

Свой сервер бэкапирования bareos с возможностью писать на ленточное хранилище

У нас в компании имеется ленточное хранилище, которое, к сожалению, не используется и не приносит никакой пользы. Я планирую настроить сервер резервного копирования на базе Bareos (Vacula) и организовать хранение архивной информации на ленточных носителях. В нашей инфраструктуре используются дисковые полки Infortrend и ленточная библиотека Quantum Scalar i40. Физический сервер будет на CentOS Stream release 9 он оснащён двумя процессорами Intel Xeon E5-2609 v3 с тактовой частотой 1.90 ГГц и 64 ГБ оперативной памяти и два сетевых адаптера для подключения к разным сетям. Также предусмотрен адаптер Fiber Channel для связи с дисковыми полками и ленточным хранилищем. У меня весь процесс начинается с перенастройки Qlogic, что бы нужные WWPN смотрели в нужные устройства. Подробностей по настройке Qlogic давать не буду т.к. это тема для отдельной статьи с подробными разъяснениями.

Приступаем к установке

Для начала обновим пакеты

```
sudo dnf update
```

```
sudo dnf upgrade
```

Устанавливаем и настраиваем Bareos

Для начала пропишем нужные порты в файрвол

```
firewall-cmd --permanent --add-port={80,443}/tcp
```

```
firewall-cmd --permanent --add-port={9101,9102,9103}/tcp
```

```
firewall-cmd --reload
```

Для того что бы корректно обращаться к репозиториям bareos, нужно выполнить команды связанные с сертификатами

```
dnf install ca-certificates
```

update-ca-trust

Теперь отключаем SELinux

В файле

```
nano /etc/selinux/config
```

Прописываем

```
SELINUX=disabled
```

Да, в идеале не нужно отключать SELinux в целях безопасности, а нужно наоборот его настроить корректно, но в моем случае сервер и хранилка не будут смотреть в интернет, а останутся исключительно в одной из выделенных подсетей компании.

Теперь создаем переменную для bareos

```
BAREOSDIST=EL_9
```

Для корректного названия привязываемого к переменной вам лучше воспользоваться ссылкой

И устанавливаем репозиторий одной командой

```
wget https://download.bareos.org/current/${BAREOSDIST}/bareos.repo -O /etc/yum.repos.d/bareos_${BAREOSDIST}.repo
```

Производим установку bareos

```
dnf install bareos bareos-database-postgresql -y
```

Включаем автозапуск сервисов bareos

```
systemctl enable bareos-dir bareos-fd bareos-sd
```

Устанавливаем сервер базы данных

```
dnf install postgresql-server postgresql-contrib
```

Инициализируем базу

```
postgresql-setup initdb
```

Включаем автозапуск и запускаем сервер базы данных

```
systemctl enable postgresql --now
```

Создаем базу с таблицами и нужными привилегиями. Скрипты идут по умолчанию с установленным bareos

```
su - postgres -c /usr/lib/bareos/scripts/create_bareos_database
```

```
su - postgres -c /usr/lib/bareos/scripts/make_bareos_tables
```

```
su - postgres -c /usr/lib/bareos/scripts/grant_bareos_privileges
```

Если вдруг скриптами не получается ничего у вас создать, то придется руками создавать базу и присваивать привилегии.

Запускаем сервисы bareos и проверяем статусы что бы было все в порядке

```
systemctl start bareos-dir bareos-fd bareos-sd
```

```
systemctl status bareos-dir bareos-fd bareos-sd
```

На этом основная часть закончена. Но если в вашей компании как у нас например есть специальный человек, который занимается архивированием, то ему для удобства можно настроить веб интерфейс.

