

MySQL cluster with Multi-master or Replica mode

If you can't afford for your database to be down for even a few minutes, a Multi-Master cluster is a great option to ensure high availability.

A multi-master scenario means that one node can be taken offline (e.g. for maintenance or upgrade purposes) without impacting availability, as the other node will continue to serve production traffic. Further, it doubles your capacity to read or write to the database and provides an additional layer of protection against data loss.

MySQL includes a Multi-Master replication, and Elestio enables you to set up your MySQL Multi-Master cluster in just a few clicks.

To begin, you will need to have deployed two MySQL instances

1) Go to elestio Dashboard > Deploy new service> Databases > select MySQL, scroll down and name it for example **mysql-1** then click on the "Create service button"

The screenshot shows the 'Create Service' interface in the Elestio dashboard. At the top, there are three steps: 1. Select service (highlighted), 2. Select provider, region & service plan, and 3. Select Support & advanced setting. Below the steps, there are tabs for 'Databases', 'Applications', 'Development', 'Hosting & Infra', 'Full Stack', 'CI/CD', and 'All'. The 'Databases' tab is selected. A search bar labeled 'Search service by name' is present. The main area displays a grid of database service cards. The 'MySQL' card is highlighted with an orange border and has 'Details' and 'Select' buttons. Other cards include PostgreSQL, MariaDB, MongoDB, Redis, KeyDB, TimescaleDB, ClickHouse, and ScyllaDB, each with its logo and a brief description.

Service Name	Description	Action
PostgreSQL	PostgreSQL is a powerful, open-source object-relational database system, known for reliability, data integrity and performance.	
MySQL	MySQL is an Oracle-backed open-source RDBMS that runs on almost all platforms.	Details, Select
MariaDB	The open source relational database	
MongoDB	MongoDB is a document-oriented NoSQL database used for high-volume data storage.	
Redis	Redis is an open-source, in-memory database, cache and message broker.	
KeyDB	KeyDB is both your cache and database, for cloud-optimized solutions.	
TimescaleDB	TimescaleDB is the leading open-source relational database with support for time-series data.	
ClickHouse	ClickHouse is an open-source, column-oriented DBMS for online analytical processing.	
ScyllaDB	ScyllaDB is a true NoSQL database for the most demanding applications.	

2) Again, go to elestio Dashboard > Deploy new service> Databases > select MySQL, scroll down and name it for example **mysql-2** then click on the "Create service button"

3) Wait for the 2 instances to be ready

4) In the elestio dashboard open the service details of **mysql-1** and click on the "Configure cluster" button

5) Select in the partner instance dropdown "**mysql-2**" as the partner, then select "Multi Master" in Cluster mode, then click on "Apply changes" button

Server type: SMALL-1C-2G (1 VCPU - 2 GB RAM - 20 GB storage) Provider: hetzner

Configure Cluster ✕

Select a partner instance from the list

Choose a Service ▼

Or indicate details from an external instance

Partner IP to replicate as a slave

Partner TCP Port

Partner user for replication

Partner password for replication

Cluster Mode ☐ Disabled ☐ Replica ☐ Multi Master

If you selected "Multi-Master" replication, you will need to do the same cluster configuration from the other node with this node infos.

Cancel

Apply Changes

6) In the elestio dashboard open the service details of **mysql-2** and click on the "Configure cluster" button

7) Select in the partner instance dropdown "**mysql-1**" as the partner, then select "Multi-Master" in Cluster mode, then click on "Apply changes" button

All done. You now have a multi-master MySQL cluster.

You can now read and write on both instances. If instance A is down you will still be able to use instance B and vice versa. Also, if you restore a backup on one instance it will be automatically replicated to the other instance.

How to use Multi-Master cluster from Node.js

If you can configure your two master clusters in Round Robin in your MySQL driver, a load balancer is not needed. The client-side will split the traffic between your instances and avoid a dead node. This helps to greatly simplify the high-availability system.

The regular MySQL driver for node.js supports this:

<https://www.npmjs.com/package/mysql#poolcluster>

How to test your High Availability Cluster

1. Shut down one of the VMs (instance A). You should still be able to connect, read and write on your cluster.
2. Restart instance A, wait 30 seconds, then shut down instance B.
3. Test your connectivity and read/write access to the cluster again.
4. Finally, restart instance B.

How to use PHPMysqlAdmin to test your cluster

1. Open the service details and click on Admin UI to get url and credentials of PHPMysqlAdmin.
2. Open a browser tab with the Admin UI for instance B.
3. Open another browser tab for instance A.
4. Create a new database in instance A, add a table, and insert a line with sample data.
5. Check if the database created from A is correctly replicated to instance B.
6. Open the database in instance B.
7. Add or edit some rows in the database on instance B and check if correctly replicated to instance A.