

Установка и настройка Bacula

- Astra Linux Смоленск 1.5
 - Подготовка инфраструктуры для управления системой резервного копирования
 - Настройка bacula
 - Настройка Director Daemon
 - Настройка Storage Daemon
 - Настройка File Daemon
- Astra Linux Смоленск 1.2
 - Подготовка инфраструктуры для управления системой резервного копирования
 - Настройка bacula
 - Настройка Director Daemon
 - Настройка Storage Daemon
 - Настройка File Daemon

Astra Linux Смоленск 1.5

Подготовка инфраструктуры для управления системой резервного копирования

Для проведения теста необходимо три машины:

Director Daemon 10.0.0.23 - главный сервер, осуществляющий резервное копирование, назовем его *bakula-dir*.

Storage Daemon 10.0.0.24 - машина, куда будут складываться бэкапы. Ее назовем *bakula-sd*.

File Daemon 10.0.0.25 - машина, откуда будут копироваться данные и куда будет разворачиваться бэкап. Это *bakula-fd*.

На Director Daemon должны быть установлены пакеты:

`postgresql-9.4 pgadmin3 bacula-common bacula-common-pgsql bacula-console bacula-director-common bacula-director-pgsql`



При настройке базы bacula может произойти ошибка, проигнорируйте ее, базу мы настроим позже.

На Storage Daemon должны быть установлены пакеты:

`bacula-sd-pgsql`

На File Daemon:

`bacula-fd`

На Director Daemon

В файле `/etc/postgresql/9.4/main/postgresql.conf` указать параметр:

```
listen_addresses='*'
```

В файле `/etc/postgresql/9.4/main/pg_hba.conf` указать метод `trust` для всех.

Обязательно добавить `host` с `ip` адресом, где будет работать `bacula-dir`.

```
local all postgres trust
local all all trust
host all all 127.0.0.1/32 trust
host all all 10.0.0.23/24 trust
```

Выполнить запуск БД:

```
pg_ctlcluster 9.4 main restart
```

Присвоить пароль пользователю postgres:

```
passwd postgres
```

Пароль ввести 1

Присвоить пароль пользователю bacula:

```
passwd bacula
```

Пароль ввести bacula

Создать пользователя БД для работы с bacula, выполнять не из-под root:

```
#psql template1 -U postgres -h 10.0.0.23 -p PORT
```

где PORT можно определить командой `pg_lsclusters`

```
template1=# CREATE ROLE bacula;  
template1=# ALTER USER bacula PASSWORD 'bacula';  
template1=# ALTER USER bacula LOGIN SUPERUSER CREATEDB CREATEROLE;
```

Создать БД bacula

```
CREATE DATABASE bacula;  
ALTER DATABASE bacula OWNER TO bacula;
```

Для корректного функционирования авторизации через PAM, пользователю postgres необходимо выдать права на чтение информации из БД пользователей и сведений о метках безопасности и привилегиях:

```
usermod -a -G shadow postgres  
setfacl -d -m u:postgres:r /etc/passwd  
setfacl -R -m u:postgres:r /etc/passwd  
setfacl -m u:postgres:rx /etc/passwd  
  
setfacl -d -m u:postgres:r /etc/capdb  
setfacl -R -m u:postgres:r /etc/capdb  
setfacl -m u:postgres:rx /etc/capdb
```

Пользователю bacula задаем минимальный и максимальный уровень:

```
pdpl-user bacula -l 0:0
```

В скриптах `/usr/share/bacula-director/make_postgresql_tables` и `/usr/share/bacula-director/grant_postgresql_privileges` вносим изменения:

```
db_name=${db_name:-bacula}  
psql -U bacula -h 10.0.0.23 -p 5433 -f - -d ${db_name} $* <<END-OF-DATA
```

Сохраняем изменения и запускаем скрипты `make_postgresql_tables` и `grant_postgresql_privileges`

На машине, где будет работать Storage Daemon необходимо создать каталог /back, в него будут сохраняться бэкапы, и присвоить ему владельца bacula:

```
mkdir /back
chown -R bacula /back
```

На машине, где будет работать File Daemon необходимо создать каталог /etc2, в него будет разворачиваться бэкап:

```
mkdir /etc2
```

Если подготовительные настройки выполнены корректно, база стартует без ошибок, скрипты выполнились без ошибок, то можно приступить к настройке bacula.

Настройка bacula

Подготовка Bacula к работе заключается в настройке каждого компонента в отдельности и последующей настройке их взаимодействия.

Настройка Director Daemon

Приступаем к конфигурированию Director Daemon в конфигурационном файле /etc/bacula/bacula-dir сервера bakula1.

