

Configuring Replication Failover

A common high availability (HA) configuration for MySQL is the Master/Master pair. In such a configuration, each side is both a Slave and a Master. The application can write to only one instance at a time (Active/Passive) or to both instances (Active/Active). Assuming you are familiar with MySQL replication configuration and management, you can configure [Master-to-Master replication](#) for this purpose.

If you are using MySQL 5.7 or higher, please see [Using ClustrixDB as a Replication Master](#) and [Using ClustrixDB as a Replication Slave](#) for information on GTIDs.

Configuring MySQL-to-Clustrix Replication in Master/Master

This section tells you how to make a MySQL Master a Slave to the ClustrixDB system. Prerequisites:

- Configure ClustrixDB to generate MySQL binary logs.
- Create a replication user on the MySQL Master server.
- Ensure that ClustrixDB and MySQL have unique values for `global server_id`.

Assuming that ClustrixDB is configured as a Slave to an existing MySQL Master, you can configure MySQL as a Slave to the ClustrixDB system. To prevent undesired "write" actions on the ClustrixDB Slave from being replicated to the Master MySQL instance, put ClustrixDB into a read-only mode by issuing the following command. Any updates coming through the binary log bypass read-only:

```
slave> SET GLOBAL read_only = true;
```

You now have an Active/Passive replication topology, with ClustrixDB acting as the Passive (Slave) system.

Promoting a ClustrixDB Slave to Master

Before promoting, ensure that the Passive Slave is keeping up with the replication stream from the Active Master. Verify that the Passive Slave is not more than a couple of seconds behind the Active Master by issuing the `SHOW SLAVE STATUS` command. If the Slave is significantly behind, wait for it to catch up, to reduce the amount of time required for the replication network to quiesce. Verify that the MySQL Active Master is replicating from the ClustrixDB Passive Slave instance.

To quiesce the system and get the Active and Passive systems into a known consistent state, disable writes to the MySQL Active Master by issuing the following command:

```
master> SET GLOBAL read_only = true;
```

To determine the active MySQL Master's position, issue the following command:

```
master> SHOW MASTER STATUS\G
```

When the two systems display the same position, they are in a consistent and quiescent state. Now you can safely promote either system to the role of Active Master.

To promote a ClustrixDB Passive Slave to Active Master, disable read-only mode on ClustrixDB by issuing the following command:

```
slave> SET GLOBAL read_only = false;
```

Next, redirect the application to use the ClustrixDB system as its write target. The ClustrixDB system is now configured as the Active Master. Verify that events are getting replicated to the MySQL server, which is now configured as the Passive Slave.

When ClustrixDB is the Active Master, you can obtain information about all the slaves connected to the master by issuing the following command:

```
master> SHOW FULL processlist;
```

To easily identify the slave(s), a best practice is to use user names that end with "_slave".

To promote a ClustrixDB Slave (Passive) to a Master (Active) replication participant, perform the following steps:

1. Put the current MySQL Master instance into read-only mode and wait for the ClustrixDB Slave instance to catch up.
2. Disable read-only mode on the ClustrixDB Slave (if it is enabled).
3. Migrate application traffic.