

MariaDB semi-sync replication using containers

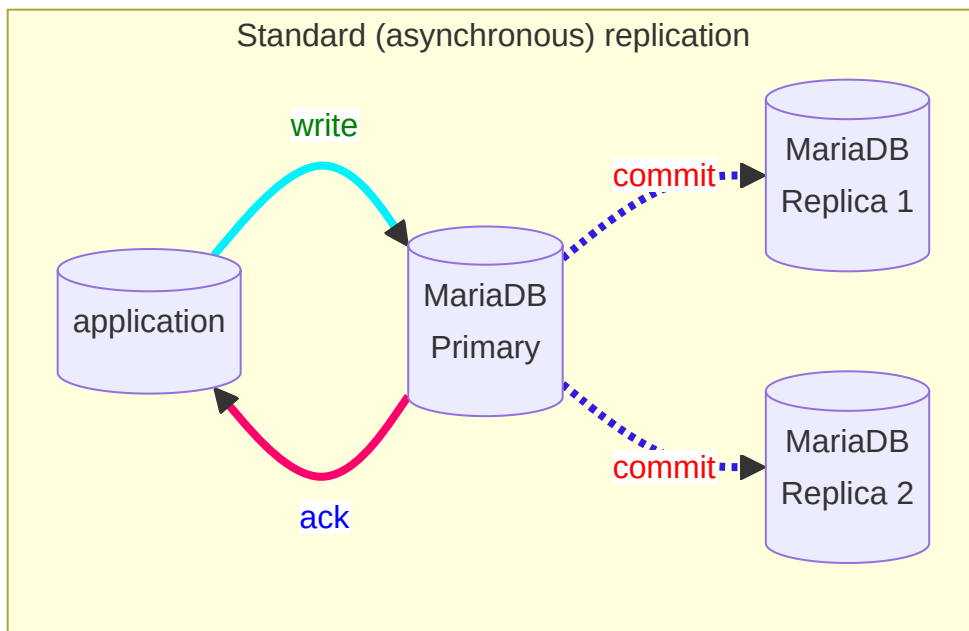
Anel Husakovic,

Zenica, Novembar, 2023

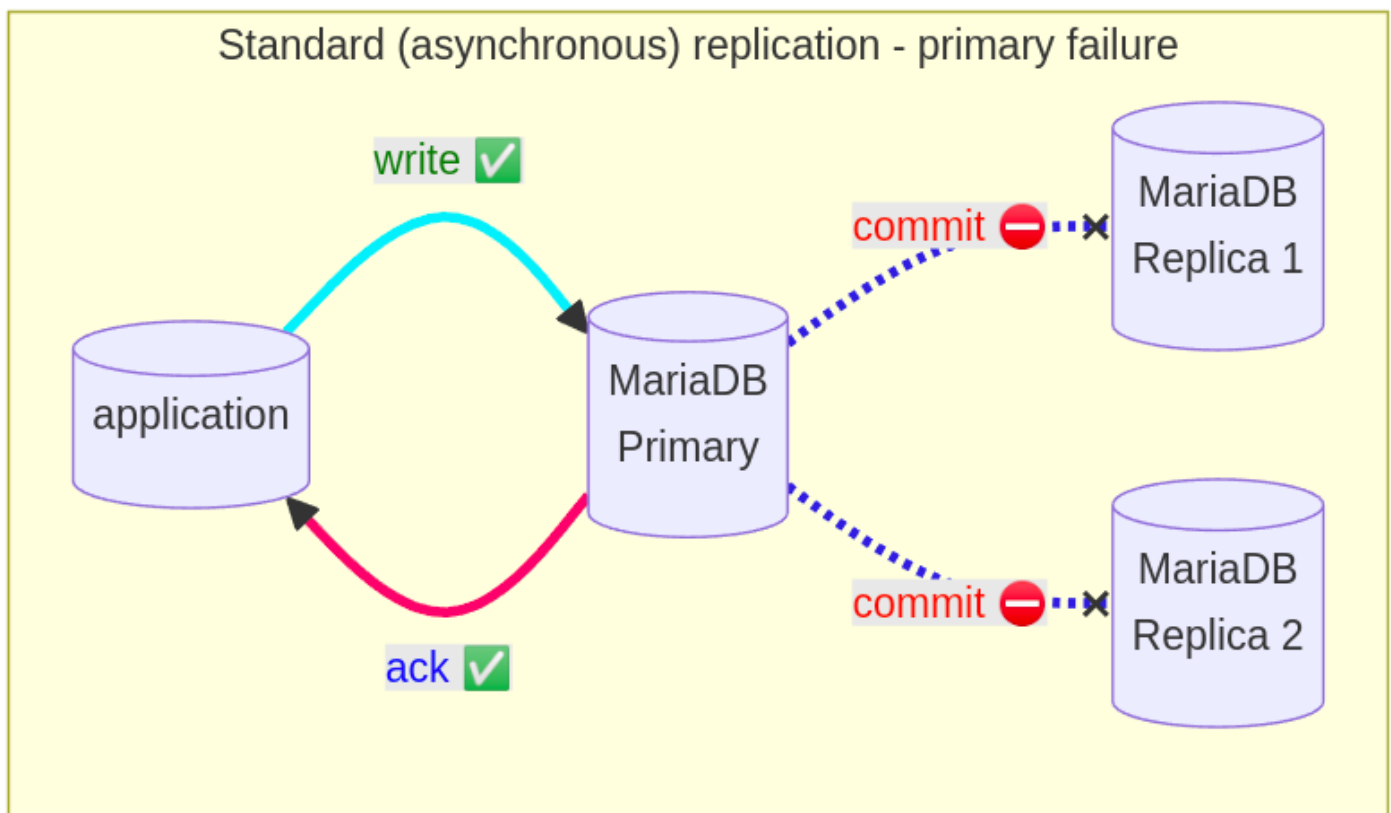
- [1. Standard replication configuration](#)
- [2. Standard replication transaction](#)
- [3. Semi-sync replication configuration](#)
- [4. Semi-sync replication transaction example](#)
- [5. Semi-sync demo example with containers](#)
 - [5.1 Check containers](#)
 - [5.1.1 Check primary](#)
 - [5.1.2 Check binary logs](#)
 - [5.1.3 Check replica\[s\]](#)
 - [5.2 Start replicating](#)
 - [5.2.1 Create table](#)
 - [5.2.2 Insert data](#)
 - [5.2.3 Check using GUI Dolphie](#)

In the last blog [MariaDB replication using containers](#), we showed how to properly replicate data in MariaDB using Docker containers.