В первую очередь необходимо определить основные параметры в секции Director. На начальном этапе важно установить параметры Name и Password. Name задаёт уникальное имя Director Daemon, а Password – пароль, который будет использоваться при соединениях BC с DD. Остальные параметры можно оставить в значениях по умолчанию.

```
✓ Director {
    Name = bacula-dir
    DIRport = 9101
    QueryFile = "/etc/bacula/scripts/query.sql"
    WorkingDirectory = "/var/lib/bacula"
    PidDirectory = "/run/bacula"
    Maximum Concurrent Jobs = 1
    Password = "1" # Console password
    Messages = Daemon
    DirAddress = 10.0.0.23
}
```

Блок JobDefs нам не нужен, его можно закомментировать.

Все настройки мы свяжем воедино с помощью секции Job, в которой мы даём задание планировщику по выполнению резервирования данных. Здесь мы уже не будем описывать все параметры, отметив лишь основные.

Параметр Type указывает на тип задания. Типов существует несколько, и к ним мы ещё вернемся в продолжение цикла. Здесь же достаточно указать Backup.

Параметр Schedule указывает на предопределённое расписание, согласно которому будет выполняться резервирование данных. Все расписания определены здесь же, в файле bacula-dir.conf. Пример расписания WeeklyCycle приведён ниже, комментировать в нём нечего.

Параметр Where указывает на каталог, где будет развернут бэкап.

Параметр Write Bootstrap указывает путь к файлу, в который будет записываться информация, с помощью которой данные могут быть восстановлены из резервной копии без наличия подключения к Bacula Catalog. Вместо %n будет подставлено значение параметра Name:

```

✔ Job {
    Name = "BackupClient1"
    Type = Backup
    Client = bacula-fd
    FileSet="Catalog"
    Schedule = "DailyCycle"
    Messages = Standard
    Pool = Default
    Write Bootstrap = "/var/lib/bacula/%n.bsr"
    Priority = 1
}

Job {
    Name = "RestoreFiles"
    Type = Restore
    Client=bacula-fd
    FileSet="Catalog"
    Storage = File
    Pool = Default
    Messages = Standard
    Where = /home2
}

```

Одна из интересных секций – FileSet, она позволяет предопределить несколько наборов резервируемых файлов. Например, один набор для Windows, другой – для Linux или один для серверов, а другой – для рабочих станций. Основные параметры здесь следующие.

Параметр Name определяет уникальное имя набора.

Секция Include содержит пути к резервируемым файлам/каталогам, а Exclude – пути к файлам и каталогам, которые необходимо исключить из списка резервируемых. В секции Include возможна секция Options, в которой определяются параметры резервирования.

Параметр signature указывает алгоритм вычисления контрольных сумм файлов.

Параметр compression указывает алгоритм компрессии файлов.

Параметр recurse указывает на необходимость рекурсивного резервирования, включая подкаталоги и файлы.

Параметр File указывает на каталог, который мы копируем.

Параметр xattrsupport указывает на возможность включения поддержки расширенных атрибутов, это обязательный параметр для работы с метками безопасности.

```

FileSet {
    Name = "Catalog"
    Include {
    Options {
        signature = MD5
        compression = GZIP
        aclsupport = yes
        xattrsupport = yes
    }
    File = /home
}

```

Блок Exclude нужно закомментировать

```

Schedule {
    Name = "DailyCycle"
    Run = Full daily at 15:30
    Run = Differential 2nd-5th sun at 23:05
    Run = Incremental mon-sat at 23:05
}

```

Второй блок Schedule и еще один блок FileSet нужно закомментировать.

Ну, а теперь укажем параметры нашего единственного Агента.

```

Client {
    Name = bacula-fd
    Address = 10.0.0.25
    FDPort = 9102
    Catalog = MyCatalog
    Password = "1"
    File Retention = 30 days
    Job Retention = 6 months
    AutoPrune = yes
}

```

Теперь настала пора определить SD, на который будет производиться передача данных для дальнейшей записи на устройство хранения. Storage Daemon у нас настроен и готов к работе, определим реквизиты доступа к нему в секции Storage файла bacula-dir.conf.

Параметр Name – уникальное имя, используемое для адресации секции Storage в рамках файла bacula-dir.conf

Параметры Device и MediaType дублируют одноименные параметры файла bacula-sd.conf.

Параметр Password содержит пароль, который будет использоваться при подключении к Storage Daemon.

```
Storage {
    Name = File
    Address = 10.0.0.24
    SDPort = 9103
    Password = "1"
    Device = FileStorage
    Media Type = File
}
```

Следующей группой параметров, которые необходимо определить, является секция Catalog. Здесь необходимо указать реквизиты доступа к базе данных, а также назначить уникальное имя данного Bacula Catalog с помощью параметра Name:

```
Catalog {
    Name = MyCatalog
    dbname = "bacula"; DB Address = "10.0.0.23"; dbuser = "bacula"; dbpassword = "bacula"; DB PORT = "5433"
}

Messages {
    Name = Standard
    mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\ (Bacula) \<%r\>\" -s \"Bacula: %t %e of %c %l\" %r"
    operatorcommand = "/usr/sbin/bsmtp -h localhost -f \"\ (Bacula) \<%r\>\" -s \"Bacula: Intervention needed for %j\" %r"
    mail = root = all, !skipped
    operator = root = mount
    console = all, !skipped, !saved
    append = "/var/log/bacula/bacula.log" = all, !skipped
    catalog = all
}

Messages {
    Name = Daemon
    mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\ (Bacula) \<%r\>\" -s \"Bacula daemon message\" %r"
    mail = root = all, !skipped
    console = all, !skipped, !saved
    append = "/var/log/bacula/bacula.log" = all, !skipped
}
```

