

Title: Kopie zapasowe (backupy) systemu eDokumenty w Linux

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Others/LinuxBackupConfiguration

Version: 26

Date: 05/05/26 23:19:56

## Table of Contents

*Kopie zapasowe (backupy) systemu eDokumenty w Linux*

3

## Kopie zapasowe (backupy) systemu eDokumenty w Linux

Poniższy artykuł przedstawia

1. Backup bazy danych
2. Backup files
3. Backup public\_html
4. Backup repos
5. Backup repository
6. Backup public\_ftp oraz users - Opcjonalnie
7. Backup na zewnętrznym nośniku

Skrypty backupów umieszczamy w katalogu /etc/cron.daily

### 1. Backup bazy danych

```
#!/bin/bash

# Tworzy zrzut bazy danych 5 razy w tygodniu (uruchamiany z crona)
# od 1-pon do 5-pia
#

WDAY=$(date +%w)
DST="/mnt/backup/db/${WDAY}"

#[ -d $DST ] || umask 007 && mkdir $DST

if [ "$WDAY" -lt 1 -o "$WDAY" -gt 6 ]
then
    exit 1
fi

su - postgres -c "pg_dump -f ${DST}/edokumenty.sql -d edokumenty"
gzip -f ${DST}/edokumenty.sql

Dwie powyższe linie można zastąpić jedną i wykonać zrzut bezpośrednio do pliku gz
su - postgres -c "pg_dump -d edokumenty | gzip > ${DST}/edokumenty.sql.gz"
```

Modyfikacja skryptu backupu usuwanie plików np starszych niż 60 dni

```
#!/bin/bash

# Tworzy zrzut bazy danych i usuwa starsze niz 60 dni

DATE=$(date +"%Y%m%d%H%M")
DST="/home/edokumenty/backup/db/"

#Zrzut bazy
su - postgres -c "pg_dump -f ${DST}/edokumenty_${DATE}.sql -d edokumenty"
gzip -f ${DST}/edokumenty_${DATE}.sql

#Ununięcie plików starszych niż 60dni
find /home/edokumenty/backup/db -name "edokumenty*.sql.gz" -mtime +60 -type f -delete
```

### 2. Backup files

```
#!/bin/bash
DST="/mnt/backup/files"

if [ -d $DST ]
then
    sleep 1
else
    mkdir $DST
fi

rsync -a /home/edokumenty/files/ /mnt/backup/files/
```

### 3. Backup public\_html

W przypadku tworzenia kopii kodu systemu eDokumenty należy dodać skrypt czyszczący zawartość plików tymczasowych

```
rm -rf /home/edokumenty/public_html/apps/edokumenty/var/tmp/*
```

```
#!/bin/bash
cd /home/edokumenty/
tar -pczf public_html.tar.gz public_html/
mv public_html.tar.gz /mnt/backup/system
```

### 4. Backup repos

```
#!/bin/bash
svnadmin dump /home/edokumenty/repos > /mnt/backup/repos/repos.dump
```

### 5. Backup repository

Poniższy skrypt wykonuje kopię plików z katalogu repository - plików wersjonowanych

```
#!/bin/bash
DST="/mnt/backup/repository"

if [ -d $DST ]
then
    sleep 1
else
    mkdir $DST
fi

rsync -a /home/edokumenty/repository/ /mnt/backup/repository/
```

### 6. Backup public\_ftp oraz users - Opcjonalnie

Poniższy skrypt wykonuje kopię plików z katalogu public\_ftp oraz users

```
#!/bin/bash
DST="/mnt/backup/user_files"

if [ -d $DST ]
then
    sleep 1
else
    mkdir $DST
fi

rsync -a /home/edokumenty/public_ftp/ /mnt/backup/user_files/
rsync -a /home/edokumenty/users/ /mnt/backup/user_files/
```

## 7. Backup na zewnętrznym nośniku

Instalacja pakietu

```
apt-get install lftp
```

Poniższy skrypt wykonuje kopię danych z serwera eDokumentów na dysk zewnętrzny

```
FTP_HOST=[ADRES_IP_SERVERA]:21
FTP_USER=[LOGIN]
FTP_PWD=[HASŁO]
LOCAL_SOURCE_PATH=/home/edokumenty/backup/${1} #param 1 from script for example
REMOTE_DEST_PATH=/edokumenty

EXCLUDE_FILES_LIST=".svn empty anotherDirectory"

# Construct excludes
for exclude in ${EXCLUDE_FILES_LIST}; do
EXCLUDES="-x ${exclude} ${EXCLUDES}"
done

# -R, --reverse reverse mirror (put files)
# mirror -e
#--delete delete files not present at remote site
lftp ftp://${FTP_USER}:${FTP_PWD}@${FTP_HOST} -e "mirror ${EXCLUDES} -R ${LOCAL_SOURCE_PATH} ${REMOTE_DEST_PATH} ; quit"
```

FTP\_HOST - adres serwera na którym mają zostać zapisane pliki

FTP\_USER - login użytkownika który ma dostęp do serwera plików

FTP\_PWD - hasło użytkownika

LOCAL\_SOURCE\_PATH - lokalizacja z której będzie wykonywany backup plików

REMOTE\_DEST\_PATH - lokalizacja do której będzie wykonywany backup plików

W przypadku gdy serwera na który będziemy wysłać dane odrzuci połączenie z powodu braku zgodności certyfikatu ostatnia linijka kodu powinna wyglądać następująco

```
lftp ftp://${FTP_USER}:${FTP_PWD}@${FTP_HOST} -e "set ftp:ssl-allow no; mirror ${EXCLUDES} -R ${LOCAL_SOURCE_PATH} ${REMOTE_DEST_PATH} ; quit"
```

Drugim programem do przeniesienia backupów na zewnętrzny nośnik jest curlftpfs. Instalujemy pakiety

```
apt-get install curlftpfs
```

Tworzymy punkt montowania

```
mkdir /mnt/my_ftp
```

Kolejnym krokiem jest pomontowanie zasobu

```
curlftpfs ftp-user:ftp-pass@my-ftp-location.local /mnt/my_ftp/
```

- użytkownik: ftp-user
- hasło: ftp-pass
- host/IP: my-ftp-location.local

Możemy także podmontować zasób automatycznie w fstab. W tym celu tworzymy plik /root/.netrc

```
machine my-ftp-location.local
login ftp-user
password ftp-pass
```

oraz ustawiamy uprawnienia 600

```
chmod 600 /root/.netrc
```

w fstab dodajemy wpis

```
curlftps#my-ftp-location.local /mnt/my_ftp fuse allow_other,uid=1000,gid=1000,umask=0022 0 0
```

a następnie montujemy zasób

```
mount -a
```

W przypadku błędu podczas wywołania skryptu "bin/bash zły interpreter nie ma takiego pliku ani katalogu" - wchodzimy w edycję pliku ze skryptem za pomocą edytora vim i zmieniamy format

```
:set fileformat=unix
```