We used standard or asynchronous or lazy replication.



The problem with this type of replication is potential data loss in case if primary goes down, before the replica gets the committed changes.



To overcome this type of errors, there is semi-sync replication.

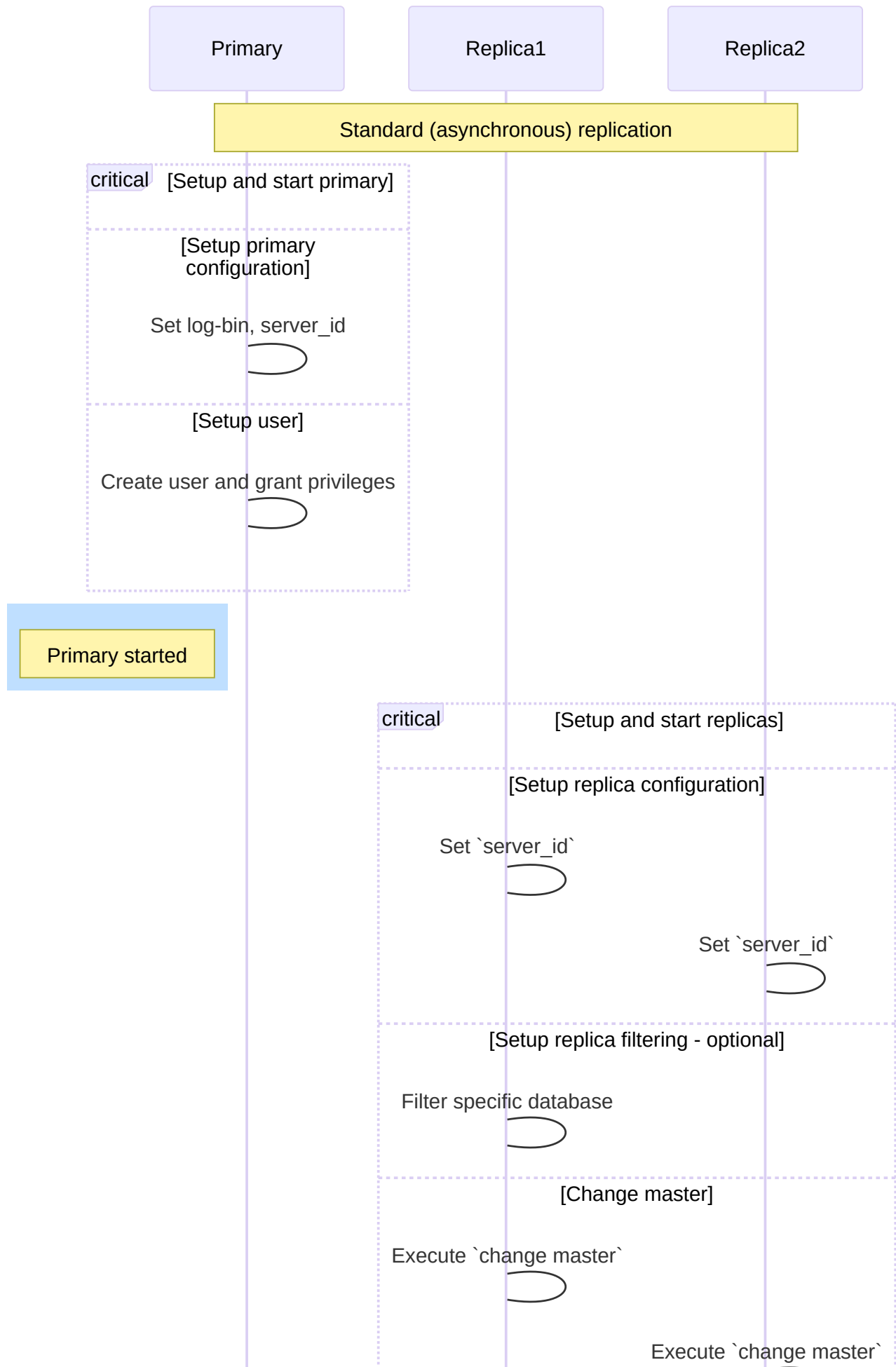
In this blog we will visualise following:

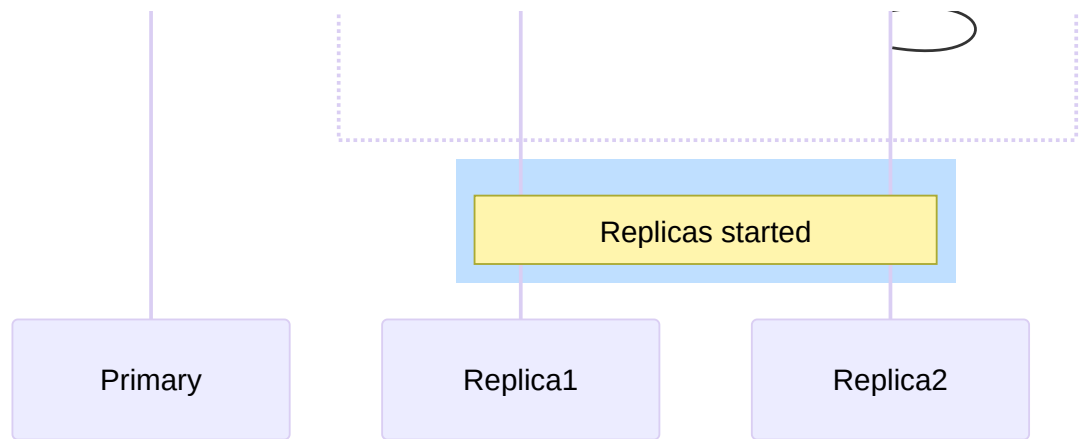
1. Standard replication configuration
2. Standard replication transaction example

- 3. Semi-sync replication configuration
- 4. Semi-sync replication transaction example
- 5. Semi-sync demo example

1. Standard replication configuration

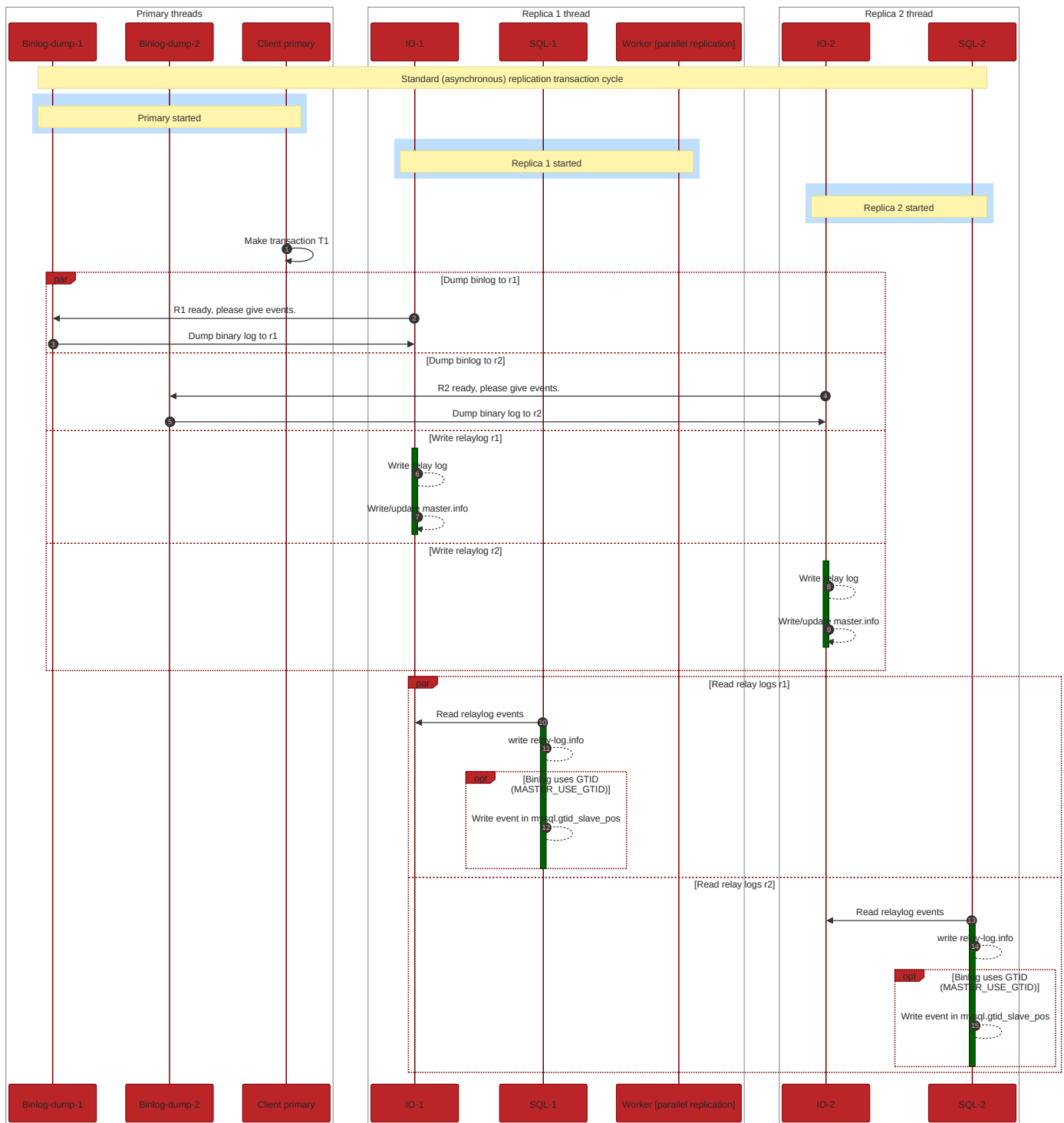
To configure the standard replication implemented in previous blog was straight forward:





2. Standard replication transaction

On thread level (see [replication-threads](#)), flow of active transaction we can express as following:

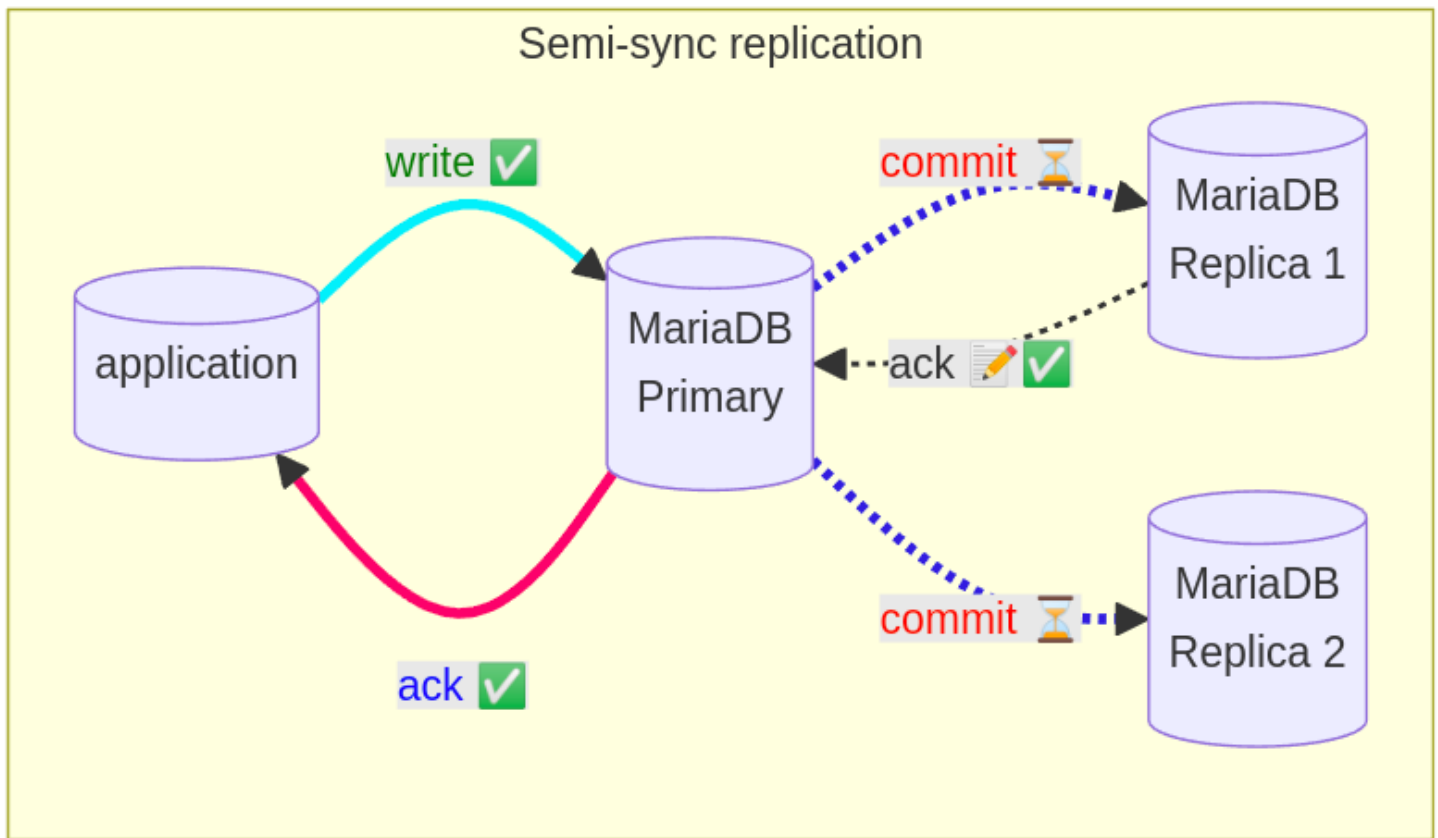


Type of the replication is asynchronous that means that we don't have any feedback information from replicas, that event has been successfully received by replica, as can be seen from picture.

3. Semi-sync replication configuration

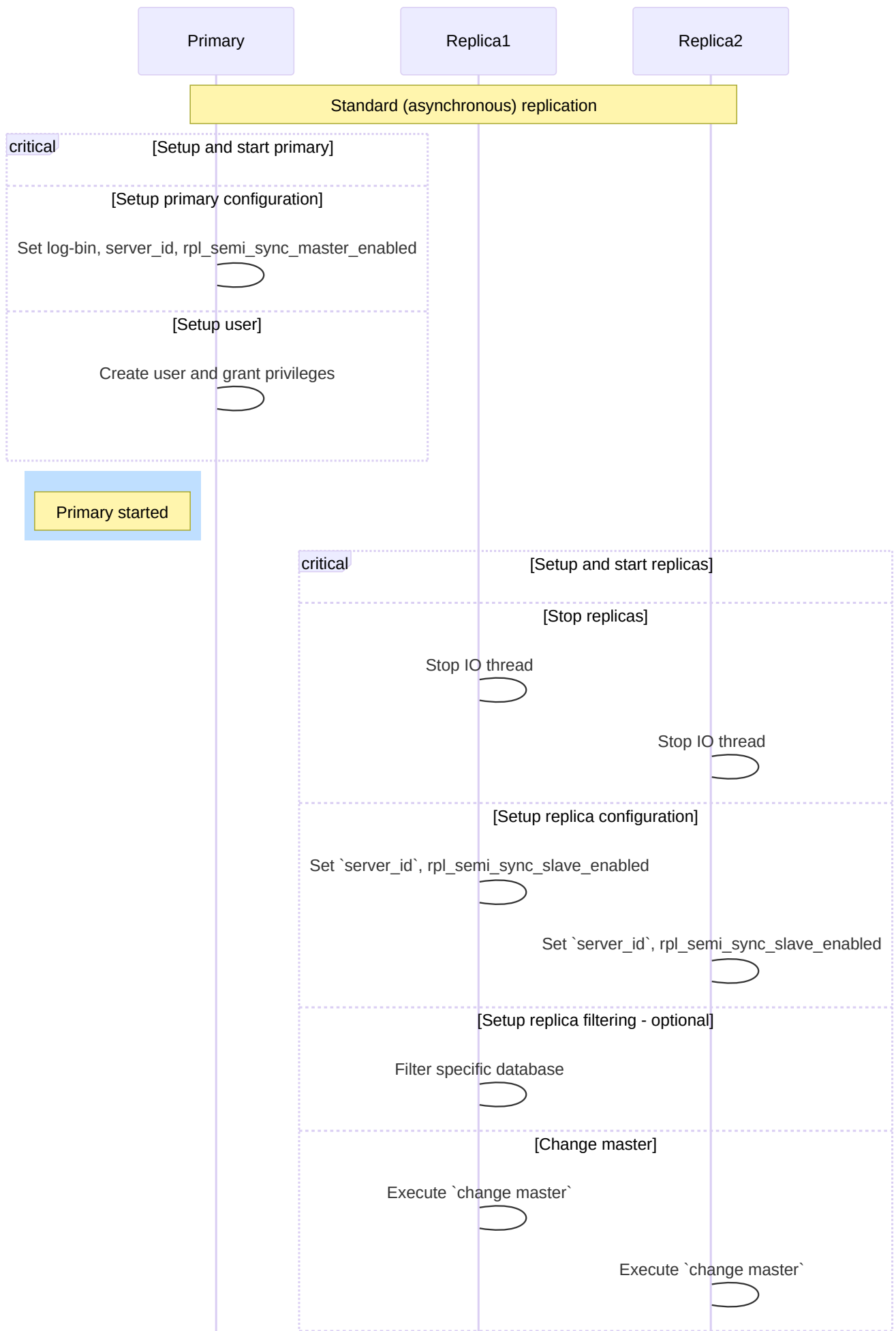
Before confirming the client request, at least one replica has to confirm receipt of data changes (IO thread),