Секция Pool определяет набор носителей информации и параметры, определяющие то, как SD будет их обрабатывать. Каждый Pool взаимодействует с устройством хранения данных, и поэтому необходимо создать столько же пулов, сколько определено устройств хранения. Фактически если для каждого File Daemon вы определяете отдельное устройство, то для каждого FD необходимо определить и Pool. Рассмотрим перечень наиболее значимых параметров.

Параметр Name определяет уникальное имя пула.

Параметр Pool Type определяет тип, и для резервных копий должен быть установлен в значение Backup.

Параметр Maximum Volume Jobs рекомендуется установить в значение 1. Это будет означать, что в рамках одного носителя данных могут быть размещены резервные данные, полученные в ходе выполнения только одного задания. Носитель данных – это устройство, на которое непосредственно записываются данные (оптические диски, магнитные ленты). Если размер созданной резервной копии много меньше размера носителя, то имеет смысл сохранить на него и другие копии, которые будут создаваться в будущем. Но если мы говорим о файлах, то желательно придерживаться правила "один файл – одна копия", т.е. в одном файле Bacula должны храниться резервные данные, которые были сформированы в рамках выполнения одного задания. Для каждого последующего будут создаваться новые файлы.

Volume Retention – время, по прошествии которого данные о резервной копии, хранящейся на носителе, будут удалены из каталога. Для обеспечения работоспособности Bacula необходимо помнить о том, что информация обо всех зарезервированных файлах хранится в базе данных, по записи на каждый файл. Если вы резервируете тысячи файлов, то очень скоро база данных станет огромной, что может затруднить работу Bacula. Поэтому очень важно своевременно очищать базу от устаревшей информации. При этом сам носитель информации не будет очищен автоматически. Он будет промаркирован как устаревший, но вы всегда сможете использовать его для восстановления данных в ручном режиме.

Maximum Volumes – максимальное количество носителей (в нашем случае файлов), доступных в данном пуле.

Параметр Recycle указывает на необходимость повторного использования носителей, помеченных как устаревшие. При этом реальная перезапись носителя произойдет лишь в случае, когда свободных носителей не останется. Свободные носители определяются из параметра Maximum Volumes.

Параметр AutoPrune указывает на то, необходимо ли производить удаление устаревших записей из Bacula Catalog автоматически после завершения выполнения очередного задания.

Параметр Label Format определяет префикс, который будет использован Bacula для маркирования носителей информации, в нашем случае – для именования файлов.

Параметр Storage указывает на имя устройства хранения данных, указанного в параметре Name секции Storage файла bacula-dir.conf.

```
Pool {
    Name = Default
    Pool Type = Backup
    Recycle = yes
    AutoPrune = yes
    Volume Retention = 1 month
    Maximum Volume Jobs = 5
    Maximum Volumes = 32
    Storage = File
    Label Format = "volume-"
}
```

Два следующих блока Pool тоже комментируем.

```
Console {
    Name = bacula-mon
    Password = "1"
    CommandACL = status, .status
}
```

Далее необходимо настроить доступ к DD со стороны Bacula Console в файле `/etc/bacula/bconsole.conf` сервера Director Daemon :

```
Director {
    Name = bacula-dir
    DIRport = 9101
    address = 10.0.0.23
    Password = "1"
}
```

На машине, где развернут Director Демон следует удалить пакеты `bacula-sd` и `bacula-fd`, если они установлены:

```
apt-get remove bacula-sd
apt-get remove bacula-fd
```

Конфиги `bacula-sd` и `bacula-fd` в `/etc/bacula` следует либо переименовать, либо удалить.
Сервисы `bacula-sd` и `bacula-fd` следует остановить:

```
/etc/init.d/bacula-sd stop
/etc/init.d/bacula-fd stop
```

Настройка Storage Daemon

Начнём подготовку Storage Daemon, который будет отвечать за непосредственную работу с устройством хранения данных. Bacula поддерживает широкий спектр устройств, начиная от оптических дисков и заканчивая полнофункциональными ленточными библиотеками. Список поддерживаемых устройств вы сможете найти на официальном сайте проекта. Мы рассмотрим самый распространённый вариант – обычный жёсткий диск с существующей файловой системой (например, `ext3`). Итак, на сервере `bakula2` редактируем файл `/etc/bacula/bacula-sd.conf`. В нём находим секцию основных параметров – `Storage`, где определяем параметр `Name`, который задаёт уникальное имя Storage Daemon. Остальные параметры можно оставить в значениях по умолчанию. Секция Director необходима для указания уникального имени DD и пароля, с которым этот DD может подключаться к SD. Секций может быть несколько, что даёт возможность использовать единый сервер хранения данных для нескольких систем резервирования. Все остальные секции Director, найденные в файле, прокомментируйте, они пока не понадобятся.

```
Storage {
    Name = bacula-sd
    DPort = 9103
    WorkingDirectory = "/var/lib/bacula"
    Pid Directory = "/var/run/bacula"
    Maximum Concurrent Jobs = 20
    SDAddress = 10.0.0.24
}

Director {
    Name = bacula-dir
    Password = "1"
}
```

