MySQL

From ArchWiki

MySQL is a widely spread, multi-threaded, multi-user SQL database. For more information about features, see the official homepage (http://www.mysql.com/) .

Note: MariaDB is now officially Arch Linux default implementation of MySQL. It is recommended for all users to upgrade to MariaDB. Oracle MySQL was dropped to the AUR. See the announcement (https://www.archlinux.org/news/mariadb-replaces-mysql-in-repositories/) .

Contents

- 1 Installation
  - 1.1 Enable at startup
  - 1.2 Upgrade from Oracle MySQL to MariaDB
  - 1.3 On update
- 2 Configuration
  - 2.1 Disable remote access
  - 2.2 Enable auto-completion
  - 2.3 Using UTF-8
  - 2.4 Using a TMPFS for tmpdir
- 3 Backup
- 4 Troubleshooting
  - 4.1 MySQL daemon cannot start
  - 4.2 Unable to run mysql_upgrade because MySQL cannot start
  - 4.3 Reset the root password
  - 4.4 Check and repair all tables
  - 4.5 Optimize all tables
- 5 See also

Installation

The MySQL implementation chosen by Arch Linux is called MariaDB (https://mariadb.org/) . Install `mariadb (https://www.archlinux.org/packages/?name=mariadb)` from the official repositories. Alternative implementations are:

- **Oracle MySQL** — Oracle official implementation.

  https://www.mysql.com/ || `mysql (https://aur.archlinux.org/packages/mysql/)`

- **Percona Server** — Alternative which offers breakthrough performance, scalability, features, and instrumentation.

http://www.percona.com/software/percona-server/ || `percona-server` (https://www.archlinux.org/packages/?name=percona-server)

Tip: If the database (in `/var/lib/mysql`) resides in a btrfs filesystem you should consider disabling Copy-on-Write for the directory before creating any database:

```
# chattr +C /var/lib/mysql
```

Start the `mysqld` daemon, run the setup script:

```
# mysql_secure_installation
```

and restart the daemon afterwards.

Frontends available are `mysql-gui-tools` (https://aur.archlinux.org/packages/mysql-gui-tools/) and `mysql-workbench` (https://aur.archlinux.org/packages/mysql-workbench/).

Enable at startup

To start the MySQL daemon at boot, enable `mysqld` systemd service.

Upgrade from Oracle MySQL to MariaDB

Note: It could be needed to remove the following files from `/var/lib/mysql`: `ib_logfile0`, `ib_logfile1` and `aria_log_control` before restarting the daemon in the following procedure.

Users who want to switch will need to stop their current `mysqld` daemon, install *mariadb*, *libmariadbclient* or *mariadb-clients*, restart `mysqld` and execute:

```
# mysql_upgrade -p
```

in order to migrate their systems.

On update

You might consider running this command after you have upgraded MySQL and started it:

```
# mysql_upgrade -u root -p
```

Configuration

Once you have started the MySQL server, you probably want to add a root account in order to maintain your MySQL users and databases. This can be done manually or automatically, as mentioned by the output of the above script. Either run the commands to set a password for the root account, or run the secure installation script.

You now should be able to do further configuration using your favorite interface. For example you can use MySQL's command line tool to log in as root into your MySQL server:

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

Disable remote access

The MySQL server is accessible from the network by default. If MySQL is only needed for the localhost, you can improve security by not listening on TCP port 3306. To refuse remote connections, uncomment the following line in `/etc/mysql/my.cnf`:

```
skip-networking
```

You will still be able to log in from the localhost.

Enable auto-completion

Note: Enabling this feature can make the client initialization longer.

The MySQL client completion feature is disabled by default. To enable it system-wide edit `/etc/mysql/my.cnf`, and replace `no-auto-rehash` by `auto-rehash`. Completion will be enabled next time you run the MySQL client.

Using UTF-8

In the `/etc/mysql/my.cnf` file section under the `mysqld` group, add:

```
[mysqld]
init_connect                = 'SET collation_connection = utf8_general_ci,NAMES utf8'
collation_server            = utf8_general_ci
character_set_client        = utf8
character_set_server        = utf8
```

Using a TMPFS for tmpdir

The directory used by MySQL for storing temporary files is named *tmpdir*. For example, it is used to perform disk based large sorts, as well as for internal and explicit temporary tables.

