

# How to Migrate from MySQL to MariaDB

without Tears

September 14-16, 2020  
Sveta Smirnova



# Sveta Smirnova

---



- MySQL Support Engineer
- Author of
  - MySQL Troubleshooting
  - JSON UDF functions
  - FILTER clause for MySQL
- Speaker
  - Percona Live, OOW, Fosdem, DevConf, HighLoad...



PERCONA

# Table of Contents

---

- Initial Setup
- Authentication
- Replication
- SQL Differences
- Functional Differences
- Diagnostics

# MariaDB

---

- Started as fork of MySQL

# MariaDB

---

- Started as fork of MySQL
- Independent product

# MariaDB

---

- Started as fork of MySQL
- Independent product
- Hundreds of unique features

# MariaDB

---

- Started as fork of MySQL
- Independent product
- Hundreds of unique features
- Important improvements are backported
  - from MySQL

# Initial Setup

# Drop-in Replacement

---

- Check version compatibility

MySQL	MariaDB
5.1	5.1, 5.2, 5.3
5.5	5.5
5.6	10.0, 10.1
5.7	10.2, 10.3, 10.4, 10.5
8.0	-

# Drop-in Replacement

---

- Check version compatibility
- Stop MySQL Server

# Drop-in Replacement

---

- Check version compatibility
- Stop MySQL Server
- Install MariaDB

# Drop-in Replacement

---

- Check version compatibility
- Stop MySQL Server
- Install MariaDB
- Point MariaDB to MySQL datadir

# Drop-in Replacement

---

- Check version compatibility
- Stop MySQL Server
- Install MariaDB
- Point MariaDB to MySQL datadir
- Start MariaDB

# Drop-in Replacement

---

- Check version compatibility
- Stop MySQL Server
- Install MariaDB
- Point MariaDB to MySQL datadir
- Start MariaDB
- Run `mariadb-upgrade`

# System Database Changes

---

- Before upgrade

```
$ ls -w100 data/mysql/
columns_priv.frm      gtid_executed.ibd          proc.MYD           slow_log.CSV
columns_priv.MYD       help_category.frm        proc.MYI           slow_log.frm
columns_priv.MYI       help_category.ibd        procs_priv.frm    tables_priv.frm
db.frm                help_keyword.frm         procs_priv.MYD     tables_priv.MYD
db.MYD                help_keyword.ibd        procs_priv.MYI     tables_priv.MYI
db.MYI                help_relation.frm       proxies_priv.frm   time_zone.frm
db.opt                help_relation.ibd        proxies_priv.MYD   time_zone.ibd
engine_cost.frm       help_topic.frm          proxies_priv.MYI   time_zone_leap_second.frm
engine_cost.ibd       help_topic.ibd         server_cost.frm   time_zone_leap_second.ibd
event.frm             innodb_index_stats.frm  server_cost.ibd   time_zone_name.frm
event.MYD              innodb_index_stats.ibd  servers.frm      time_zone_name.ibd
event.MYI              innodb_table_stats.frm servers.ibd      time_zone_transition.frm
func.frm              innodb_table_stats.ibd  slave_master_info.frm
...
...
```



PERCONA

# System Database Changes

---

- After upgrade

```
$ ls -w100 data/mysql/
columns_priv.frm      gtid_executed.ibd          proc.frm           slow_log.frm
columns_priv.MAD       gtid_slave_pos.frm        proc.MAD          tables_priv.frm
columns_priv.MAI       gtid_slave_pos.ibd        proc.MAI          tables_priv.MAD
column_stats.frm       help_category.frm        procs_priv.frm    tables_priv.MAI
column_stats.MAD       help_category.MAD        procs_priv.MAD   table_stats.frm
column_stats.MAI       help_category.MAI        procs_priv.MAI   table_stats.MAD
db.frm                help_keyword.frm         proxies_priv.frm  table_stats.MAI
db.MAD                help_keyword.MAD        proxies_priv.MAD  time_zone.frm
db.MAI                help_keyword.MAI        proxies_priv.MAI  time_zone_leap_second.frm
db.opt                help_relation.frm       roles_mapping.frm time_zone_leap_second.MAD
engine_cost.frm        help_relation.MAD      roles_mapping.MAD time_zone_leap_second.MAI
engine_cost.ibd        help_relation.MAI      roles_mapping.MAI time_zone.MAD
event.frm              help_topic.frm        server_cost.frm  time_zone.MAI
...
...
```