Но основные настройки, определяющие взаимодействие с устройствами хранения, кроются в секции `Device`. Рассмотрим параметры, необходимые для хранения резервных копий в рамках существующей файловой системы, подключенной в каталог `/back`.

Параметр `Name` определяет уникальное имя подключенного устройства. Если вы планируете создавать изолированные друг от друга резервные копии для каждого из File Daemon, то вам необходимо создать несколько секций `Device` с уникальными именами. В противном случае резервируемые файлы со всех FD будут размещаться в одном и том же файле, что может несколько затруднить дальнейшее обслуживание системы.

Параметр `Media Type` определяет произвольное уникальное имя, которое будет использоваться Bacula при восстановлении данных. Согласно ему определяется устройство хранения, с которого будет производиться восстановление. Если вы храните резервные копии в файлах, то для КАЖДОЙ секции `Device` должен быть задан уникальный `Media Type`.

Параметр `Archive Device` указывает путь к файлу устройства в каталоге `/dev` или путь к каталогу, в котором будут размещаться резервные копии.

Параметр `Device Type` определяет тип устройства. Для размещения в существующей файловой системе указывается `File`.

Параметр `Random Access` указывает на возможность случайной (непоследовательной) адресации. Для файлов указывается `Yes`.

Параметр `RemovableMedia` указывает, возможно ли извлечение устройства хранения. Необходимо для ленточных устройств, приводов оптических дисков и т.д. Для файлов устанавливается в значение `No`.

Параметр `LabelMedia` указывает на необходимость автоматического маркирования носителей информации.

```
Device {
    Name = FileStorage
    Media Type = File
    Archive Device = /back
    LabelMedia = yes;
    Random Access = Yes;
    AutomaticMount = yes;
    RemovableMedia = no;
    AlwaysOpen = no;
}

Messages {
    Name = Standard
    director = bacula-dir = all
}
```

Для базовой настройки этого достаточно

На машине, где будет Storage Daemon следует удалить пакет bacula-fd:

```
apt-get remove bacula-fd
```

Конфиг bacula-fd в /etc/bacula следует либо переименовать, либо удалить.
Сервис bacula-fd следует остановить:

```
/etc/init.d/bacula-fd stop
```

Настройка File Daemon

Перейдём к настройке File Daemon на рабочей станции bakula3.

Для настройки используется файл /etc/bacula/bacula-fd, в котором для базовой настройки достаточно лишь определить параметры секции Director, где указывается пароль, который будет использовать DD при подключении к FD, а также секции FileDaemon, где указываются настройки FD. Все остальные секции Director, найденные в файле, прокомментируйте, они пока не понадобятся.

```
Director {
    Name = bacula-dir
    Password = "1"
}
```

В секции FileDaemon нас, на данном этапе, интересует только параметр Name, в котором указывается уникальное имя File Daemon:

```
FileDaemon {
    Name = bacula-fd
    Fdport = 9102
    WorkingDirectory = /var/lib/bacula
    Pid Directory = /var/run/bacula
    Maximum Concurrent Jobs = 20
    FdAddress = 10.0.0.25
}
```

```
Messages {
    Name = Standard
    director = bacula-dir = all, !skipped, !restored
}
```

На машине, где будет File Daemon следует удалить пакет bacula-sd:

```
apt-get remove bacula-sd
```

Конфиг bacula-sd в /etc/bacula следует либо переименовать, либо удалить.
Сервис bacula-sd следует остановить:

```
/etc/init.d/bacula-sd stop
```

Теперь запустим все компоненты соответствующими командами, данными на соответствующих серверах::

```
/etc/init.d/bacula-director restart
/etc/init.d/bacula-sd restart
/etc/init.d/bacula-fd restart
```

После этого bacula будет работать. Управление bacula осуществляется через bconsole. Настройки каталогов, заданий, расписаний и прочие задаются в конфигурационных файлах.

Для тестовой проверки выполните bconsole, далее выполните run, выберете job 1, войдите в меню набрав mod, выберете 1 (Level), выберете 1 (Full), подтвердите выполнение набрав yes. Будет создан бэкап в каталоге /back на машине с Storage Daemon. Для разворота бэкапа выполните restore, выберете пункт 12, введите номер job id, укажите параметр маркировки mark *, подтвердите выполнение командой done. Будет развернут бэкап в каталоге /etc2 на машине с File Daemon.

Astra Linux Смоленск 1.2



Все действия выполняются от root, настройка нашего стенда может отличаться именами и ip адресами от Ваших.