Устанавливаем веб-интерфейс

```
dnf install bareos-webui
```

Редактируем файл

```
nano /etc/php.ini
```

В файле добавляем таймзону

```
date.timezone = "Europe/Moscow"
```

Запускаем веб-сервер

```
systemctl enable httpd --now
```

Теперь создаем учетку для входа в веб интерфейс Редактируем файл

```
nano /etc/bareos/bareos-dir.d/console/admin.conf
```

Содержимое файла

```
Console {  
  
Name = "admin"  
  
Password = "ВАШ СЛОЖНЫЙ ПАРОЛЬ"  
  
Profile = webui-admin  
  
TlsEnable = no  
  
}
```

Пароль должен быть сложным как и любой другой пароль в вашей системе bareos

Перезапускаем сервисы bareos

```
systemctl restart bareos-dir bareos-fd bareos-sd
```

Идем проверять веб интерфейс по адресу **Ваш IP\bareos-webui**

Далее ставим дополнительно компоненты для работы с лентами

```
dnf install bareos-storage-tape mt-st mtx
```

Проверить в наличие ленточных устройств можно несколькими командами

```
ls /dev/tape/by-id/
```

```
lsscsi
```

В моем случае это Quantum с двумя драйвами и авточейнджером

Создаем файл хранения.

```
nano /etc/bareos/bareos-dir.d/storage/Tape.conf
```

С вот таким содержимым

```
Storage {  
  
Name = Tape  
  
Address = bacularestore  
  
Password = "ВАШ СЛОЖНЫЙ ПАРОЛЬ"  
  
Device = autochanger-0  
  
Media Type = LTO  
  
Auto Changer = yes  
  
}
```

Где:

- Name – Имя любое, но лучше использовать Tape что бы было понятно что это ленточное хранилище
- Address – имя вашего сервера, на котором установлен и настроен bareos
- Password – очень сложный пароль для связи устройства с директором. Пароль должен совпадать с паролем из файла /etc/bareos/bareos-sd.d/director/bareos-dir.conf
- Device – Название вашего файла с настройками авточейнджера (робота, который кассеты перекладывает)
- Media Type – Указывается LTO что бы система понимала что это ленты
- Auto Changer – указываем использовать ваш авточейнджер или нет.

Создаем файл авточейнджера

```
nano /etc/bareos/bareos-sd.d/autochanger/autochanger-0.conf
```

С вот таким содержимым

```
Autochanger {  
  
Name = "autochanger-0"
```

```
Changer Device = /dev/tape/by-id/scsi-QUANTUMD0H0112431_LLA  
  
Device = ULTRIUM-HH6-0  
  
Device = ULTRIUM-HH6-1  
  
Changer Command = "/usr/lib/bareos/scripts/mtx-changer %c %o %S %a %d"  
  
}
```

Где:

- Name - должен совпадать со строкой Device из файла /etc/bareos/bareos-dir.d/storage/Tape.conf
- Changer Device - Путь к вашему авточейнджеру, тут может быть вариант как у меня, либо подобный вариант /dev/sg37 где sg будет отражать номер вашего устройства
- Device - это названия ваших драйвов, их может быть два как у меня или один в вашем случае. Эти названия должны совпадать с именем из файла /etc/bareos/bareos-sd.d/device/tapedrive-0.conf
- Changer Comand - путь к скрипту, который будет управлять вашим ленточным хранилищем

Теперь создаем файл устройства

```
nano /etc/bareos/bareos-sd.d/device/tapedrive-0.conf
```

С вот таким содержимым

```
Device {  
  
Name = "ULTRIUM-HH6-0"  
  
Media Type = "LT0"  
  
ArchiveDevice = /dev/tape/by-id/scsi-C38DE40000-nst  
  
LabelMedia = yes  
  
Autochanger = yes  
  
AlwaysOpen = yes  
  
AutomaticMount = yes  
  
RemovableMedia = yes  
  
DeviceType = "Tape"  
  
}
```

```
Device {  
Name = "ULTRIUM-HH6-1"  
Media Type = "LT0"  
ArchiveDevice = /dev/tape/by-id/scsi-C38DE40004-nst  
LabelMedia = yes  
Autochanger = yes  
AlwaysOpen = yes  
AutomaticMount = yes  
RemovableMedia = yes  
DeviceType = "Tape"  
}
```

