

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

restic

????????????? ?? ??????????????

restic ? MinIO

??? 1: ?????????????? restic

Ubuntu/Debian:

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

Проверка установки:

```
restic version
```

??? 2: ?????????????? ?????????? ????

?????????????????

Сгенерируйте надежный пароль для шифрования данных:

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

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

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

???

MinIO

3.1

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

????

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 ??????? ?????? ? ?????????? ??????????

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

3.4 ?????????? ?????????????? ??????????????

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

??? 4: ?????????? ?????????????????? ? ????????????????????? ??????????????????

4.1 ?????????? ?????????? ? MinIO

Проверьте, что restic видит ваш MinIO:

```
restic version
```

Если команда выполняется без ошибок, можно переходить к инициализации.

4.2 ?????????????????? ??????????????????

6.1 ?????????? ??????????

```
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 ?????????? ????? ? ?????????????????

```
# 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
```

??? 7: ?????????????? ?????????????????????? ????????? (cron)

7.1 ?????????????? ? 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 ??????????????????: ?????????????????? systemd ??????????

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
```

??? 8: ?????????? ?????????? ??? ????????

Команда	Описание
<code>restic snapshots</code>	Показать все снимки (бэкапы)

Команда	Описание
<code>restic snapshots --latest 1</code>	Показать последний снимок
<code>restic stats</code>	Показать статистику репозитория
<code>restic check</code>	Проверить целостность репозитория
<code>restic restore <snapshot_ID> --target /путь/восстановления</code>	Восстановить данные
<code>restic forget --keep-last 10 --prune</code>	Оставить последние 10 снимков, остальные удалить
<code>restic diff <snapshot1> <snapshot2></code>	Сравнить два снимка

??? 9: ?????????????????? ???????

9.1 ?????? ??????? ????????

```
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 ?????????????????? ????????

```
# 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"
```

??? 10: ?????????????? ? ????????

??????????

10.1 ?????????????? ??????????? ??????????

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 ?????????????? ?????????????? ??????????????

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 ?????????????????? ? ?????????????????????

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
```

?????????? ????????????

?????????: "repository does not exist"

Решение: Выполните инициализацию:

```
restic init
```

?????????: "AccessDenied"

Решение: Проверьте учетные данные MinIO:

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

?????????: "connection refused"

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

?????????: "wrong password"

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

```
echo $RESTIC_PASSWORD
```

?????????: "invalid repository"

Решение: Проверьте правильность URL репозитория:

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

????????? ??????????????????????????????

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 #2

Created 2026-03-21 15:44:14 UTC by gasick

Updated 2026-03-21 15:44:42 UTC by gasick