# Differences

---

- Specific MariaDB features
  - Storage engines
  - SQL statements
  - Options
  - Just use them

# Differences

---

- Specific MariaDB features
- Specific MySQL features
  - SQL statements
  - Plugins
  - They are missed

# Differences

---

- Specific MariaDB features
- Specific MySQL features
- Features, present in both products
  - Working differently
  - Our focus today

# Authentication

# Initial Setup

---

- MySQL
  - -initialize
  - mysql\_install\_db deprecated, removed in 8.0

# Initial Setup

---

- MySQL
  - -initialize
  - mysql\_install\_db deprecated, removed in 8.0
- MariaDB
  - Only mysql\_install\_db

# Authentication Plugins

---

- Supported plugins
  - MySQL
    - Native
    - Caching SHA-2 - Default in 8.0
    - SHA-256
    - Cleartext
    - No-login
    - Socket Peer-Credential

# Authentication Plugins

---

- Supported plugins
  - MariaDB

- `mysql_native_password`
- `mysql_old_password`
- SHA-256 - Client only!
- ed25519
- GSSAPI
- PAM
- Named Pipe
- Unix Socket

# Authentication Plugins

---

- Supported plugins
- User management SQL is same

# Authentication Plugins

---

- Supported plugins
- User management SQL is same
- MariaDB supports multiple plugins per account

# Password Validation

---

- MySQL
  - validate\_password plugin
    - Deprecated!
  - validate\_password component

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name          | Value   |
+-----+-----+
| validate_password_check_user_name | ON      |
| validate_password_dictionary_file |         |
| validate_password_length        | 8       |
| validate_password_mixed_case_count | 1       |
| validate_password_number_count  | 1       |
| validate_password_policy        | MEDIUM  |
| validate_password_special_char_count | 1       |
+-----+-----+
```

# Password Validation

---

- MariaDB
  - Password Validation Plugin API
  - Two plugins

- simple\_password\_check

```
MariaDB [performance_schema]> show variables like 'simple_password_check%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| simple_password_check_digits | 1     |
| simple_password_check_letters_same_case | 1     |
| simple_password_check_minimal_length    | 8     |
| simple_password_check_other_characters  | 1     |
+-----+-----+
4 rows in set (0.001 sec)
```

# Password Validation

---

- MariaDB
  - Password Validation Plugin API
  - Two plugins

- cracklib\_password\_check

```
MariaDB [performance_schema]> show variables like 'cracklib%';
+-----+-----+
| Variable_name          | Value           |
+-----+-----+
| cracklib_password_check_dictionary | /var/cache/cracklib/cracklib_dict |
+-----+-----+
1 row in set (0.001 sec)
```

# Password Validation

---

- MySQL
- MariaDB
- Password expiration works same way

# Roles

---

- Syntax is same

# Roles

---

- Syntax is same
- MySQL
  - Details are in Information Schema
    - ADMINISTRABLE\_ROLE\_AUTHORIZATIONS
    - APPLICABLE\_ROLES
    - ENABLED\_ROLES
    - ROLE\_COLUMN\_GRANTS
    - ROLE\_ROUTINE\_GRANTS
    - ROLE\_TABLE\_GRANTS



# Roles

---

- Syntax is same
- MySQL
  - Stored in mysql

- user
- role\_edges

```
mysql> select * from mysql.role_edges\G
***** 1. row *****
    FROM_HOST: %
    FROM_USER: my_role
    TO_HOST: %
    TO_USER: sveta
WITH_ADMIN_OPTION: N
1 row in set (0.00 sec)
```

- default\_roles

# Roles

---

- Syntax is same
- MariaDB
  - Details are in Information Schema
    - APPLICABLE\_ROLES
    - ENABLED\_ROLES

# Roles

---

- Syntax is same
- MariaDB
  - Stored in mysql

- user
- roles\_mapping