Где:

- Name – имя устройства, которое должно совпадать с Device из файла /etc/bareos/bareos-sd.d/autochanger/autochanger-0.conf
- ArchiveDevice – Путь к вашему драйву, тут может быть вариант как у меня, либо подобный вариант /dev/sg36 где sg будет отражать номер вашего устройства

После этих настроек у вас уже будет отображаться ленточное хранилище и управление им как через веб интерфейс



так и через bconsole командой

```
list storages
```

Теперь в bconsole командой

```
Status slots
```

Мы можем видеть сколько у нас слотов в ленточном хранилище и какие заняты и свободны



Теперь нужно настроить задания для того что бы на ленту попадали файлы.

Для этого начнем с создания файла в котором будет информация откуда забирать данные для копирования их на ленту

```
nano /etc/bareos/bareos-dir.d/fileset/0nTape.conf
```

Содержимое файла

```
FileSet {  
  
Name = "0nTape"  
  
Include {  
  
Options {  
  
Compression=GZIP  
  
}  
  
# File = "/mnt/backup/Bareos/Конфиги сервера и базы данных/"  
  
}  
  
}
```

Как видно тут указан способ сжатия и путь откуда берем данные (У меня он закомментирован т.к. был тестовый вариант и пока не использую его), ничего сложного

Теперь нужно создать пул для ленты

Для этого создаем файл

```
nano /etc/bareos/bareos-dir.d/pool/LT0.conf
```

Содержимое файла

```
Pool {  
  
Name = "LT0_Pool"  
  
Pool Type = Archive  
  
Storage = Tape  
  
LabelFormat = "<Barcode>"  
  
Maximum Volumes = 500  
  
# VolumeRetention = 365 days  
  
AutoPrune = no  
  
}
```

```
Pool {  
  
Name = "LTO_Pool_OLD"  
  
Pool Type = Archive  
  
Storage = Tape  
  
LabelFormat = "<Barcode>"  
  
Maximum Volumes = 500  
  
# VolumeRetention = 365 days  
  
AutoPrune = no  
  
}
```

Где:

- Name - любое имя удобное для вашего восприятия
- Storage - имя ранее созданного файла /etc/bareos/bareos-dir.d/storage/Tape.conf
- LabelFormat - У меня указан баркод это значит что будут считываться штрихкоды с кассет, если у вас иначе все, то можно оставить LTO- и будет присваиваться номер по порядку

Теперь создаем файл отвечающий за само задание

```
nano /etc/bareos/bareos-dir.d/job/BackupOnTape.conf
```

Содержимое файла

```
Job {  
  
Name = "BackupOnTape"  
  
Type = Backup  
  
FileSet="OnTape"  
  
Schedule="LTO_Migration"  
  
Storage="Tape"  
  
Pool="LTO_Pool"  
  
Messages = Standard  
  
Priority=10
```

```
Client = "bareos-fd"

Write Bootstrap = "/tmp/bareos.bsr"

Backup Format = "Tar"

}
```

Где:

- Name – любое имя, которое вам будет удобно
- FileSet – имя ранее созданного файла /etc/bareos/bareos-dir.d/fileset/OnTape.conf
- Storage – имя ранее созданного файла /etc/bareos/bareos-dir.d/storage/Tape.conf
- Pool – имя из файла /etc/bareos/bareos-dir.d/pool/LTO.conf
- Client – имя вашего клиента взятое из каталога /etc/bareos/bareos-dir.d/client/ т.к. в этом каталоге создаются конфиги клиентов

После всех манипуляций нужно перезагрузить сервисы и проверить их статус работы

```
systemctl restart bareos-dir bareos-fd bareos-sd
```

```
systemctl status bareos-dir bareos-fd bareos-sd
```

Если ошибок не появилось, то вы все сделали верно и уже можете выполнять копирование на ленту в ручном режиме через созданное задание, а если вам необходимо выполнять копирование с помощью планировщика, то нужно будет еще создать отдельные конфиги или отредактировать существующие по пути /etc/bareos/bareos-dir.d/schedule/WeeklyCycle.conf