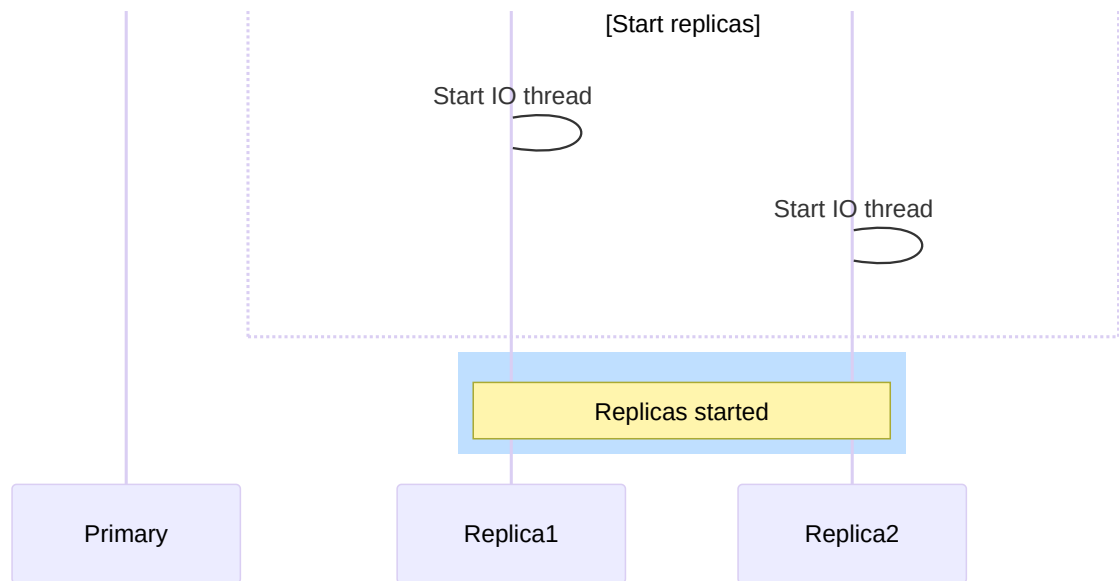
not that actually replica applied those data changes.



To configure the semi-sync replication we need to stop replicase and set environment variables on primary and replicas.

On primary set `rpl_semi_sync_master_enabled` and on replicas set `rpl_semi_sync_slave_enabled`.