В нашем примере мы будем рассматривать следующую инфраструктуру:

- выделенный сервер `bacula1.my.dom` с ip адресом `11.11.11.21` (на нём будет функционировать **Director Daemon**, это главный сервер осуществляющий резервное копирование);
- выделенный сервер `bacula2.my.dom` с ip адресом `11.11.11.22` (на нём будет функционировать **Storage Daemon** это storage машина, куда будут складываться бэкапы);
- персональный компьютер `bacula3.my.dom` с ip адресом `11.11.11.23` (на нём будет функционировать **File Daemon** это машина, откуда будут копироваться данные и куда будет разворачиваться бэкап).

Подготовка инфраструктуры для управления системой резервного копирования

Установить **Postgresql-8.4** на сервер, где будет работать **Director Daemon**.

```
aptitude install postgresql-8.4
```

Предполагается, что на всех машинах изначально установлены все пакеты касающиеся bacula из состава операционной системы Astra Linux Special Edition версии 1.2 РУСБ 10015-07. Необходимо установить **bacula-director-common**, **bacula-director-pgsql** из состава операционной системы Astra Linux Special Edition версии 1.2. Потребуется доустановить **mailx** из состава операционной системы Astra Linux Special Edition версии 1.2 РУСБ 10015-07 и **dbconfig-common** из состава операционной системы Astra Linux Special Edition версии 1.2.

```
aptitude install mailx
```

Скачайте пакеты **bacula-director-common**, **bacula-director-pgsql**, **dbconfig-common** из состава операционной системы Astra Linux Special Edition версии 1.2 и выполните их установку:

```
dpkg -i dbconfig-common*.deb
dpkg -i bacula-director-common*.deb
dpkg -i bacula-director-pgsql*.deb
```



При настройке базы **bacula** может произойти ошибка, проигнорируйте ее, базу мы настроим позже.

Подготовить БД для **bacula**

В файле `/etc/postgresql/8.4/main/postgresql.conf` указать `listen = '*'`

В файле `/etc/postgresql/8.4/main/pg_hba.conf` внести необходимые изменения, для простоты можно указать метод `trust` для всех, удалить любую дополнительную конфигурацию после метода, типа `mod=`

Обязательно добавить `host` с ip адресом, где будет работать **bacula-dir**. В случае если все демоны **bacula** будут установлены на одну машину, указывать ip не обязательно, так как работа будет идти через `localhost`

Для первоначальной настройки Вы можете использовать наш файл [pg_hba.conf](#)

Выполните запуск БД:

```
pg_ctlcluster 8.4 main restart
```

Присвоить пароль postgres:

```
passwd postgres
```

запомните пароль.

Присвоить пароль bacula:

```
passwd bacula
```

пароль присваивайте bacula.

Создать пользователя БД для работы с **bacula**, выполнять не из-под `root`:


```
# psql template1 postgres
```

```
postgres=# CREATE ROLE bacula;  
postgres=# ALTER USER bacula PASSWORD 'bacula';  
postgres=# ALTER USER bacula LOGIN SUPERUSER CREATEDB CREATEROLE;
```

Создать БД **bacula**.

Выполнить не от root **fly-admin-postgres**, указать имя *template1*, пользователя *postgres*, пароль присвоенный *postgres*. В секции Роли входа добавить роль входа *bacula*. Создать базу данных *bacula*, владельцем назначить *bacula*.

На сервере bakula1.my.dom запускаем скрипты, которые создадут все необходимые таблицы и привилегии:

В скриптах */usr/share/bacula-director/make_postgresql_tables.sh* и */usr/share/bacula-director/grant_postgresql_privileges.sh* в строке после `if $bindir/psql` добавить `-U bacula`, сохранить изменения и выполнить скрипты:

```
make_postgresql_tables.sh  
grant_postgresql_privileges.sh
```

При возникновении необходимости файлы [make_postgresql_tables.sh](#), [grant_postgresql_privileges.sh](#) можно скачать у нас с готовыми правками.

На машине, где будет работать **Storage Daemon** необходимо создать каталог */back*, в него будут сохраняться бэкапы, и присвоить ему владельца *bacula*:

```
mkdir /back  
chown -R bacula /back
```

На машине, где будет работать **File Daemon** необходимо создать каталог */etc2*, в него будет разворачиваться бэкап:

```
mkdir /etc2
```

Если подготовительные настройки выполнены корректно, база стартует без ошибок, скрипты выполнились без ошибок, то можно приступить к настройке *bacula*.

Настройка bacula



Подготовка Bacula к работе заключается в настройке каждого компонента в отдельности и последующей настройке их взаимодействия.