Дополнения по улучшению (на ваше усмотрение)

Создаем пароль для пользователя, который работает с базой данных Заходим в базу

```
sudo -u postgres psql
```

Меняем пароль для пользователя bareos т.к. по умолчанию он без пароля

```
ALTER USER bareos WITH PASSWORD 'ВАШ ПАРОЛЬ';
```

Выходим из базы

```
\q
```

Теперь заносим этот пароль в файл

```
/etc/bareos/bareos-dir.d/catalog/MyCatalog.conf
```

Еще необходимо изменить тип аутентификации в файле

```
nano /var/lib/pgsql/data/pg_hba.conf
```

Вносим изменения таким образом. Тут выставлен md5 вместо ident и peer

```
local all all md5
host all all 127.0.0.1/32 md5
host all all ::1/128 md5
```

И перезагружаем сервисы bareos и postgresql, после этого обязательно проверяем статус что бы все было в порядке

```
systemctl restart bareos-dir bareos-fd bareos-sd postgresql
```

```
systemctl status bareos-dir bareos-fd bareos-sd postgresql
```

Установка клиентов и настройка клиентов на сервере

Установка клиентов — процесс несложный, но если делать всё руками, то можно очень долго просидеть, создавая нужные конфиги и прописывая в них данные. Поэтому я разработал два скрипта на bash, которые выполняют всё за вас.

1. Первый скрипт запускаете на сервере, где у вас установлен директор.
2. Второй скрипт запускаете на самом клиенте.

Клиентский скрипт сделан таким образом, что система понимает вашу версию Ubuntu и Centos и ничего лишнего не скачает и не установит. Само собой, скрипты я разрабатывал под свои нужды, если вам что-то нужно исправить в них, то вы можете свободно это делать, поэтому обязательно прочитайте, что там написано, прежде чем выполнять их.

