

MariaDB

MariaDB ist eine von der Community entwickelte und kommerziell unterstützte Abspaltung des relationalen Datenbankmanagementsystems (RDBMS) MySQL, die als freie und quelloffene Software unter der GNU General Public License weitergeführt werden soll. Die Entwicklung wird von einigen der ursprünglichen Entwickler von MySQL geleitet, die es aufgrund von Bedenken wegen der Übernahme durch die Oracle Corporation im Jahr 2009 abgezweigt haben.

MariaDB soll eine hohe Kompatibilität mit MySQL aufrechterhalten, mit Bibliotheks-Binärparität und exakter Übereinstimmung mit MySQL-APIs und -Befehlen, so dass es in vielen Fällen als direkter Ersatz für MySQL fungieren kann. Die neuen Funktionen weichen jedoch voneinander ab. Sie umfasst neue Speicher-Engines wie Aria, ColumnStore und MyRocks.

Der leitende Entwickler/CTO ist Michael "Monty" Widenius, einer der Gründer von MySQL AB und der Gründer von Monty Program AB. Am 16. Januar 2008 gab MySQL AB bekannt, dass das Unternehmen einer Übernahme durch Sun Microsystems für rund 1 Milliarde US-Dollar zugestimmt hat. Die Übernahme wurde am 26. Februar 2008 abgeschlossen. Sun wurde im darauffolgenden Jahr von der Oracle Corporation übernommen. MariaDB ist nach der jüngeren Tochter von Widenius, Maria, benannt. (MySQL ist nach seiner anderen Tochter My benannt.)

Paket

```
pacman -S mariadb
```

Datenverzeichnis anlegen

```
mariadb-install-db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

Start

```
systemctl enable --now mariadb.service
```

Verbesserung der Sicherheit

Gebe das Passwort für den mysql-Root-Benutzer ein, und die meisten Fragen können mit Ja beantwortet werden, wenn du zum Beispiel keinen Fernzugriff brauchst.

```
mysql_secure_installation
```

Lokalen Benutzer anlegen

```
mysql -u root -p
```

```
CREATE USER 'user_name'@'localhost' IDENTIFIED BY 'mypassword';  
FLUSH PRIVILEGES;  
EXIT;
```

Remote Benutzer erstellen

```
mysql -u root -p
```

```
CREATE USER 'user_name'@'%' IDENTIFIED BY 'mypassword';  
FLUSH PRIVILEGES;  
EXIT;
```

Datenbank erstellen

```
mysql -u root -p
```

```
CREATE DATABASE db_name;  
GRANT ALL ON db_name.* to 'user_name'@'localhost' IDENTIFIED BY 'password'  
WITH GRANT OPTION;  
FLUSH PRIVILEGES;  
\q
```

oder etwas Ausführlicheres

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, INDEX, DROP, ALTER, CREATE  
TEMPORARY TABLES, LOCK TABLES ON db_name_here.* TO 'myuser'@'localhost';
```

Datenbank anzeigen und löschen

```
mysql -u root -p  
SHOW DATABASES;  
DROP DATABASE db_name;  
FLUSH PRIVILEGES;  
EXIT;
```

Benutzer anzeigen und löschen

```
mysql -u root -p
```

```
SELECT User FROM mysql.user;
DROP USER user_name@localhost;
FLUSH PRIVILEGES;
EXIT;
```

Passwort festlegen oder ändern

```
SET PASSWORD FOR 'user_name'@'localhost' = PASSWORD('password');
FLUSH PRIVILEGES;
EXIT;
```

Datenbank sichern

Du kannst alle Datenbanken mit dem folgenden Befehl `mysqldump -u root -p --all-databases | gzip > all_databases.sql.gz` sichern oder nur eine einzige Datenbank mit `mysqldump -u root -p --databases database_name | gzip > database_name.sql.gz`.

