync-filer.conf
```

--force|-f

Перезаписать существующие архивы.

--no-upload

Отключить процесс выгрузки.

--no-burn

Отключить процесс записи.

--no-purge

Отключить процесс очистки.

backup-manager-purge



```
backup-manager-purge --ttl=<TTL> --files-from=<FILE>
```

Это скрипт на [Perl](#), который предназначен для удаления устаревших файлов.

Учитывая время жизни (TTL) и список архивов, backup-manager-purge будет возвращать список архивов, соответствующих тем, которые устарели в TTL.

Программа выводит список устаревших файлов на устройство стандартного вывода (stdout), по одному файлу в строке.

При возникновении ошибки на устройство стандартного вывода (stdout) выводиться сообщение об ошибке и происходит завершение программы.

Коды ошибок:

- Нет ошибок: 0
- Ошибка в командной строке (неправильные аргументы): 10
- Внутренняя ошибка: 20

--ttl|-t

Обязательный параметр.

```
--ttl=time-to-live
```

\$HOSTNAME Время жизни (в днях) для архивов. Любой архив, который старше этого количества дней будет считаться устаревшими.

--files-from|-f

```
--files-from=file
```

Файл который содержит список архивов для обработки, по одному файлу в строке. Вместо этого параметра можно

использовать устройство стандартного ввода (stdin).

backup-manager-upload

Скрипт на [Perl](#) для выгрузки архивов.

```
backup-manager-upload --help  
/usr/bin/backup-manager-upload -m=mode -h=host -u=user [options] date
```

При возникновении ошибки на устройство стандартного вывода (stdout) выводиться сообщение об ошибке и происходит завершение программы.

Коды ошибок:

- Нет ошибок: 0
- Ошибка в командной строке (неправильные аргументы): 10
- Сбой при передаче по [ftp](#): 20
- Сбой при передаче по [scp](#): 21
- Сбой при передаче по [s3](#): 22
- Неизвестный метод выгрузки: 23

-v|--verbose

Выводить информацию о ходе работы на устройство стандартного вывода (stdout).

-m|--mode

Обязательный параметр.

Методы выгрузки:

- [ftp](#)
- [scp](#)
- [s3](#)
- [ssh-gpg](#)

-h|--host

Обязательный параметр.

Список хостов для выгрузки (через запятую).

Пример:

```
--host=hostname1,hostname2,...,hostnameN
```

-u|--user

Обязательный параметр.

Имя пользователя.

-p|--password

Пароль (нужны для выгрузки по [FTP](#) и [Amazon S3](#)).

-k|--key

SSH ключ для пользователя для [scp](#) сессии.

Пример:

```
--key=path_to_private_key
```

-d|--directory

Абсолютный путь к папке, находящейся на удалённых хостах, в которую будут выгружены файлы. (значение по умолчанию: `/var/archives/uploads`).

-b|--bucket

Имя bucket для [Amazon S3](#).

-r|--root

Путь к локальной папке хранилищу архивов (значение по умолчанию: `/var/archives`).

-l|--list

Только вывести список файлов для выгрузки.

--gpg-recipient

Идентификатор получателя [GPG](#) шифрования (только для [ssh-gpg](#)).

Пример:

```
--gpg-recipient=gpg-recipient
```

--ftp-purge

Очистить удалённую папку до начала выгрузки по [FTP](#).

--ftp-test

Создать тестовый файл начала выгрузки по [FTP](#).

--s3-purge

Очистить удалённую папку до начала выгрузки в режиме [S3](#).

--s3-maxsize

Максимальный размер файла для выгрузки по [S3](#). Значение по умолчанию: 4 Гбайт.

--ssh-purge

Очистить удалённую папку до начала выгрузки по [SSH](#).

date

Выгружать все файлы, дата которых $\geq$ (больше или равна) заданной.

Формат даты: ГГГГММДД. Также можно использовать ключевые слова: today, yesterday.

Примеры

Только выгрузка архива по FTP

```
/usr/bin/backup-manager-upload -m=ftp -h=СЕРВЕР -u=ПОЛЬЗОВАТЕЛЬ -p=ПАРОЛЬ -  
r=ЛОКАЛЬНЫЙ_ПУТЬ_К_АРХИВАМ -d=/УДАЛЕННЫЙ_ПУТЬ_К_АРХИВАМ
```

Cron

Для автоматизации процесса резервного копирования можно использовать [Cron](#).

Для настройки [Cron](#) на ежедневный запуск Backup Manager вы должны создать файл: /etc/cron.daily/backup-manager:

[/etc/cron.daily/backup-manager](#)

```
#!/bin/sh  
  
/usr/sbin/backup-manager
```

Если вы хотите получать уведомления по электронной почте о проблемах во время выполнения резервного копирования, вам нужно убедиться, что вы получаете письма от Cron и выбрать добавит параметр `--verbose` в сроку запуска Backup Manager:

[/etc/cron.daily/backup-manager](#)

```
#!/bin/sh  
  
/usr/sbin/backup-manager --verbose
```

Ссылки

[User Guide](#)

[Установка и конфигурирование backup-manager](#)

[openwiki.ru : Backup Manager](#)

1)

Для получения дополнительной информации читайте: [Перевод на русский язык программ](#)

<http://sysadminmosaic.ru/backup-manager/backup-manager>

2020-09-19 12:34