Содержимое серверного скрипта

```
#!/bin/bash
# Запрос имени клиента, каталога, адреса и пароля
read -p "Введите название каталога для монтирования расположенного в
//10.0.5.4/backup (например, 'Grafana_Loki'): " mount_name
read -p "Введите имя клиента (например, dokuwiki): " client_name
read -p "Введите адрес клиента (например, 10.0.0.115): " client_address
read -s -p "Введите сложный пароль клиента не менее 30 символов: "
client_password
echo
# Формирование имени без суффикса "-fd"
client_base_name="${client_name%-fd}"
# Отдельные переменные для пулов и storage
storage_name="$client_base_name" # Имя storage будет такое же как и имя клиента
без -fd
device_name="${client_base_name}Storage" # Имя устройства
pool_name="I_${client_name}"
```

```
full_pool_name="F_${client_name}"
differential_pool_name="D_${client_name}"
# Создание клиента в конфигурационном файле
client_config_file="/etc/bareos/bareos-dir.d/client/${client_name}-fd.conf"
cat <<EOL > "$client_config_file"
Client {
Name = "${client_name}-fd"
Address = "${client_address}"
Password = "${client_password}"
}
EOL
# Проверка успешности создания файла клиента
if [ $? -ne 0 ]; then
echo "Ошибка при создании конфигурационного файла клиента."
exit 1
fi
echo "Клиент ${client_name}-fd успешно добавлен в конфигурацию."
# Создание файла <Имя клиента>Default.conf в каталоге /etc/bareos/bareos-
dir.d/jobdefs/
jobdefs_file="/etc/bareos/bareos-dir.d/jobdefs/${client_name}Default.conf"
cat <<EOL > "$jobdefs_file"
JobDefs {
Name = "${client_name}Default"
Type = Backup
Level = Incremental
Client = "${client_name}-fd"
FileSet = "LinuxAll" # ${client_name} fileset
Schedule = "WeeklyCycle"
Storage = ${storage_name}
Messages = Standard
Pool = ${pool_name}
Priority = 10
Write Bootstrap = "/var/lib/bareos/%c.bsr"
Full Backup Pool = $full_pool_name # write Full Backups into "Full" Pool
Differential Backup Pool = $differential_pool_name # write Diff Backups into
"Differential" Pool
Incremental Backup Pool = $pool_name # write Incr Backups into "Incremental"
Pool
Backup Format = "Tar"
}
EOL
# Создание файла backup-<Имя клиента>.conf в каталоге /etc/bareos/bareos-
dir.d/job/
backup_file="/etc/bareos/bareos-dir.d/job/backup-${client_name}-fd.conf"
cat <<EOL > "$backup_file"
Job {
Name = "backup-${client_name}-fd"
JobDefs = "${client_name}Default"
Client = "${client_name}-fd"
}
EOL
```

```
# Создание пулов в каталоге /etc/bareos/bareos-dir.d/pool/
pool_files=("$pool_name" "$full_pool_name" "$differential_pool_name")
for pool in "${pool_files[@]"; do
pool_file="/etc/bareos/bareos-dir.d/pool/${pool}.conf"
if [[ $pool == I_* ]]; then
pool_type="Incremental"
label_format="I_${client_name}-"
retention="30 days"
volume_bytes="1G"
elif [[ $pool == D_* ]]; then
pool_type="Differential"
label_format="D_${client_name}-"
retention="90 days"
volume_bytes="10G"
elif [[ $pool == F_* ]]; then
pool_type="Full"
label_format="F_${client_name}-"
retention="365 days"
volume_bytes="50G"
fi
cat <<EOL > "$pool_file"
Pool {
Name = "${pool}"
Pool Type = Backup
Recycle = yes # Bareos может автоматически перерабатывать тома
AutoPrune = yes # Удалять тома с истекшим сроком действия
Volume Retention = $retention # Как долго следует хранить резервные копии
${pool_type} ?
Maximum Volume Bytes = $volume_bytes # Ограничить размер тома до разумного уровня
Maximum Volumes = 100 # Ограничить количество томов в пуле
Label Format = "$label_format" # Тома будут помечены как "${pool_type}-
<volume-id>"
}
EOL
done
# Создание Storage в каталоге /etc/bareos/bareos-dir.d/storage/
storage_file="/etc/bareos/bareos-dir.d/storage/${storage_name}.conf"
cat <<EOL > "$storage_file"
Storage {
Name = "${storage_name}"
Address = "ВАШ hostname" # Адрес хранилища замените на ваш (по сути это hostname
где у вас установлен bareos-dir)
Password = "ВАШ СЛОЖНЫЙ ПАРОЛЬ 30 и более символов" # Замените на ваш пароль
Device = "${device_name}"
Media Type = "File"
}
EOL
# Создание Device в каталоге /etc/bareos/bareos-sd.d/device/
device_file="/etc/bareos/bareos-sd.d/device/${device_name}.conf"