```
MariaDB [information_schema]> select * from mysql.roles_mapping\G
***** 1. row *****
      Host: localhost
      User: root
      Role: my_role
Admin_option: Y
***** 2. row *****
      Host: %
      User: sveta
      Role: my_role
Admin_option: N
2 rows in set (0.001 sec)
```



# Roles

---

- Syntax is same
- MariaDB
  - The role automatically available for the grantor

# Replication

# No GTIDs

---

- Compatible
- Can work any way around

# No GTIDs

---

- Compatible
- Can work any way around
- Unless data types are incompatible
- Before 8.0!

# GTIDs

---

- Implementation is different
- MariaDB understands MySQL GTIDs
- MySQL does **not** accept MariaDB GTIDs
- Only replication **from** MySQL is possible!

# Multi-threaded Replication

---

- Different concurrency control

# Multi-threaded Replication

---

- Different concurrency control
- MySQL: slave\_parallel\_workers

# Multi-threaded Replication

---

- Different concurrency control
- MySQL: slave\_parallel\_workers
- MySQL: slave\_parallel\_type=DATABASE | LOGICAL\_CLOCK
- MySQL: binlog\_transaction\_dependency\_tracking = COMMIT\_ORDER | WRITESET | WRITESET\_SESSION

# Multi-threaded Replication

---

- Different concurrency control
- MariaDB: `slave_parallel_threads`

# Multi-threaded Replication

---

- Different concurrency control
- MariaDB: slave\_parallel\_threads
- MariaDB: slave\_parallel\_max\_queued

# Multi-threaded Replication

---

- Different concurrency control
- MariaDB: slave\_parallel\_threads
- MariaDB: slave\_parallel\_max\_queued
- MariaDB: slave\_domain\_parallel\_threads

# Multi-threaded Replication

---

- Different concurrency control
- MariaDB: slave\_parallel\_threads
- MariaDB: slave\_parallel\_max\_queued
- MariaDB: slave\_domain\_parallel\_threads
- MariaDB: slave\_parallel\_mode=optimistic | conservative | aggressive | minimal | none

# Multi-source Replication

---

- Different syntax for the source name
- CHANGE MASTER
  - MySQL
    - CHANGE MASTER TO MASTER\_HOST='source1' FOR CHANNEL 'source\_1';
  - MariaDB
    - CHANGE MASTER 'source\_1' TO MASTER\_HOST='source1';

# Multi-source Replication

---

- Different syntax for the source name
- Slave status

- MySQL

- SHOW SLAVE STATUS FOR CHANNEL 'source1';
- SELECT \* FROM performance\_schema.replication\_connection\_status WHERE CHANNEL\_NAME='source\_1';
- Default is for all channels

- MariaDB

- SHOW SLAVE 'source\_1' STATUS
- SHOW ALL SLAVES STATUS
- Default is default connection

# SQL Differences

# ANALYZE

---

- MySQL

```
mysql> explain analyze select * from ol where thread_id=10432 and site_id != 9939
      -> order by id limit 3\G
*****
EXPLAIN: -> Limit: 3 row(s)  (actual time=364.792..364.792 rows=0 loops=1)
      -> Filter: ((ol.thread_id = 10432) and (ol.site_id <> 9939))  (cost=0.06 rows=3)
          (actual time=364.789..364.789 rows=0 loops=1)
      -> Index scan on ol using PRIMARY  (cost=0.06 rows=33)
          (actual time=0.417..337.585 rows=100000 loops=1)
```



# ANALYZE

---

- MariaDB

```
MariaDB [test]> analyze select * from ol
      -> where thread_id=10432 and site_id != 9939 order by id limit 3\G
***** 1. row *****
...
      type: index
possible_keys: thread_id
      key: PRIMARY
key_len: 4
      ref: const
      rows: 100000
r_rows: 100000.00
      filtered: 8.96
r_filtered: 0.00
      Extra: Using where
```



# EXPLAIN for Running Thread

---

- MySQL