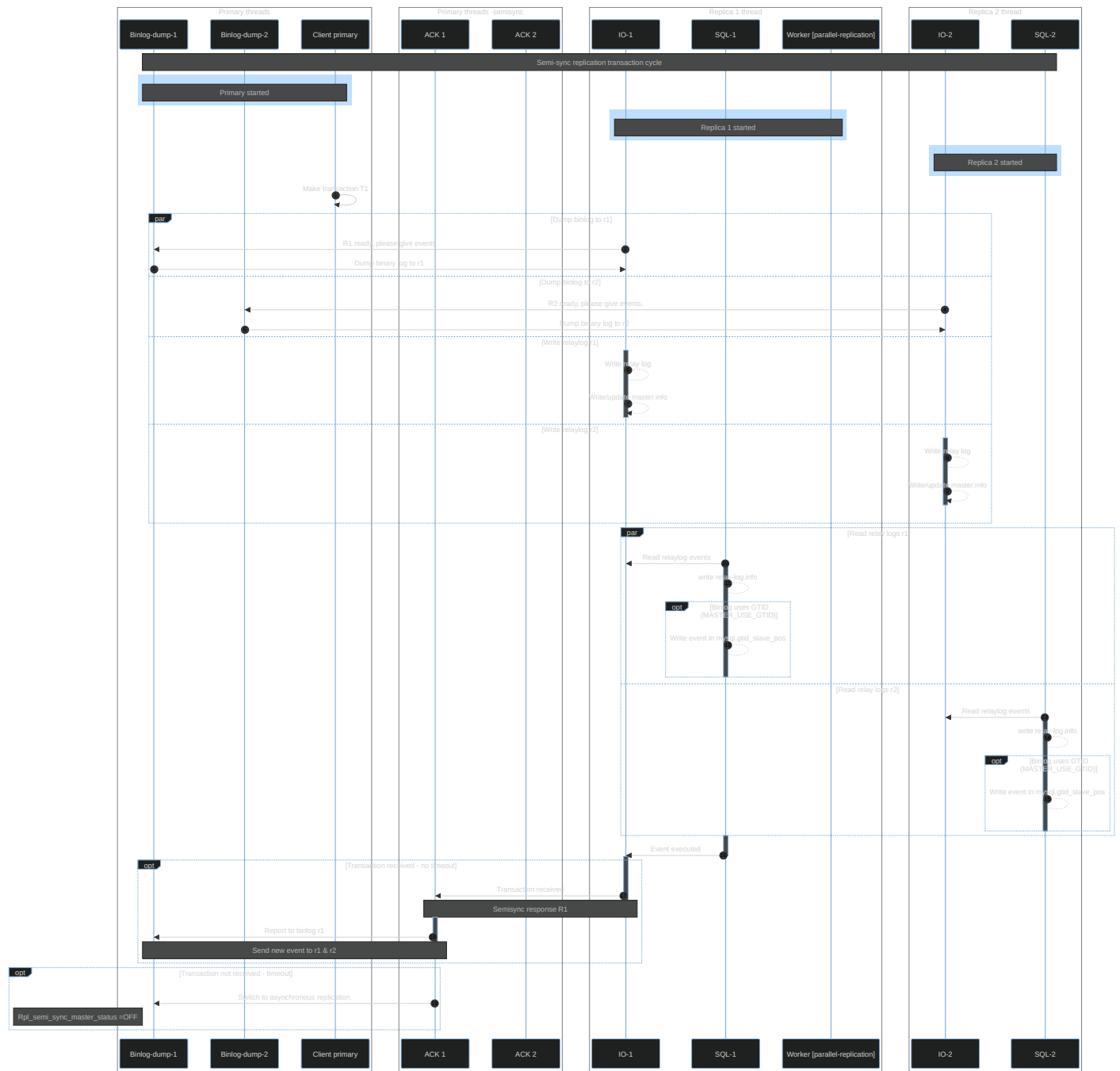




4. Semi-sync replication transaction example

Semi-sync should overcome that problem, with introducing additional primary thread , called "[ACK Receiver Thread](#)".

Only one replica is needed to confirm, that it has received and logged the events, as showed on following picture:



5. Semi-sync demo example with containers

- In this example we will be using stateless application, just as a proof of concept. LINK todo
- If statefull example is needed, consider using persistent volumes as I explained in this example.LINK todo
- We will be using GTIDs, that are enabled automatically.

This way replication will start at the position of the last GTID replicated to replica (seen from `gtid_slave_pos` system variable).

5.1 Check containers

- Start containers

```
$ docker compose up
```

- Check container statuses

```
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
1ddb1fa67fef   mariadb:lts    "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes (healthy)
dca7a79526ca   mariadb:lts    "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes (healthy)
addabd306bb4   mariadb:lts    "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes (healthy)
```

- Check logs

#	primary	replica 1	replica 2
1.	"mariadb-primary 2023-11-23 12:21:47 0 [Note] Semi-sync replication initialized for transactions."		
2.	"mariadb-primary 2023-11-23 12:21:47 0 [Note] Semi-sync replication enabled on the master."		
3.		"mariadb-replica-1 2023-11-23 12:21:58 4 [Note] Master connection name: " Master_info_file: 'master.info' Relay_info_file: 'relay-log.info'"	

#	primary	replica 1	replica 2
4.			<p>mariadb-replica-2 2023-11-23 12:21:58 4 [Note] Master connection name: "</p> <p>Master_info_file: 'master.info'</p> <p>Relay_info_file: 'relay-log.info'</p>
5.		<p>mariadb-replica-1 2023-11-23 12:21:58 4 [Note] 'CHANGE MASTER TO executed'.</p> <p>Previous state master_host="", master_port='3306', master_log_file="", master_log_pos='4'. New state master_host='mariadb-primary', master_port='3306', master_log_file="", master_log_pos='4'.</p>	
6.		<p>mariadb-replica-1 2023-11-23 12:21:58 4 [Note] Previous Using_Gtid=Slave_Pos. New Using_Gtid=Slave_Pos</p>	
7.			<p>mariadb-replica-2 2023-11-23 12:21:58 4 [Note] 'CHANGE MASTER TO executed'.</p> <p>Previous state master_host="", master_port='3306', master_log_file="", master_log_pos='4'. New state master_host='mariadb-primary', master_port='3306', master_log_file="", master_log_pos='4'.</p>
8.			<p>mariadb-replica-2 2023-11-23 12:21:58 4 [Note] Previous</p>

#	primary	replica 1	replica 2
			Using_Gtid=Slave_Pos. New Using_Gtid=Slave_Pos
9.		mariadb-replica-1 2023-11-23 12:21:58 5 [Note] Slave I/O thread: Start semi-sync replication to master 'repl@mariadb-primary:3306' in log " at position 4	
10.		mariadb-replica-1 2023-11-23 12:21:58 6 [Note] Slave SQL thread initialized, starting replication in log 'FIRST' at position 4, relay log './mariadb- relay-bin.000001' position: 4; GTID position "	
11.		mariadb-replica-1 2023-11-23 12:21:58 5 [Note] Slave I/O thread: connected to master 'repl@mariadb- primary:3306',replication starts at GTID position "	
12.	mariadb-primary 2023-11-23 12:21:58 7 [Note] Start binlog_dump to slave_server(2), pos(, 4), using_gtid(1), gtid(")		
13.	mariadb-primary 2023-11-23 12:21:58 7 [Note] Start semi-sync binlog_dump to		