Настройка Director Daemon

Приступаем к конфигурированию **DirectorDaemon** в конфигурационном файле */etc/bacula/bacula-dir* сервера bakula1.my.dom. В первую очередь необходимо определить основные параметры в секции **Director**. На начальном этапе важно установить параметры *Name* и *Password*. *Name* задаёт уникальное имя *Director Daemon*, а *Password* – пароль, который будет использоваться при соединениях **BC** с **DD**. Остальные параметры можно оставить в значениях по умолчанию.

```
Director { # define myself  
Name = bacula-dir  
DIRport = 9101 # where we listen for UA connections  
QueryFile = "/etc/bacula/scripts/query.sql"  
WorkingDirectory = "/var/lib/bacula"  
PidDirectory = "/var/run/bacula"  
Maximum Concurrent Jobs = 1  
Password = "1" # Console password  
Messages = Daemon  
DirAddress = 11.11.11.21  
}
```

Следующей группой параметров, которые необходимо определить, является секция *Catalog*. Здесь необходимо указать реквизиты доступа к базе данных, а также назначить уникальное имя данного Bacula *Catalog* с помощью параметра *Name*:

```

Catalog {
Name = MyCatalog
# Uncomment the following line if you want the dbi driver
# dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
DB Address = 11.11.11.21
}

```

Теперь настала пора определить **SD**, на который будет производиться передача данных для дальнейшей записи на устройство хранения. **Storage Daemon** у нас настроен и готов к работе, определим реквизиты доступа к нему в секции Storage файла *bacula-dir.conf*.

Параметр Name – уникальное имя, использующееся для адресации секции Storage в рамках файла *bacula-dir.conf*

Параметры Device и MediaType дублируют одноимённые параметры файла *bacula-sd.conf*.

Параметр Password содержит пароль, который будет использоваться при подключении к Storage Daemon.

```

Storage {
Name = File
# Do not use "localhost" here
Address = 11.11.11.22 # N.B. Use a fully qualified name here
SDPort = 9103
Password = "1"
Device = FileStorage
Media Type = File
}

```

До завершения базовой настройки Director Daemon осталось совсем немного. Секция Pool определяет набор носителей информации и параметры, определяющие то, как SD будет их обрабатывать. Каждый Pool взаимодействует с устройством хранения данных, и поэтому необходимо создать столько же пулов, сколько определено устройств хранения. Фактически если для каждого File Daemon вы определяете отдельное устройство, то для каждого FD необходимо определить и Pool. Рассмотрим перечень наиболее значимых параметров.

Параметр Name определяет уникальное имя пула.

Параметр Pool Type определяет тип, и для резервных копий должен быть установлен в значение Backup.

Параметр Maximum Volume Jobs рекомендуется установить в значение 1. Это будет означать, что в рамках одного носителя данных могут быть размещены резервные данные, полученные в ходе выполнения только одного задания. Носитель данных – это устройство, на которое непосредственно записываются данные (оптические диски, магнитные ленты). Если размер созданной резервной копии много меньше размера носителя, то имеет смысл сохранить на него и другие копии, которые будут создаваться в будущем. Но если мы говорим о файлах, то желательно придерживаться правила "один файл – одна копия", т.е. в одном файле Bacula должны храниться резервные данные, которые были сформированы в рамках выполнения одного задания. Для каждого последующего будут создаваться новые файлы.

Volume Retention – время, по прошествии которого данные о резервной копии, хранящейся на носителе, будут удалены из каталога. Для обеспечения работоспособности Bacula необходимо помнить о том, что информация обо всех зарезервированных файлах хранится в базе данных, по записи на каждый файл. Если вы резервируете тысячи файлов, то очень скоро база данных станет огромной, что может затруднить работу Bacula. Поэтому очень важно своевременно очищать базу от устаревшей информации. При этом сам носитель информации не будет очищен автоматически. Он будет промаркирован как устаревший, но вы всегда сможете использовать его для восстановления данных в ручном режиме.

Maximum Volumes – максимальное количество носителей (в нашем случае файлов), доступных в данном пуле.

Параметр Recycle указывает на необходимость повторного использования носителей, помеченных как устаревшие. При этом реальная перезапись носителя произойдёт лишь в случае, когда свободных носителей не останется. Свободные носители определяются из параметра Maximum Volumes.

Параметр AutoPrune указывает на то, необходимо ли производить удаление устаревших записей из Bacula Catalog автоматически после завершения выполнения очередного задания.

Параметр Label Format определяет префикс, который будет использован Bacula для маркирования носителей информации, в нашем случае – для именования файлов.

Параметр Storage указывает на имя устройства хранения данных, указанного в параметре Name секции Storage файла bacula-dir.conf.

```
Pool {
Name = Default
Pool Type = Backup
Recycle = yes # Bacula can automatically recycle Volumes
AutoPrune = yes # Prune expired volumes
Volume Retention = 1 month # one year
Maximum Volume Jobs = 1
Maximum Volumes = 32
Storage = File
Label Format = "volume-"
}
```

Одна из интересных секций – FileSet, она позволяет предопределить несколько наборов резервируемых файлов. Например, один набор для Windows, другой – для Linux или один для серверов, а другой – для рабочих станций. Основные параметры здесь следующие.

Параметр Name определяет уникальное имя набора.