Create the directory with appropriate permissions:

```
# mkdir -pv /var/lib/mysqltmp
# chown mysql:mysql /var/lib/mysqltmp
```

Find the id and gid of the `mysql` user and group:

```
$ id mysql
uid=27(mysql) gid=27(mysql) groups=27(mysql)
```

Add to your `/etc/fstab` file.

```
tmpfs   /var/lib/mysqltmp   tmpfs   rw,gid=27,uid=27,size=100m,mode=0750,noatime   0 0
```

Add to your `/etc/mysql/my.cnf` file under the `mysqld` group:

```
tmpdir      = /var/lib/mysqltmp
```

Then reboot or (shutdown mysql, mount the tmpdir, start mysql).

Backup

The database can be dumped to a file for easy backup. The following shell script will do this for you, creating a `db_backup.gz` file in the same directory as the script, containing your database dump:

```
#!/bin/bash

THISDIR=$(dirname $(readlink -f "$0"))

mysqldump --single-transaction --flush-logs --master-data=2 --all-databases \
 | gzip > $THISDIR/db_backup.gz
echo 'purge master logs before date_sub(now(), interval 7 day);' | mysql
```

See also the official `mysqldump` page (http://dev.mysql.com/doc/refman/5.6/en/mysqldump.html) in the MySQL manual.

Troubleshooting

MySQL daemon cannot start

If MySQL fails to start and there is no entry in the log files, you might want to check the permissions of files in the directories `/var/lib/mysql` and `/var/lib/mysql/mysql` . If the owner of files in these directories is not `mysql:mysql` , you should do the following:

```
# chown mysql:mysql /var/lib/mysql -R
```

If you run into permission problems despite having followed the above, ensure that your `my.cnf` is copied to `/etc/` :

```
# cp /etc/mysql/my.cnf /etc/my.cnf
```

Now try and start the daemon.

If you get these messages in your `/var/lib/mysql/hostname.err` :

```
[ERROR] Can't start server : Bind on unix socket: Permission denied
[ERROR] Do you already have another mysqld server running on socket: /var/run/mysqld/mysqld.sock
```

```
[ERROR] Aborting
```

the permissions of `/var/run/mysqld` could be the culprit.

```
# chown mysql:mysql /var/run/mysqld -R
```

If you run mysqld and the following error appears:

```
Fatal error: Can't open and lock privilege tables: Table 'mysql.host' doesn't exist
```

Run the following command from the `/usr` directory to install the default tables:

```
# cd /usr
# mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Unable to run mysql_upgrade because MySQL cannot start

Try run MySQL in safemode:

```
# mysqld_safe --datadir=/var/lib/mysql/
```

And then run:

```
# mysql_upgrade -u root -p
```

Reset the root password

Stop the `mysqld` daemon. Issue the following command:

```
# mysqld_safe --skip-grant-tables &
```

Connect to the mysql server. Issue the following command:

```
# mysql -u root mysql
```

Change root password:

```
mysql> UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Start the `mysqld` daemon.

Check and repair all tables

Check and auto repair all tables in all databases, see more (http://dev.mysql.com/doc/refman/5.7/en/mysqlcheck.html) :

```
# mysqlcheck -A --auto-repair -u root -p
```

Optimize all tables

Forcefully optimize all tables, automatically fixing table errors that may come up.

```
# mysqlcheck -A --auto-repair -f -o -u root -p
```

See also

- LAMP - Arch wiki article covering the setup of a LAMP server (Linux Apache MySQL PHP)
- PhpMyAdmin - Arch wiki article covering the web-based tool to help manage MySQL databases using an Apache/PHP frontend.
- PHP - Archi wiki article on PHP.
- MySQL Performance Tuning Scripts and Know-How (http://www.askapache.com/mysql/performance-tuning-mysql.html)

Retrieved from "https://wiki.archlinux.org/index.php?title=MySQL&oldid=280298"
Category: Database management systems