#	primary	replica 1	replica 2
	slave (server_id: 2), pos(/.mariadb-bin.000001, 4)		
14.			mariadb-replica-2 2023-11-23 12:21:58 5 [Note] Slave I/O thread: Start semi-sync replication to master 'repl@mariadb-primary:3306' in log " at position 4
15.			mariadb-replica-2 2023-11-23 12:21:58 6 [Note] Slave SQL thread initialized, starting replication in log 'FIRST' at position 4, relay log './mariadb-relay-bin.000001' position: 4; GTID position "
16.			mariadb-replica-2 2023-11-23 12:21:58 5 [Note] Slave I/O thread: connected to master 'repl@mariadb-primary:3306',replication starts at GTID position "
17.	mariadb-primary 2023-11-23 12:21:58 8 [Note] Start binlog_dump to slave_server(3), pos(, 4), using_gtid(1), gtid(")		

5.1.1 Check primary

- Check that semi-sync is enabled using `mariadb client`

```
$ docker exec -it mariadb-primary mariadb -uroot -psecret -e "select @@rpl_semi_sync_ma
+-----+
| @@rpl_semi_sync_master_enabled |
+-----+
|                                1 |
+-----+
```

- Check `rpl_semi_sync_master_timeout` variable (default 10[s]).

When this time elapses and primary doesn't get any acknowledgment from replica, it switches back to asynchronous replication.

```
$ docker exec -it mariadb-primary mariadb -uroot -psecret -e "select @@rpl_semi_sync_ma
+-----+
| @@rpl_semi_sync_master_timeout |
+-----+
|                               10000 |
+-----+
```

- Check master status

```
$ docker exec mariadb-primary mariadb -uroot -psecret -e "show master status\G;"
***** 1. row *****
      File: mariadb-bin.000002
      Position: 344
      Binlog_Do_DB:
      Binlog_Ignore_DB:
```

- Check databases for replication

```
$ docker exec -it mariadb-primary mariadb -uroot -psecret -e "show databases like '%test
+-----+
| Database (%test%) |
+-----+
| testdb           |
+-----+
```

5.1.2 Check binary logs

- There are 2 binary logs and index:

```
$ docker exec mariadb-primary bash -c "ls /var/lib/mysql/|grep maria"
mariadb-bin.000001
mariadb-bin.000002
mariadb-bin.index
```

- The same can be seen from mariadb client:

```
$ docker exec mariadb-primary mariadb -uroot -psecret -e "show binary logs\G;"
***** 1. row *****
Log_name: mariadb-bin.000001
File_size: 1166
***** 2. row *****
Log_name: mariadb-bin.000002
File_size: 344
```

- You are free to inspect binary logs with mariadb-binlog client.
- Example of fresh start


```

$ docker exec mariadb-primary mariadb-binlog /var/lib/mysql/mariadb-bin.000002
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!40019 SET @@session.max_delayed_threads=0*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE, COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#231123 12:21:47 server id 1  end_log_pos 256 CRC32 0x21866123  Start: binlog v 4, server
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
20NfZQ8BAAAA/AAAAABAAABAAQMTAuMTEuNi1NYXJpYURCLTE6MTAuMTEuNittYXJpYX51YnUy
MjA0LWxvZwAAAAAADbQ19lEzgNAAGAEgAEBAQEEgAA5AAEGggAAAAICAgCAAAACgoKAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAEEwQADQgICAoKCgEjYYYh
'/*!*/;
# at 256
#231123 12:21:47 server id 1  end_log_pos 299 CRC32 0xdd5f27d6  Gtid list [0-1-5]
# at 299
#231123 12:21:47 server id 1  end_log_pos 344 CRC32 0xca6cbf75  Binlog checkpoint maria
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

```

- When creating the table and inserting the values, we updated

5.1.3 Check replica[s]

- Check replica semi-sync status (let's check just single replica)

```

$ docker exec -it mariadb-replica-2 mariadb -uroot -psecret -e "select @@rpl_semi_sync_
+-----+
| @@rpl_semi_sync_slave_enabled |
+-----+
|                                1 |
+-----+

```

- Check replicated database

```
$ docker exec -it mariadb-replica-1 mariadb -uroot -psecret -e "show databases like '%test'"
+-----+
| Database (%test%) |
+-----+
| testdb            |
+-----+
```

```
$ docker exec -it mariadb-replica-2 mariadb -uroot -psecret -e "show databases like '%test'"
+-----+
| Database (%test%) |
+-----+
| testdb            |
+-----+
```

5.2 Start replicating

5.2.1 Create table

5.2.1.1 Primary

- Transaction T1: create table t(t int);
- Status

```
MariaDB [testdb]> show master status;
```

```
+-----+-----+-----+-----+
| File                | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000003  | 485      |              |                  |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

5.2.1.2 Binlog

- Binlog (GTID 0-1-6)

```

$ docker exec mariadb-primary mariadb-binlog /var/lib/mysql/mariadb-bin.000003
#231123 12:57:18 server id 1  end_log_pos 299 CRC32 0x0f7f30b3  Gtid list [0-1-5]
# at 299
#231123 12:57:18 server id 1  end_log_pos 344 CRC32 0x263247e6  Binlog checkpoint maria
# at 344
#231123 14:05:49 server id 1  end_log_pos 386 CRC32 0x9f9ba209  GTID 0-1-6 ddl
/*!100101 SET @@session.skip_parallel_replication=0/*!*/;
/*!100001 SET @@session.gtid_domain_id=0/*!*/;
/*!100001 SET @@session.server_id=1/*!*/;
/*!100001 SET @@session.gtid_seq_no=6/*!*/;
# at 386
#231123 14:05:49 server id 1  end_log_pos 485 CRC32 0xb88ff6c5  Query  thread_id=821
use `testdb`/*!*/;
SET TIMESTAMP=1700748349/*!*/;
SET @@session.pseudo_thread_id=821/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1/*!*/;
SET @@session.sql_mode=1411383296/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*!*/;
/*!\C utf8mb3 *//*!*/;
SET @@session.character_set_client=utf8mb3,@@session.collation_connection=33,@@session.collation_database=DEFAULT/*!*/;
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
create table t(t int)
/*!*/;
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

```

5.2.1.3 Replica