```
mysql> show processlist\G
***** 1. row *****
    Id: 2
    User: root
    Host: localhost:41352
    db: test
    Command: Query
    Time: 37
    State: User sleep
    Info: select sleep(100)
  Rows_sent: 0
Rows_examined: 0
...
```



# EXPLAIN for Running Thread

---

- MySQL

```
mysql> explain for connection 2\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: NULL
    partitions: NULL
        type: NULL
possible_keys: NULL
          key: NULL
     key_len: NULL
         ref: NULL
        rows: NULL
    filtered: NULL
       Extra: No tables used
1 row in set (0.00 sec)
```

# EXPLAIN for Running Thread

---

- MariaDB

```
MariaDB [(none)]> show processlist\G
***** 1. row *****
    Id: 3
    User: root
    Host: localhost:51126
    db: test
    Command: Query
    Time: 2
    State: User sleep
    Info: select sleep(30)
Progress: 0.000
```

# EXPLAIN for Running Thread

---

- MariaDB

```
MariaDB [(none)]> show explain for 3\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: NULL
        type: NULL
possible_keys: NULL
          key: NULL
     key_len: NULL
       ref: NULL
      rows: NULL
    Extra: No tables used
1 row in set, 1 warning (25.718 sec)
```



# EXPLAIN for Running Thread

---

- MariaDB

```
MariaDB [(none)]> show processlist\G
***** 1. row *****
    Id: 3
    User: root
    Host: localhost:51126
    db: test
    Command: Query
    Time: 2
    State: User sleep
    Info: select sleep(40)
Progress: 0.000
...
MariaDB [(none)]> show explain for 3\G
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
```



# Histograms and Table-Independent Statistics

---

- MySQL
  - Only histograms
  - Collected on demand

# Histograms and Table-Independent Statistics

---

- MySQL
  - Only histograms
  - Collected on demand
- MariaDB
  - Histograms collected and used by default
  - Other
    - Collected on demand
    - Used by default

# Functional Differences

# JSON

---

- MySQL
  - Special data type
  - Functions, not available in MariaDB
    - JSON\_OVERLAPS
    - JSON\_PRETTY
    - JSON\_STORAGE\_FREE
    - JSON\_STORAGE\_SIZE
    - JSON\_TABLE

# JSON

---

- MySQL
- MariaDB
  - Data type JSON mapped to LONGTEXT
    - CHECK (JSON\_VALID(...)) automatically enabled if JSON alias is used
  - Functions, not available in MySQL
    - JSON\_COMPACT
    - JSON\_DETAILED
    - JSON\_EXISTS
    - JSON\_LOOSE
    - JSON\_QUERY
    - JSON\_VALUE



# JSON

---

- MySQL
- MariaDB
- Replication
  - Statement-based works fine
  - Row-based
    - Convert MySQL JSON data type to LONGTEXT

# GIS

---

- "SQL with Geometry Types" of OGC
  - Both MySQL and MariaDB
- **Not** all MySQL GIS functions are in MariaDB

# Regular Expressions

---

- MySQL
  - Full Unicode support since 8.0.4
    - ICU library
  - Extended regex syntax

# Regular Expressions

---

- MySQL
  - Full Unicode support since 8.0.4
    - ICU library
  - Extended regex syntax
- MariaDB
  - Full Unicode support since 10.5
  - PCRE syntax

# Regular Expressions

---

- MySQL
  - Full Unicode support since 8.0.4
    - ICU library
  - Extended regex syntax
- MariaDB
  - Full Unicode support since 10.5
  - PCRE syntax
- REGEXP\_LIKE only in MySQL

# Localization

---

- Character set gb18030 only in MySQL
- MariaDB has more collations
- MariaDB supports **not all** MySQL collations

# Diagnostics

# Progress for ALTER TABLE

---

- MySQL
  - In Performance Schema

