



# How to write your first patch ?

Anel Husaković anel@mariadb.org

> MariaDB Unconference New York 21-22 February 2019

#### **Getting started**

#### Agenda:

- 1. Get the source code
- 2. Configure the build
- 3. Compile
- 4. Testing the server
- 5. Starting mysqld after build
- 6. Write the patch

[1] https://mariadb.org/get-involved/

[2] <u>https://mariadb.org/get-involved/getting-started-for-developers/</u>

[3] <u>https://mariadb.org/get-involved/getting-started-for-developers/get-code-build-test/</u>

[4] <u>https://mariadb.org/get-involved/getting-started-for-developers/writing-good-test-cases-mariadb-server/</u>

[5] <u>https://mariadb.org/get-involved/getting-started-for-developers/submitting-pull-request/</u>





#### **Get the source code**

Starting from 5.5.42 (19.02.2015) source code has been moved to github <u>https://github.com/MariaDB/server/</u>

• Fork the repo and clone the forked directory git clone <a href="https://github.com/an31/server">https://github.com/an31/server</a>

Result:

- Remote repository as a fork of Mariadb server
- Local "server" repository with `10.4` as a current branch and that only remote is origin

anel@ubuntu:~/workspace/mypresentation/server\$ git branch
\* 10.4

anel@ubuntu:~/workspace/mypresentation/server\$ git remote -v
origin https://github.com/an3l/server (fetch)
origin https://github.com/an3l/server (push)

## **Configure the build**

On Debian based distribution build dependencies may be needed

- apt-get build-dep mysql-server
- apt-get install libgnutls28-dev # This may or may-not be already installed

To generate the *Makefiles* used to compile the server **cmake** is used.

Make sure to clean everything before you started and in case if needed run the update of submodules (example libmariadb or upstream submodule rocksdb, wsrep-lib).

- cmake . -DCMAKE\_BUILD\_TYPE=Debug
- cmake . -DCMAKE\_BUILD\_TYPE=Debug -G Ninja
- cmake . -DCONC\_WITH\_{UNITTEST,SSL}=OFF -DWITH\_EMBEDDED\_SERVER=OFF -DWITH\_UNIT\_TESTS=OFF -DCMAKE\_BUILD\_TYPE=Debug -DPLUGIN\_{TOKUDB,MROONGA,OQGRAPH,ROCKSDB,CONNECT,PERFSCHEMA,SPIDER}=NO -DWITH\_SAFEMALLOC=OFF -DWITH\_SSL=bundled -G Ninja # 2277 vs 1229 files
- $\operatorname{ccmake}$  . # To see the list of flags

Result:

- Generated MakeFiles
- CMakeFiles/CMakeOutput.log CMakeFiles/CMakeError.log
- cmake/build\_configuration/mysql\_release.cmake

#### **Compile and test the server**

- make #or ninja

As a result of previous step *cmake/build\_configuration/mysql\_release.cmake* script is generated which will be invoked by calling the make/ninja.

Result:

- Executable files (sql/mysqld, client/mysql)
- Navigate to mysql-test folder and run the mysql test with mysql-test-run (mtr) . /mtr mysql
- Torun all tests ./mtr --parallel=5 --mem --force --max-test-fail=0
- Try to see the server version ./sql/mysqld -V

anel@ubuntu:~/workspace/mypresentation/server\$ ./sql/mysqld -v 2019-02-19 6:52:55 0 [Note] ./sql/mysqld (mysqld 10.4.3-MariaDB-debug) starting as process 83398 ... 2019-02-19 6:52:55 0 [Warning] Could not increase number of max\_open\_files to more than 1024 (request: 4186) 2019-02-19 6:52:55 0 [Warning] Changed limits: max\_open\_files: 1024 max\_connections: 151 (was 151) table\_cache: 421 (was 2000) 2019-02-19 6:52:55 0 [ERROR] Can't find messagefile '/usr/local/mysql/share/errmsg.sys' 2019-02-19 6:52:55 0 [ERROR] Aborting

# Start mysqld after build

When running MariaDB for first time one need to install system tables by running *mysql\_install\_db* script. This will create default directory ./data in the source directory:

./scripts/mysql\_install\_db --srcdir=.

Note: If you already have specified .my.cnf file yo may get information:

mysql.user table already exists!

Run mysql\_upgrade, not mysql\_install\_db

It is possible to specify your own data directory.

- 1. Create a directory for your data (*/mydatadir*).
- 2. Run the script *mysql\_install\_db*

./scripts/mysql\_install\_db --srcdir=. --datadir=/path/to/data/dir Now in order to start mysqld run:

```
./sql/mysqld --datadir=./data --lc_messages_dir=./sql/share
```

Alternative you can create configuration file in ~/.my.cnf and run just ./sql/mysqld or use --defaults-file option to create custom conf file and call

```
./sql/mysqld --defaults-file=/path/myfile.cnf
```

```
To run the client ./client/mysql
```

```
To get help use \ensuremath{\mathsf{mysqld}}\xspace -v --help
```

anel@ubuntu:~/workspace/mypresentation/server\$ cat ~/.my.cnf # The Mariadh server group
imariadbl
latadir=/home/anel/workspace/mypresentation/server/data
# path to source dir + sql/share
lc_messages_dir=/home/anel/workspace/mypresentation/server/sql/share
nax-connections=20
.c-messages=en_us



# Write the patch

#### Write the patch

- Add new remote upstream which is used to make sure your fork is up to date so you could sync with it and rebase on top of it.
  - git remote add upstream <a href="https://github.com/MariaDB/server.git">https://github.com/MariaDB/server.git</a>
  - git remote --v # 2 remotes should be present
  - git fetch upstream

```
git branch 10.3 upstream/10.3 # branch '10.3' set up to track remote branch '10.3' from 'upstream'.
```

```
git checkout 10.3 && git rebase upstream/10.3
```

# push rebased branch on your fork: git push origin 10.3

Note: when switching branches one need to clean and compile again everything.

- Create a new branch for your patch git checkout -b mypatch
- Write your patch. On which branch to write the patch? *Rule of thumb:* The earliest one which is affected with.
- Commit messages notes good commit messages

#### Write the patch - mtr

- It is recommended to have a test case for each patch that showcases the wrong behavior.
- ./mysql-test
- ./mysql-test/main ; ./mysql-test/<suite>/t [r] | <test-name.test>
- ./mtr <test-name>
  - ./mtr mysql # options: --mem --force --max-test-fail=0 -suite -embedded --record
- List of mtr's command\_names can be found in ./client/mysqltest.cc
- Where to find tasks? Jira <u>https://jira.mariadb.org</u> (search for task with: labels = beginner-friendly and status !=closed order by updated desc)
- In 10.4 there is ./CONTRIBUTING.md
- Live QA in 2 time slots each week for new contributors.
- Push the patch to github and create pull request (PR).



# Thank you