```
$ docker exec -it mariadb-replica-2 mariadb -uroot -psecret -e "use testdb; show tables;
+-----+
| Tables_in_testdb |
+-----+
| t                  |
+-----+
+-----+-----+
| Table | Create Table
+-----+-----+
| t      | CREATE TABLE `t` (
`t` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci |
+-----+-----+
MariaDB [(none)]> show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: mariadb-primary
      Master_User: repl
      Master_Port: 3306
      Connect_Retry: 10
      Master_Log_File: mariadb-bin.000003
Read_Master_Log_Pos: 485
      Relay_Log_File: mariadb-relay-bin.000002
      Relay_Log_Pos: 786
Relay_Master_Log_File: mariadb-bin.000003
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
Replicate_Rewrite_DB:
      Replicate_Do_DB:
Replicate_Ignore_DB:
      Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
Exec_Master_Log_Pos: 485
      Relay_Log_Space: 1097
      Until_Condition: None
      Until_Log_File:
```

```

    Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
    Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
    Last_IO_Errno: 0
    Last_IO_Error:
    Last_SQL_Errno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:
    Master_Server_Id: 1
    Master_SSL_Crl:
    Master_SSL_Crlpath:
        Using_Gtid: Slave_Pos
        Gtid_IO_Pos: 0-1-6
Replicate_Do_Domain_Ids:
Replicate_Ignore_Domain_Ids:
    Parallel_Mode: optimistic
    SQL_Delay: 0
    SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
    Slave_DDL_Groups: 1
Slave_Non_Transactional_Groups: 0
Slave_Transactional_Groups: 0
1 row in set (0.000 sec)

```

5.2.2 Insert data

5.2.2.1 Primary

- Transactino T2: insert into t values (1),(2),(314),(1618);
- Status

```
MariaDB [testdb]> show master status;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000003 |      668 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

5.2.2.2 Binlog

- GTID 0-1-7

```
# at 485
```

```
#231123 14:46:19 server id 1  end_log_pos 527 CRC32 0x9ef73274  GTID 0-1-7 trans
```

```
/*!100001 SET @@session.gtid_seq_no=7*//*!*/;
```

```
START TRANSACTION
```

```
/*!*/;
```

```
# at 527
```

```
#231123 14:46:19 server id 1  end_log_pos 637 CRC32 0xaaafb199a  Query    thread_id=821
```

```
SET TIMESTAMP=1700750779*//*!*/;
```

```
insert into t values (1),(2),(314),(1618)
```

```
/*!*/;
```

```
# at 637
```

```
#231123 14:46:19 server id 1  end_log_pos 668 CRC32 0xb4fb80ec  Xid = 2644
```

```
COMMIT*//*!*/;
```

```
DELIMITER ;
```

```
# End of log file
```

5.2.2.3 Replica

```
$ docker exec -it mariadb-replica-2 mariadb -uroot -psecret -e "select * from testdb.t"
+-----+
| t      |
+-----+
|      1 |
|      2 |
|    314 |
|   1618 |
+-----+
# Slave status changes
MariaDB [(none)]> show slave status \G
      Slave_IO_State: Waiting for master to send event
      Master_Log_File: mariadb-bin.000003
  Read_Master_Log_Pos: 668
      Relay_Log_File: mariadb-relay-bin.000002
      Relay_Log_Pos: 969
  Exec_Master_Log_Pos: 668
      Relay_Log_Space: 1280
       Using_Gtid: Slave_Pos
       Gtid_IO_Pos: 0-1-7
  Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
```

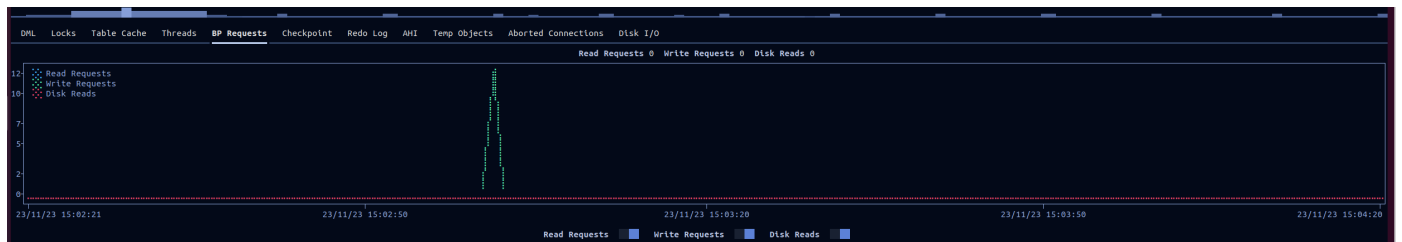
5.2.3 Check using GUI Dolphie

- I wanted to check [dolphie](#), so I installed it on my primary container:

```
apt-get update && \
apt-get -y install python3-pip && \
pip install dolphie
```

- There is very nice GUI, but it supports MySQL's group replication only





Conclusion and further readings

Special thanks to Brandon & (others) for reviewing this blog post.

If you come across any problems in this blog, with the design, or edge cases that don't work as expected, please let us know. You are welcome to chat about it on Zulip. As always you can use our JIRA bug/feature request in the MDEV project for any bug/feature request you may encounter.

- This blog closes [MDBF](#). (this will not be part of the blog)

// Not working <https://github.com/mermaid-js/mermaid/issues/821>