

```
nano backup.sh
```

```
-----
```

```
#!/bin/bash
```

```
# --- AYARLAR ---
```

```
DB_USER="root"
```

```
DB_PASS='dddddddddddd'
```

```
BACKUP_DIR="/root/backups"
```

```
EMAIL="mehmetdddddddddd@gmail.com"
```

```
DATE=$(date +"%Y-%m-%d_%H-%M")
```

```
# Yedeklenecek Veritabanları Listesi (İstemediklerinizi buradan silebilirsiniz)
```

```
# Not: System ve Plesk DB'lerini (psa, mysql, horde vb.) genelde dosya sistemi yedeğiyle almak daha güvenlidir ama buraya ekleyebilirsiniz.
```

```
DATABASES=(
```

```
    "mkl"
```

```
    "nots"
```

```
    "prj"
```

```
)
```

```
# Klasör yoksa oluştur
```

```
mkdir -p "$BACKUP_DIR"
```

```
# E-posta eklerini tutacak dizi
```

```
ATTACHMENTS=()
```

```
ZIP_FILES=()
```

```
# --- YEDEKLEME DÖNGÜSÜ ---
```

```
echo "--- Yedekleme Başlıyor: $DATE ---"
```

```
for DB_NAME in "${DATABASES[@]}"; do

    FILE_NAME="${DB_NAME}_${DATE}.sql"

    ZIP_NAME="${DB_NAME}_${DATE}.zip"


    echo "Processing: $DB_NAME"


    # 1. MySQL Dump Al

    # --skip-lock-tables opsiyonu bazen kilitli tablolarda hatayı önler

    mysqldump -u "$DB_USER" -p"$DB_PASS" --skip-lock-tables "$DB_NAME" >
"$BACKUP_DIR/$FILE_NAME"


    if [ $? -ne 0 ]; then

        echo "❌ HATA: $DB_NAME yedeği alınamadı."

        continue # Hata varsa bu veritabanını geç, diğerine devam et

    fi


    # 2. Zip Oluştur (-j ile klasör yolunu gizle)

    zip -j -q "$BACKUP_DIR/$ZIP_NAME" "$BACKUP_DIR/$FILE_NAME"


    if [ $? -eq 0 ]; then

        # Başarılıysa eklenecekler listesine koy

        ATTACHMENTS+=("-a" "$BACKUP_DIR/$ZIP_NAME")

        ZIP_FILES+=("$BACKUP_DIR/$ZIP_NAME")

        # SQL dosyasını hemen sil (Zip kaldı)

        rm -f "$BACKUP_DIR/$FILE_NAME"

    else

        echo "❌ HATA: Zip oluşturulamadı ($DB_NAME)"

    fi

done
```

```
# --- E-POSTA GÖNDERİMİ ---
```

```
if [ ${#ATTACHMENTS[@]} -eq 0 ]; then
```

```
    echo "❌ Gönderilecek başarılı yedek bulunamadı."
```

```
    exit 1
```

```
fi
```

```
echo "Tüm veritabanı yedekleri ekte. ($DATE)" | mail -s "Multi DB Backup $DATE"  
"${ATTACHMENTS[@]}" "$EMAIL"
```

```
if [ $? -eq 0 ]; then
```

```
    echo "✅ E-posta başarıyla gönderildi."
```

```
    # Gönderim başarılıysa Zip dosyalarını temizle
```

```
    rm -f "${ZIP_FILES[@]}"
```

```
else
```

```
    echo "❌ E-posta gönderilemedi! Logları kontrol edin: tail /var/log/msmtp.log"
```

```
fi
```

```
chmod +x /root/backup.sh
```

```
crontab -e
```

Eğer bu yedeğin her gün gece 03:00'te otomatik alınmasını isterseniz:

```
0 3 * * * /root/backup.sh > /dev/null 2>&1
```

1) msmtplib ve mailutils kur

dnf install msmtplib s-nail -y

2) Gmail için uygulama şifresi oluştur

Gmail → Güvenlik → App Passwords (Uygulama Şifreleri)

Yeni bir şifre oluştur → 16 haneli şifre verecek.

Bu şifreyi aşağıdaki ayara yazacağız.

