Если вы видите что-то необычное, просто сообщите мне.

restic

Инструкция по настройке restic с MinIO

Шаг 1: Установка restic

Ubuntu/Debian:

```
sudo apt update
sudo apt install restic -y
```

или взять тута: https://github.com/restic/restic/releases

```
wget https://github.com/restic/restic/releases/download/v0.3.3/restic_linux_amd64.bz2
bzip2 -d restic_linux_amd64.bz2
chmod +x restic_linux_amd64
mv restic_linux_amd64 /usr/local/bin/restic
```

Проверка установки:

```
restic version
```

Шаг 2: Создание пароля для репозитория

Сгенерируйте надежный пароль для шифрования данных:

```
# Вариант 1: сгенерировать случайный пароль
openssl rand -base64 32

# Вариант 2: использовать apg (если установлен)
sudo apt install apg -y
apg -a 1 -m 32 -n 1 -M NCL
```

Важно: Запишите этот пароль в надежном месте! Без него восстановить данные будет невозможно.

Шаг 3: Настройка подключения к MinIO

3.1 Формат подключения

Для MinIO URL репозитория имеет формат:

```
s3:http://<IP_или_хост>:<порт>/<имя_бакета>
```

Примеры:

- Если MinIO запущен локально: `s3:http://localhost:9000/restic-backups`
- Если MinIO на другом сервере: `s3:http://192.168.1.100:9000/my-bucket`
- Если используется HTTPS: `s3:https://minio.example.com:9000/backups`

3.2 Создание файла с переменными окружения

Создайте файл для хранения учетных данных:

```
sudo nano /etc/restic.env
```

Добавьте следующие строки (замените значения на свои):

```
# Учетные данные MinIO
export AWS_ACCESS_KEY_ID="<ваш_minio_access_key>"
export AWS_SECRET_ACCESS_KEY="<ваш_minio_secret_key>"

# URL репозитория в формате S3
export RESTIC_REPOSITORY="s3:http://<адрес_minio>:<порт>/<имя_бакета>"

# Пароль для шифрования репозитория
export RESTIC_PASSWORD="<ваш_сгенерированный_пароль>"

# Опционально: отключение проверки SSL (только для http)
export AWS_DISABLE_SSL=true
```

Пример реальной конфигурации:

```
export AWS_ACCESS_KEY_ID="minioadmin"
export AWS_SECRET_ACCESS_KEY="minioadmin123"
export RESTIC_REPOSITORY="s3:http://192.168.1.50:9000/backups"
export RESTIC_PASSWORD="MyVeryStrongPassword123!@#"
export AWS_DISABLE_SSL=true
```

3.3 Защита файла с учетными данными

```
sudo chmod 600 /etc/restic.env
```

3.4 Загрузка переменных окружения

```
source /etc/restic.env
```

Шаг 4: Проверка подключения и инициализация репозитория

4.1 Проверка доступа к MinIO

Проверьте, что restic видит ваш MinIO:

```
restic version
```

Если команда выполняется без ошибок, можно переходить к инициализации.

4.2 Инициализация репозитория

Выполните инициализацию репозитория (требуется только один раз):

```
restic init
```

При успешной инициализации вы увидите сообщение:

```
created restic repository <уникальный_идентификатор> at s3:http://...
```

4.3 Проверка, что бакет в MinIO создан

Убедитесь, что в MinIO появилась папка `restic-repo` или аналогичная с файлами репозитория.

Шаг 5: Создание тестового бэкапа

5.1 Тестовый бэкап одного файла

```
# Создайте тестовый файл
echo "Тестовый бэкап от $(date)" > ~/test-backup.txt

# Выполните бэкап
restic backup ~/test-backup.txt

# Просмотр созданных снимков
restic snapshots
```

5.2 Бэкап нескольких директорий

```
restic backup /etc /home /var/www --tag "test-backup"
```

Шаг 6: Создание скрипта для автоматического бэкапа

6.1 Создание скрипта

```
sudo nano /usr/local/bin/backup-minio.sh
```

```
#!/bin/bash

# Загрузка переменных окружения
```

```
source /etc/restic.env

# Переменные для логирования
LOG_FILE="/var/log/restic-backup.log"
DATE=$(date '+%Y-%m-%d %H:%M:%S')

echo "=== Backup started at $DATE ===" >> $LOG_FILE

# Выполнение бэкапа
# Добавьте нужные директории для бэкапа
restic backup /etc /home /var/www \
    --tag "daily-backup" \
    --exclude="*.tmp" \
    --exclude="*.log" \
    >> $LOG_FILE 2>&1

# Проверка статуса
if [ $? -eq 0 ]; then
    echo "Backup completed successfully" >> $LOG_FILE
else
    echo "Backup FAILED" >> $LOG_FILE
fi

# Очистка старых снимков (оставить последние 7)
restic forget --keep-last 7 --prune >> $LOG_FILE 2>&1

# Проверка целостности репозитория (опционально, выполняется раз в неделю)
# if [ $(date +%u) -eq 7 ]; then
#     restic check >> $LOG_FILE 2>&1
# fi

echo "=== Backup finished at $(date '+%Y-%m-%d %H:%M:%S') ===" >> $LOG_FILE
echo "" >> $LOG_FILE
```

6.2 Настройка прав и тестирование

```
# Сделать скрипт исполняемым
sudo chmod +x /usr/local/bin/backup-minio.sh
```

```
# Запустить скрипт вручную для тестирования
sudo /usr/local/bin/backup-minio.sh

# Проверить лог
cat /var/log/restic-backup.log
```

Шаг 7: Настройка автоматического запуска (cron)