archive_device="/mnt/backup/${client_base_name}" # Путь к хранилищу, имя
клиента без суффикса "-fd"
```

```
# Проверка существования директории
if [ ! -d "$archive_device" ]; then
echo "Директория $archive_device не существует. Создание..."
sudo mkdir -p "$archive_device" # Создание директории, если она не существует
else
echo "Директория $archive_device уже существует."
fi
cat <<EOL > "$device_file"
Device {
Name = "${device_name}"
Media Type = "File"
Archive Device = "${archive_device}"
LabelMedia = yes; # позволяет Bareos помечать немаркированные носители
Random Access = yes;
AutomaticMount = yes; # когда устройство будет открыто, прочтите его
RemovableMedia = no;
AlwaysOpen = no;
Description = "Файловое устройство. Подключаемый директор должен иметь то же имя и
тип носителя."
}
EOL
# Проверка успешности создания файлов storage и device
if [ $? -ne 0 ]; then
echo "Ошибка при создании конфигураций хранилища или устройства."
exit 1
fi
echo "Хранилище и устройство успешно созданы."
# Изменение владельца файлов
# Каталог, который нужно изменить
target_dir="/etc/bareos/bareos-dir.d/pool/"
# Проверка существования каталога
if [ ! -d "$target_dir" ]; then
echo "Каталог $target_dir не существует."
exit 1
fi
# Поиск и изменение владельца, группы и прав на файлы
for file in "$target_dir"*; do
if [ -e "$file" ]; then # Проверка, существует ли файл
# Получаем текущих владельца и группу
owner=$(stat -c '%U' "$file")
group=$(stat -c '%G' "$file")
# Проверка, является ли владелец и группа root
if [ "$owner" == "root" ] && [ "$group" == "root" ]; then
echo "Изменение владельца и группы для файла: $file"
sudo chown bareos:bareos "$file"
sudo chmod 750 "$file"
fi
fi
done
echo "Готово! Все необходимые файлы были изменены."
# Автомонтирование папок
```

```
# Формирование строки для fstab
remote_path="//10.0.0.175/backup/${mount_name}" # Укажите ваше удаленное
хранилище
local_path="/mnt/backup/${client_name}"
options="cifs username=bareos,password=ВАШ ПАРОЛЬ,uid=ВАШ UID,gid=ВАШ
GID,file_mode=0775,dir_mode=0775 0 0" # Укажите ваши данные для монтирования
каталога
# Обработка пробелов только в первой части
remote_path_escaped=$(echo "$remote_path" | sed 's/ /\ /g')
# Формирование окончательной строки для fstab
mount_entry="${remote_path_escaped} ${local_path} ${options}"
# Проверка, существует ли уже такая строка в fstab
if grep -q -F "$mount_entry" /etc/fstab; then
echo "Строка для монтирования уже существует в /etc/fstab."
else
# Добавление строки в /etc/fstab
echo "$mount_entry" | sudo tee -a /etc/fstab > /dev/null
# Проверка успешности добавления
if [ $? -eq 0 ]; then
echo "Строка для монтирования успешно добавлена в /etc/fstab."
# Перезапуск fstab и монтирование всех файловых систем
echo "Перезапускаем /etc/fstab..."
sudo mount -a
# Проверка успешности монтирования
if [ $? -eq 0 ]; then
echo "Все файловые системы успешно смонтированы."
else
echo "Ошибка при монтировании файловых систем."
fi
else
echo "Ошибка при добавлении строки в /etc/fstab."
fi
fi
# Вывод информации о завершении
echo "Конфигурация для клиента ${client_name} успешно завершена!"
echo "Пожалуйста, проверьте созданные файлы:"
echo "Клиент: $client_config_file"
echo "JobDefs: $jobdefs_file"
echo "Backup Job: $backup_file"
echo "Pools: "
for pool in "${pool_files[@]}"; do
echo " - /etc/bareos/bareos-dir.d/pool/${pool}.conf"
done
echo "Storage: $storage_file"
echo "Device: $device_file"
# Сервисы для перезагрузки
services=("bareos-dir" "bareos-sd" "bareos-fd")
# Перезагрузка сервисов
for service in "${services[@]}"; do
echo "Перезагрузка сервиса $service..."
sudo systemctl restart "$service"
```

```
# Проверка успешности перезагрузки
if [ $? -eq 0 ]; then
echo "Сервис $service успешно перезагружен."
else
echo "Ошибка при перезагрузке сервиса $service."
fi
done
echo ""
# Вывод статусов сервисов
for service in "${services[@]}"; do
echo "Статус сервиса $service:"
sudo systemctl status "$service" --no-pager
echo ""
done
# Конец скрипта
```