Секция Include содержит пути к резервируемым файлам/каталогам, а Exclude – пути к файлам и каталогам, которые необходимо исключить из списка резервируемых. В секции Include возможна секция Options, в которой определяются параметры резервирования.

Параметр signature указывает алгоритм вычисления контрольных сумм файлов.

Параметр compression указывает алгоритм компрессии файлов.

Параметр recurse указывает на необходимость рекурсивного резервирования, включая подкаталоги и файлы.

Параметр File указывает на каталог, который мы копируем.

Параметр xattrsupport указывает на возможность включения поддержки расширенных атрибутов, это обязательный параметр для работы с метками безопасности.

```
FileSet {
Name = "Catalog"
Include {
Options {
signature = MD5
compression = GZIP
# recurse = yes
aclsupport = yes
xattrsupport = yes
}
File = /etc
}
}
```

Все настройки мы свяжем воедино с помощью секции Job, в которой мы даём задание планировщику по выполнению резервирования данных. Здесь мы уже не будем описывать все параметры, отметив лишь основные.

Параметр Type указывает на тип задания. Типов существует несколько, и к ним мы еще вернемся в продолжение цикла. Здесь же достаточно указать Backup.

Параметр Schedule указывает на предопределённое расписание, согласно которому будет выполняться резервирование данных. Все расписания определены здесь же, в файле bacula-dir.conf. Пример расписания WeeklyCycle приведён ниже, комментировать в нём нечего.

Параметр Where указывает на каталог, где будет развернут бэкап.

Параметр `Write Bootstrap` указывает путь к файлу, в который будет записываться информация, с помощью которой данные могут быть восстановлены из резервной копии без наличия подключения к `Bacula Catalog`. Вместо `%p` будет подставлено значение параметра `Name`:

```
Schedule {
Name = "DailyCycle"
Run = Full daily at 16:10
# Run = Differential 2nd-5th sun at 23:05
Run = Incremental mon-sat at 23:05
}
```

```
Job {
Name = "RestoreFiles"
Type = Restore
Client= bacula-fd
FileSet="Catalog"
Storage = File
Pool = Default
Messages = Standard
Where = /etc2
}
```

Ну, а теперь укажем параметры нашего единственного Агента.

```
Client {
Name = bacula-fd
Address = 11.11.11.23
FDPort = 9102
Catalog = MyCatalog
Password = "1" # password for FileDaemon
File Retention = 30 days # 30 days
Job Retention = 6 months # six months
AutoPrune = yes # Prune expired Jobs/Files
}
```

Последний штрих – закомментируйте все остальные секции: `Job`, `JobDefs`, `Client` и `Console`, на данном этапе они нам не понадобятся. Как вы понимаете, трафик будет идти по тем портам, что указаны в конфигурационных файлах каждого из компонентов `Bacula`.

Далее необходимо настроить доступ к `DD` со стороны `Bacula Console` в файле `/etc/bacula/bconsole.conf` сервера `bakula1.my.dom`:

```
Director {
Name = bacula-dir
DIRport = 9101
address = 11.11.11.21
Password = "1"
}
```

- На машине, где будет `Director Daemon` следует удалить пакеты `bacula-sd` и `bacula-fd`:

```
apt-get remove bacula-sd
apt-get remove bacula-fd
```

Конфиги `bacula-sd` и `bacula-fd` в `/etc/bacula` следует либо переименовать, либо удалить.
Сервисы `bacula-sd` и `bacula-fd` следует остановить:

```
/etc/init.d/bacula-sd stop
/etc/init.d/bacula-fd stop
```

Скопировать готовые конфигурационные файлы [bconsole.conf](#) и [bacula-dir.conf](#) для нашего примера можно [здесь](#).

Настройка Storage Daemon

- Начнём подготовку `Storage Daemon`, который будет отвечать за непосредственную работу с устройством хранения данных. `Bacula` поддерживает широкий спектр устройств, начиная от оптических дисков и заканчивая полнофункциональными ленточными библиотеками. Список поддерживаемых устройств вы сможете найти на официальном сайте проекта. Мы рассмотрим самый распространённый вариант – обычный жёсткий диск с существующей файловой системой (например, `ext3`). Итак, на сервере `bakula2.my.dom` редактируем файл `/etc/bacula/bacula-sd.conf`. В нём находим секцию основных параметров – `Storage`, где определяем параметр `Name`, который задаёт уникальное имя `Storage Daemon`. Остальные параметры можно оставить в значениях по умолчанию.

Секция Director необходима для указания уникального имени DD и пароля, с которым этот DD может подключаться к SD. Секций может быть несколько, что даёт возможность использовать единый сервер хранения данных для нескольких систем резервирования. Все остальные секции Director, найденные в файле, закомментируйте, они пока не понадобятся.

```
Storage { # definition of myself
Name = bacula-sd
SDPort = 9103 # Director's port
WorkingDirectory = "/var/lib/bacula"
Pid Directory = "/var/run/bacula"
Maximum Concurrent Jobs = 20
SDAddress = 11.11.11.22
}

Director {
Name = bacula-dir
Password = "1"
}
```

Но основные настройки, определяющие взаимодействие с устройствами хранения, кроются в секции Device. Рассмотрим параметры, необходимые для хранения резервных копий в рамках существующей файловой системы, подключенной в каталог /back.

