

Создание всех настроек для клиента Bareos на стороне Director

Подробнее о Bareos [тут](#)

```
#!/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
```

Bareos

From:

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

Permanent link:

<https://wiki.fellk.ru/dokuwiki/doku.php/scripts/createclientsbareosrv>

Last update: **2024/09/19 14:10**

