MariaDB

MariaDB is a community-developed, commercially supported fork of the MySQL relational database management system (RDBMS), intended to remain free and open-source software under the GNU General Public License. Development is led by some of the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation in 2009.

MariaDB is intended to maintain high compatibility with MySQL, with library binary parity and exact matching with MySQL APIs and commands, allowing it in many cases to function as drop-in replacement for MySQL. However, new features are diverging. It includes new storage engines like Aria, ColumnStore, and MyRocks.

Its lead developer/CTO is Michael “Monty” Widenius, one of the founders of MySQL AB and the founder of Monty Program AB. On 16 January 2008, MySQL AB announced that it had agreed to be acquired by Sun Microsystems for approximately $1 billion. The acquisition completed on 26 February 2008. Sun was then bought the following year by Oracle Corporation. MariaDB is named after Widenius’ younger daughter, Maria. (MySQL is named after his other daughter, My.)

Package

```
pacman -S mariadb
```

Create data dir

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

Start

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

Improve initial security

Set the password for the mysql root user, and most questions can be answered yes, for example, if you do not need remote access.

```
mysql_secure_installation
```

Create local user

```
mysql -u root -p
```
CREATE USER 'user_name'@'localhost' IDENTIFIED BY 'mypassword';
FLUSH PRIVILEGES;
EXIT;

Create remote user

mysql -u root -p

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

Create database

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

or something more detailed

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

Show and delete database

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

Show and delete user

mysql -u root -p
SELECT User FROM mysql.user;
DROP USER user_name@localhost;
FLUSH PRIVILEGES;
EXIT;
Set or change password

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

Backup database

You can dump all databases with the following command:
`mysqldump --single-transaction --flush-logs --master-data=2 --all-databases -u root -p | gzip > all_databases.sql.gz`

Or use the following script, which contains:

- separate databases (when you extract the file)
- define after how many days the old backups should be deleted
- email notification

```bash
#!/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="$(which mysql)"
MYSQLEDUMP="$(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"
```
# Get all databases

```
DBS="\($\text{MYSQL} -h \text{MyHOST} -u \text{MyUSER} -p\text{MyPASS} -Bse 'show databases')" 
```

# Archive database dumps

```
for \( \text{db} \) in $\text{DBS}
do
    \( \text{skipdb}=-1 \)
    if [ "\$\text{SKIP}" != "" ]; then
        for i in $\text{SKIP}
do
            [ "$\text{db}" == "$\text{i}" ] && \text{skipdb}=1 || :
done
fi

if [ "$\text{skipdb}" == "-1" ]; then
    \( \text{FILE}=\$\text{MBD/\text{db}.sql} \)
    \text{MYSQLDUMP} -h \text{MyHOST} -u \text{MyUSER} -p\text{MyPASS} \text{db} > $\text{FILE}
fi
done
```

# Archive the directory, send mail and cleanup

```
cd $\text{DEST}
tar -cf $\text{NOW.tar} $\text{NOW}
$\text{GZIP} -9 $\text{NOW.tar}
```

```
#echo -e "\text{Subject: MySQL backup }$(\text{hostname})\\nMySQL backup is completed! Backup name is $\text{NOW.tar.gz}" | msmtp -a default $\text{EMAIL}
rm -rf $\text{NOW}
```

# Remove old files

```
find $\text{DEST} -mtime +$\text{DAYS} -exec rm -f {} \;
```

Use **crontab** for automation.

## Restore database

```
\text{mysql} -u root -p \text{db \_name} < \text{database.sql}
```

## Check/update database

Upon a major version release of mariadb (for example mariadb-10.7.4-1 to mariadb-10.8.3-1), it is wise to upgrade databases:

1. keep the 10.7.4-1 database daemon running
2. upgrade the package
3. run `mysql_upgrade` (from the new package version) against the old still-running daemon. This will produce some error messages; however, the upgrade will succeed.
4. restart the daemon, so the 10.8.3 daemon runs.

```bash
mysql_upgrade -u root -p --skip-version-check
systemctl restart mariadb.service
```

To check all tables in all databases:

```bash
mysqlcheck --all-databases -u root -p -c
```

To analyze all tables in all databases:

```bash
mysqlcheck --all-databases -u root -p -a
```

To repair all tables in all databases:

```bash
mysqlcheck --all-databases -u root -p -r
```

To optimize all tables in all databases:

```bash
mysqlcheck --all-databases -u root -p -o
```

All “Check/update database” commands in one block

```bash
mysql_upgrade -u root -p --skip-version-check
systemctl restart mariadb.service
mysqlcheck --all-databases -u root -p -c
mysqlcheck --all-databases -u root -p -a
mysqlcheck --all-databases -u root -p -r
mysqlcheck --all-databases -u root -p -o
```

From:

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

Permanent link:


Last update: 2022/06/04 00:36