Параметр Name определяет уникальное имя подключенного устройства. Если вы планируете создавать изолированные друг от друга резервные копии для каждого из File Daemon, то вам необходимо создать несколько секций Device с уникальными именами. В противном случае резервируемые файлы со всех FD будут размещаться в одном и том же файле, что может несколько затруднить дальнейшее обслуживание системы.

Параметр Media Type определяет произвольное уникальное имя, которое будет использоваться Bacula при восстановлении данных. Согласно ему определяется устройство хранения, с которого будет производиться восстановление. Если вы храните резервные копии в файлах, то для КАЖДОЙ секции Device должен быть задан уникальный Media Type.

Параметр Archive Device указывает путь к файлу устройства в каталоге /dev или путь к каталогу, в котором будут размещаться резервные копии.

Параметр Device Type определяет тип устройства. Для размещения в существующей файловой системе указывается File.

Параметр Random Access указывает на возможность случайной (непоследовательной) адресации. Для файлов указывается Yes.

Параметр RemovableMedia указывает, возможно ли извлечение устройства хранения. Необходимо для ленточных устройств, приводов оптических дисков и т.д. Для файлов устанавливается в значение No.

Параметр LabelMedia указывает на необходимость автоматического маркирования носителей информации.

```
Device {
Name = FileStorage
Media Type = File
Archive Device = /back
LabelMedia = yes; # lets Bacula label unlabeled media
Random Access = Yes;
AutomaticMount = yes; # when device opened, read it
RemovableMedia = no;
AlwaysOpen = no;
}
```

Для базовой настройки этого достаточно

На машине, где будет Storage Daemon следует удалить пакет bacula-fd:

```
apt-get remove bacula-fd
```

Конфиг bacula-fd в /etc/bacula следует либо переименовать, либо удалить.
Сервис bacula-fd следует остановить:

```
/etc/init.d/bacula-fd stop
```

Скопировать готовый конфигурационный файл [bacula-sd.conf](#) для нашего примера можно [здесь](#).

Настройка File Daemon

- Перейдём к настройке File Daemon на рабочей станции bakula3.my.dom.

Для настройки используется файл `/etc/bacula/bacula-fd`, в котором для базовой настройки достаточно лишь определить параметры секции `Director`, где указывается пароль, который будет использовать `DD` при подключении к `FD`, а также секции `FileDaemon`, где указываются настройки `FD`. Все остальные секции `Director`, найденные в файле, прокомментируйте, они пока не понадобятся.

```
Director {
Name = bacula-dir
Password = "1"
}
```

В секции `FileDaemon` нас, на данном этапе, интересует только параметр `Name`, в котором указывается уникальное имя `File Daemon`:

```
FileDaemon { # this is me
Name = bacula-fd
FDport = 9102 # where we listen for the director
WorkingDirectory = /var/lib/bacula
Pid Directory = /var/run/bacula
Maximum Concurrent Jobs = 20
FDAddress = 11.11.11.23
}
```

- На машине, где будет `File Daemon` следует удалить пакет `bacula-sd`:

```
apt-get remove bacula-sd
```

Конфиг `bacula-sd` в `/etc/bacula` следует либо переименовать, либо удалить.
Сервис `bacula-sd` следует остановить:

```
/etc/init.d/bacula-sd stop
```

Скопировать готовый конфигурационный файл `bacula-fd.conf` для нашего примера можно здесь.

- Теперь запустим все компоненты соответствующими командами, данными на соответствующих серверах::

```
/etc/init.d/bacula-director restart
/etc/init.d/bacula-sd restart
/etc/init.d/bacula-fd restart
```

После этого `bacula` будет работать. Управление `bacula` осуществляется через `bconsole`. Настройки каталогов, заданий, расписаний и прочие задаются в конфигурационных файлах.

Для тестовой проверки выполните `bconsole`, далее выполните `run`, выберите `job 1`, войдите в меню набрав `mod`, выберите `1 (Level)`, выберите `1 (Full)`, подтвердите выполнение набрав `yes`. Будет создан бэкап в каталоге `/back` на машине с `Storage Daemon`. Для разворота бэкапа выполните `restore`, выберите пункт `12`, введите номер `job id`, укажите параметр маркировки `mark *`, подтвердите выполнение командой `done`. Будет развернут бэкап в каталоге `/etc2` на машине с `File Daemon`.

Дополнительная информация

Настройка системы резервного копирования `bacula` с описанием функций представлена, например, в следующих руководствах:

- [Системы резервного копирования и управления ИТ-активами предприятия: Часть 4. Установка и настройка Bacula](#)
- [Система бекапирования Bacula](#)
- [Руководство по Ubuntu Server Резервное копирование Bacula](#)