```
mysql> SELECT EVENT_NAME, WORK_COMPLETED, WORK_ESTIMATED
      -> FROM performance_schema.events_stages_current where event_name like '%alter%';
+-----+-----+-----+
| EVENT_NAME          | WORK_COMPLETED | WORK_ESTIMATED |
+-----+-----+-----+
| stage/innodb/alter table (read PK and internal sort) |        488 |       8512 |
...
| stage/innodb/alter table (read PK and internal sort) |      9396 |      9396 |
+-----+-----+-----+
1 row in set (0.00 sec)
```



# Progress for ALTER TABLE

---

- MariaDB
  - In Information Schema

```
MariaDB [information_schema]> select stage, max_stage, progress
      -> from processlist where info like 'alter table%';
+-----+-----+-----+
| stage | max_stage | progress |
+-----+-----+-----+
|     1 |         2 |    46.881 |
...
|     1 |         2 |    68.103 |
...
|     1 |         2 |    83.452 |
+-----+-----+-----+
1 row in set (0.001 sec)
```



# Progress for ALTER TABLE

---

- MariaDB
  - In SHOW PROCESSLIST

```
MariaDB [information_schema]> show processlist\G
***** 1. row *****
    Id: 5
    User: root
    Host: localhost:53616
    db: employees
    Command: Query
    Time: 4
    State: copy to tmp table
    Info: alter table salaries engine=innodb
Progress: 31.510
```



# Progress for ALTER TABLE

---

- MariaDB
  - In the supporting client

```
MariaDB [employees]> alter table salaries engine=innodb;
Stage: 1 of 2 'copy to tmp table'    73.4% of stage done
```



# Performance Schema

---

- Tables, which MariaDB misses
  - Locks
    - `data_lock_waits`
    - `data_locks`
    - MariaDB: in Information Schema `SE_*`

# Performance Schema

---

- Tables, which MariaDB misses
  - Variables
    - `global_variables` - MariaDB: in I\_S
    - `session_variables` - MariaDB: in I\_S
    - `variables_by_thread` - MariaDB: do not exist
    - `variables_info` - MariaDB: I\_S.SYSTEM\_VARIABLES

# Performance Schema

---

- Tables, which MariaDB misses
  - Replication

- replication\_applier\_filters
- replication\_applier\_global\_filters
- replication\_applier\_status\_by\_worker
- replication\_connection\_status
- MariaDB: SHOW SLAVE STATUS

# Performance Schema

---

- Tables, which MariaDB misses
  - Errors

- `events_errors_summary_by_account_by_error`
- `events_errors_summary_by_host_by_error`
- `events_errors_summary_by_thread_by_error`
- `events_errors_summary_by_user_by_error`
- `events_errors_summary_global_by_error`

# Performance Schema

---

- Tables, which MariaDB misses
  - Statements histograms
    - `events_statements_histogram_by_digest`
    - `events_statements_histogram_global`

# Performance Schema

---

- Tables, which MariaDB misses
  - User-defined functions
    - `user_defined_functions`
    - MariaDB: `mysql.func`

# Performance Schema

---

- Tables, which MariaDB misses
  - Binary log status and LSN
    - `log_status`
    - MariaDB: SHOW MASTER STATUS and SHOW ENGINE INNODB STATUS

# Conclusions

---

- Initial upgrade is straightforward
- Feature differences are easily solvable
- MariaDB has options which simplify migration
- Migrate to explore all MariaDB advantages!

# More Details

---



Migrating to MariaDB from MySQL

MariaDB vs MySQL: Features

Function Difference

Replication Compatibility Matrix

Optimizer Features Comparison Matrix

Spatial Support Matrix



PERCONA

# Thank you!

---



[www.slideshare.net/SvetaSmirnova](http://www.slideshare.net/SvetaSmirnova)



[twitter.com/svetsmirnova](http://twitter.com/svetsmirnova)



[github.com/svetasmirnova](http://github.com/svetasmirnova)



DATABASE PERFORMANCE  
MATTERS