Содержимое клиентского скрипта

```
#!/bin/bash
# Определяем переменную DIST в зависимости от дистрибутива
if grep -q -i "ubuntu" /etc/os-release; then
DIST="xUbuntu_$(lsb_release -rs | cut -d. -f1,2)"
elif grep -q -i "centos" /etc/os-release || grep -q -i "fedora" /etc/os-
release; then
DIST="EL_$(grep -oP '(?<=release )\d+' /etc/redhat-release | head -1)"
else
echo "Unsupported operating system."
exit 1
fi
# Скачиваем настройки для репозитория
if [[ $DIST == xUbuntu* ]]; then
wget https://download.bareos.org/current/${DIST}/bareos.list -O
/etc/apt/sources.list.d/bareos_${DIST}.list
# Устанавливаем пакет для шифрования и Дешифровки цифровых подписей
apt install -y gnupg2
# Устанавливаем ключ подписи пакетов
wget -q https://download.bareos.org/current/${DIST}/Release.key -O- | apt-
key add -
# Обновляем список пакетов
apt update
# Устанавливаем пакет для bareos-fd
apt install -y bareos-filedaemon
# Разрешаем автозапуск сервиса
systemctl enable bareos-filedaemon
# Настраиваем брандмауэр
iptables -I INPUT -p tcp --dport 9102 -j ACCEPT
apt-get install -y iptables-persistent
netfilter-persistent save
elif [[ $DIST == EL* ]]; then
```

```
wget https://download.bareos.org/current/${DIST}/bareos.repo -O
/etc/yum.repos.d/bareos_${DIST}.repo
# Устанавливаем пакет для bareos-fd
yum install -y bareos-fd
# Разрешаем автозапуск сервиса и стартуем его
systemctl enable bareos-fd --now
# Настраиваем брандмауэр
firewall-cmd --permanent --add-port=9102/tcp
firewall-cmd --reload
else
echo "Unsupported operating system."
exit 1
fi
# Проверяем доступность сервера
ping -c 3 ВАШ СЕРВЕР ГДЕ УСТАНОВЛЕН ДИРЕКТОР
if [ $? -ne 0 ]; then
echo "10.0.0.111 ВАШ СЕРВЕР ГДЕ УСТАНОВЛЕН ДИРЕКТОР" >> /etc/hosts
ping -c 3 ВАШ СЕРВЕР ГДЕ УСТАНОВЛЕН ДИРЕКТОР
if [ $? -ne 0 ]; then
echo "Server ВАШ СЕРВЕР ГДЕ УСТАНОВЛЕН ДИРЕКТОР is still not reachable after
adding to /etc/hosts."
exit 1
fi
fi
# Настройка файла /etc/bareos/bareos-fd.d/director/bareos-dir.conf
cat <<EOL > /etc/bareos/bareos-fd.d/director/bareos-dir.conf
Director {
Name = bareos-dir
Password = "ПАРОЛЬ НУЖНО ВЗЯТЬ ИЗ СОЗДАННОГО КЛИЕНТА НА СЕРВЕРЕ
/etc/bareos/bareos-dir.d/client/"
Description = "Allow the configured Director to access this file daemon."
}
EOL
# Перезапускаем bareos-fd и выводим его статус
systemctl restart bareos-fd
systemctl status bareos-fd --no-pager
echo "Bareos installation and configuration completed successfully."
echo "Замените пароль вот тут /etc/bareos/bareos-fd.d/director/bareos-dir.conf"
```

Таким образом, после выполнения скриптов в вашем веб-интерфейсе появятся созданные клиенты, готовые к работе. Мы завершили это увлекательное путешествие в мир резервного копирования, где каждый шаг приближал нас к созданию надежной инфраструктуры для защиты данных. Я надеюсь, эта информация была для вас полезной и вдохновила на новые свершения. Спасибо, что были со мной на протяжении всей статьи! Не забывайте следить за обновлениями на моем канале, где я продолжаю делиться ценными знаниями и практическими рекомендациями в области технологий. Удачи вам в ваших проектах и до новых встреч!

From:

<https://wiki.fellk.ru/dokuwiki/> - **Игорь FELLK**

Permanent link:

<https://wiki.fellk.ru/dokuwiki/doku.php/manuals/bareos1to?rev=1726146773>

Last update: **2024/09/12 13:12**