7.1 Добавление в crontab

Откройте crontab для редактирования:

```
sudo crontab -e
```

Добавьте строки для автоматического запуска:

```
# Ежедневный бэкап в 2:00 ночи
0 2 * * * /usr/local/bin/backup-minio.sh

# Проверка репозитория каждое воскресенье в 3:00
0 3 * * 0 restic --repository-file /etc/restic.env check
```

7.2 Альтернатива: использование systemd таймера

Создайте службу для systemd:

```
sudo nano /etc/systemd/system/restic-backup.service
```

```
[Unit]
Description=Restic backup to MinIO
```

```
After=network.target

[Service]
Type=oneshot
EnvironmentFile=/etc/restic.env
ExecStart=/usr/local/bin/backup-minio.sh
User=root
```

Создайте таймер:

```
sudo nano /etc/systemd/system/restic-backup.timer
```

```
[Unit]
Description=Run restic backup daily

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

Активируйте таймер:

```
sudo systemctl daemon-reload
sudo systemctl enable restic-backup.timer
sudo systemctl start restic-backup.timer
```

Шаг 8: Полезные команды для работы

Команда	Описание
`restic snapshots`	Показать все снимки (бэкапы)
`restic snapshots --latest 1`	Показать последний снимок

Команда	Описание
`restic stats`	Показать статистику репозитория
`restic check`	Проверить целостность репозитория
`restic restore <snapshot_ID> --target /путь/восстановления`	Восстановить данные
`restic forget --keep-last 10 --prune`	Оставить последние 10 снимков, остальные удалить
`restic diff <snapshot1> <snapshot2>`	Сравнить два снимка

Шаг 9: Восстановление данных

9.1 Найти нужный снимок

```
source /etc/restic.env
restic snapshots
```

Пример вывода:

```
ID        Time                 Host        Tags        Paths
---------------------------------------------------------------------------
a1b2c3d4  2024-03-21 02:00:01  myserver    daily-backup /etc /home /var/www
e5f6g7h8  2024-03-20 02:00:01  myserver    daily-backup /etc /home /var/www
```

9.2 Восстановление данных

```
# Восстановить весь снимок
restic restore a1b2c3d4 --target /tmp/restore

# Восстановить только конкретную директорию
restic restore a1b2c3d4 --target /tmp/restore --include "/etc/nginx"

# Восстановить конкретный файл
restic restore a1b2c3d4 --target /tmp/restore --include "/etc/nginx/nginx.conf"
```

Шаг 10: Оптимизация и лучшие практики

10.1 Исключение ненужных файлов

Создайте файл исключений:

```
sudo nano /etc/restic-excludes.txt
```

Добавьте паттерны исключения:

```
*.tmp
*.swp
*.log
/proc/*
/sys/*
/dev/*
/tmp/*
*.cache
*.pid
```

Используйте его в скрипте:

```
restic backup / --exclude-file=/etc/restic-excludes.txt
```

10.2 Настройка политики хранения

Вместо `--keep-last 7` можно использовать более гибкую политику:

```
# Хранить: 7 последних, 4 еженедельных, 6 ежемесячных
restic forget --keep-last 7 --keep-weekly 4 --keep-monthly 6 --prune
```

10.3 Мониторинг и уведомления

Добавьте в скрипт отправку уведомлений:

```
# Отправка в Telegram (пример)
if [ $? -eq 0 ]; then
    curl -s "https://api.telegram.org/bot<TOKEN>/sendMessage" \
        -d "chat_id=<CHAT_ID>" \
        -d "text=✅ Backup completed successfully for $(hostname)"
else
    curl -s "https://api.telegram.org/bot<TOKEN>/sendMessage" \
        -d "chat_id=<CHAT_ID>" \
        -d "text=❌ Backup FAILED for $(hostname)"
fi
```

Устранение неполадок

Проблема: "repository does not exist"

Решение: Выполните инициализацию:

```
restic init
```

Проблема: "AccessDenied"

Решение: Проверьте учетные данные MinIO:

```
# Проверка доступности MinIO
curl http://<адрес_minio>:<порт>/minio/health/live
```

Проблема: "connection refused"

Решение: Проверьте, что MinIO запущен и доступен по указанному адресу и порту.

Проблема: "wrong password"

Решение: Убедитесь, что переменная `RESTIC_PASSWORD` загружена корректно:

```
echo $RESTIC_PASSWORD
```

Проблема: "invalid repository"

Решение: Проверьте правильность URL репозитория:

```
echo $RESTIC_REPOSITORY
# Должно быть: s3:http://ip:port/имя-бакета
```

Проверка работоспособности

Выполните полный цикл проверки:

```
# 1. Загрузка переменных
source /etc/restic.env

# 2. Инициализация (если еще не делали)
restic init

# 3. Создание тестового бэкапа
echo "Test backup $(date)" > /tmp/test.txt
restic backup /tmp/test.txt

# 4. Просмотр снимков
restic snapshots

# 5. Получение ID последнего снимка
LAST_SNAPSHOT=$(restic snapshots --latest 1 --json | grep -o '"short_id":"[^"]*"' | head -1 |
cut -d'"' -f4)

# 6. Восстановление в другую директорию
```

```
restic restore $LAST_SNAPSHOT --target /tmp/restore-test

# 7. Проверка восстановленного файла
ls -la /tmp/restore-test/

# 8. Удаление тестовых данных
rm -rf /tmp/restore-test /tmp/test.txt
```

Готово! Restic настроен для работы с вашим MinIO. Регулярно проверяйте логи и тестируйте восстановление, чтобы убедиться в надежности бэкапов.

Revision #4
Created 2026-03-21 15:44:14 UTC by gasick
Updated 2026-07-22 17:12:05 UTC by gasick