Oder verwende das folgende Skript, das Folgendes enthält:

- getrennte Datenbanken (wenn du die Datei extrahierst)
- Festlegen, nach wie vielen Tagen die alten Backups gelöscht werden sollen
- E-Mail-Benachrichtigung

```
#!/bin/bash
# Shell script to backup MySQL database

# Set these variables
MyUSER="my_user"           # DB_USERNAME
MyPASS="mypassword"        # DB_PASSWORD
MyHOST="localhost"         # DB_HOSTNAME

# Backup Dest directory
DEST="/path/to/backup/mysql/"

# Email for notifications
# EMAIL="email"
# How many days old files must be to be removed
DAYS="60"

# Linux bin paths
MYSQL="$(which mysql)"
MYSQLDUMP="$(which mysqldump)"
GZIP="$(which gzip)"

# Get date in dd-mm-yyyy format
NOW="$(date +"%d-%m-%Y_%H-%M")"

# Create Backup sub-directories
```

```
MBD="$DEST/$NOW/mysql"
install -d $MBD

# DB skip list
SKIP="information_schema
performance_schema"

# Get all databases
DBS="$($MYSQL -h $MyHOST -u $MyUSER -p$MyPASS -Bse 'show databases')"
```

```
# Archive database dumps
for db in $DBS
do
    skipdb=-1
    if [ "$SKIP" != "" ];
    then
        for i in $SKIP
        do
            [ "$db" == "$i" ] && skipdb=1 || :
        done
    fi

    if [ "$skipdb" == "-1" ] ; then
        FILE="$MBD/$db.sql"
        $MYSQLDUMP -h $MyHOST -u $MyUSER -p$MyPASS $db > $FILE
    fi
done

# Archive the directory, send mail and cleanup
cd $DEST
tar -cf $NOW.tar $NOW
$GZIP -9 $NOW.tar

#echo -e "Subject: MySQL backup $(hostname)\r\n\r\nMySQL backup is
completed! Backup name is $NOW.tar.gz" | msmtpt -a default $EMAIL
rm -rf $NOW

# Remove old files
find $DEST -mtime +$DAYS -exec rm -f {} \;
```

Use [crontab](#) for automation.

Restore database

```
mysql -u root -p db_name < database.sql
```

Überprüfung/Aktualisierung der Datenbank

Bei einer neuen Hauptversion von Mariadb (z.B. mariadb-10.7.4-1 auf mariadb-10.8.3-1) ist es ratsam, die Datenbanken zu aktualisieren:

1. den Datenbank-Daemon 10.7.4-1 laufen lassen
2. das Paket aktualisieren
3. führe `mysql_upgrade` (von der neuen Paketversion) gegen die alte, noch laufenden Daemon aus. Dies wird einige Fehlermeldungen erzeugen, aber das Upgrade wird erfolgreich sein
4. den Daemon neu starten, damit der 10.8.3-Daemon läuft

```
systemctl restart mariadb.service
```

```
mariadb-upgrade -u root -p
```

Um alle Tabellen in allen Datenbanken zu prüfen:

```
mariadb-check --all-databases -u root -p -c
```

Um alle Tabellen in allen Datenbanken zu analysieren:

```
mariadb-check --all-databases -u root -p -a
```

Um alle Tabellen in allen Datenbanken zu reparieren:

```
mariadb-check --all-databases -u root -p -r
```

Um alle Tabellen in allen Datenbanken zu optimieren:

```
mariadb-check --all-databases -u root -p -o
```

Alle Befehle "Datenbank prüfen/aktualisieren" in einem Block

```
systemctl restart mariadb.service
mariadb-upgrade -u root -p
mariadb-check --all-databases -u root -p -c
mariadb-check --all-databases -u root -p -a
mariadb-check --all-databases -u root -p -r
mariadb-check --all-databases -u root -p -o
```

From:

<http://wiki.techsaviours.org/> - **Your Digital Privacy DIY Solutions | TECH SAVIOURS .ORG**

Permanent link:

<http://wiki.techsaviours.org/de/server/services/mariadb>

Last update: **2023/02/14 21:44**