3) msmtplib yapılandırma dosyasını oluştur

defaults

auth on

tls on

tls_trust_file /etc/ssl/certs/ca-bundle.crt

logfile /var/log/msmtplib.log

account gmail

host smtp.gmail.com

port 587

from mehmetiddddd@gmail.com

user mehmetiddddd@gmail.com

password UYGULAMA_SIFRESI_BURAYA

account default : gmail

Kaydet → CTRL+O → ENTER → CTRL+X

4) Dosya izinlerini düzelt

```
chmod 600 /etc/msmtprc
```

```
ln -sf /usr/bin/msmtp /usr/sbin/sendmail
```

```
ln -sf /usr/bin/msmtp /usr/bin/sendmail
```

5) Test maili gönder

```
echo "test" | mail -s "test mail" mehmetddddd@gmail.com
```

msmtp logunu kontrol et

```
tail -f /var/log/msmtp.log
```

```
#!/bin/bash
```

```
# --- AYARLAR ---
```

```
DB_USER="888888888"
```

```
DB_PASS='8888888888'
```

```
BACKUP_DIR="/root/backups"
```

```
EMAIL="mehmet888888@gmail.com"
```

```
DATE=$(date +"%Y-%m-%d_%H-%M")
```

```
# Yedeklenecek Veritabanları Listesi (İstemediklerinizi buradan silebilirsiniz)
```

```
# Not: System ve Plesk DB'lerini (psa, mysql, horde vb.) genelde dosya sistemi yedeğiyle almak daha güvenlidir ama buraya ekleyebilirsiniz.
```

```
DATABASES=(
```

```
"apsc"
```

```
"blog"
```

```
)
```

```
# Klasör yoksa oluştur
```

```
mkdir -p "$BACKUP_DIR"
```

```
# E-posta eklerini tutacak dizi
```

```
ATTACHMENTS=()
```

```
ZIP_FILES=()
```

```
# --- YEDEKLEME DÖNGÜSÜ ---
```

```
echo "--- Yedekleme Başlıyor: $DATE ---"
```

```
for DB_NAME in "${DATABASES[@]"; do
```

```
    FILE_NAME="${DB_NAME}_${DATE}.sql"
```

```
    ZIP_NAME="${DB_NAME}_${DATE}.zip"
```

```
    echo "Processing: $DB_NAME"
```

```
# 1. MySQL Dump Al
```

```
# --skip-lock-tables opsiyonu bazen kilitli tablolarda hatayı önler
```

```
mysqldump -u "$DB_USER" -p"$DB_PASS" --skip-lock-tables "$DB_NAME" >  
"$BACKUP_DIR/$FILE_NAME"
```

```
if [ $? -ne 0 ]; then
```

```
    echo "❌ HATA: $DB_NAME yedeği alınamadı."
```

```
    continue # Hata varsa bu veritabanını geç, diğerine devam et
```

```
fi
```

2. Zip Oluştur (-j ile klasör yolunu gizle)

```
zip -j -q "$BACKUP_DIR/$ZIP_NAME" "$BACKUP_DIR/$FILE_NAME"
```

```
if [ $? -eq 0 ]; then
```

```
    # Başarılıysa eklenecekler listesine koy
```

```
    ATTACHMENTS+=("-a" "$BACKUP_DIR/$ZIP_NAME")
```

```
    ZIP_FILES+=("$BACKUP_DIR/$ZIP_NAME")
```

```
    # SQL dosyasını hemen sil (Zip kaldı)
```

```
    rm -f "$BACKUP_DIR/$FILE_NAME"
```

```
else
```

```
    echo "❌ HATA: Zip oluşturulamadı ($DB_NAME)"
```

```
fi
```

```
done
```

--- E-POSTA GÖNDERİMİ ---

```
if [ ${#ATTACHMENTS[@]} -eq 0 ]; then
```

```
    echo "❌ Gönderilecek başarılı yedek bulunamadı."
```

```
    exit 1
```

```
fi
```

```
echo "Tüm veritabanı yedekleri ektedir. ($DATE)" | mail -s "Multi DB Backup $DATE"  
"${ATTACHMENTS[@]}" "$EMAIL"
```

```
if [ $? -eq 0 ]; then
```

```
    echo "✅ E-posta başarıyla gönderildi."
```

```
    # Gönderim başarılıysa Zip dosyalarını temizle
```

```
    rm -f "${ZIP_FILES[@]}"
```

```
else
```

```
    echo "❌ E-posta gönderilemedi! Logları kontrol edin: tail /var/log/msmtp.log"
```

```
fi
```

