

# MARIADB MAXSCALE TECHNICAL BRIEF

## Enterprise High Availability

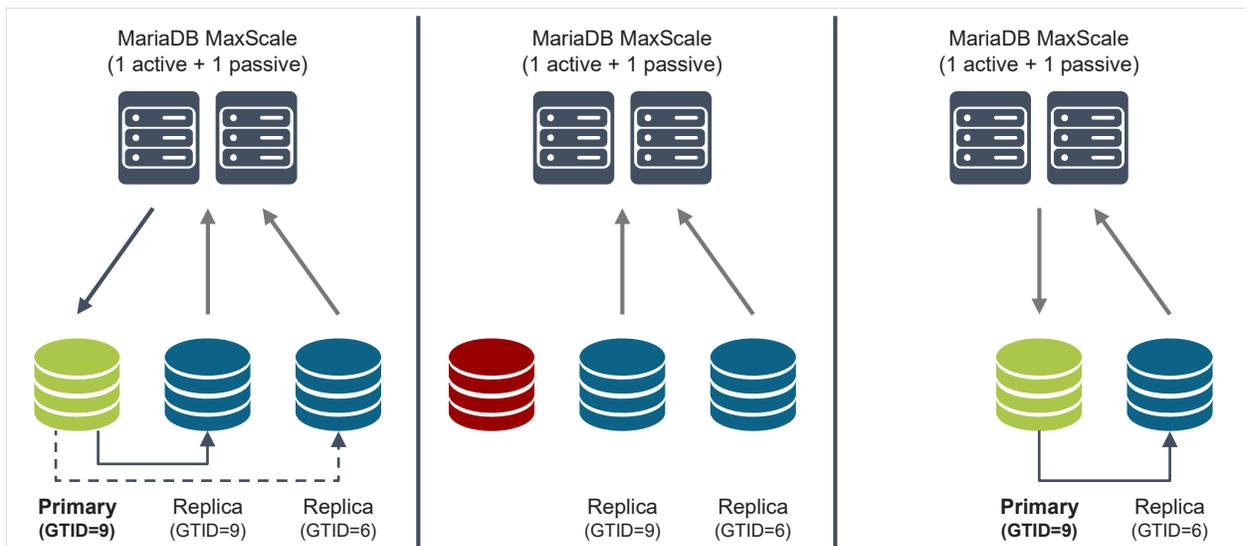
MariaDB MaxScale is an advanced database proxy for MariaDB Server, and a core component of MariaDB Platform – providing it with enterprise high availability, scalability, security and integration features. This technical brief provides an overview of the enterprise high availability features in MariaDB MaxScale.

### INFRASTRUCTURE ABSTRACTION

MariaDB MaxScale provides enterprise high availability by abstracting away the underlying database infrastructure from applications so they are neither aware of, nor disrupted by, changes in the topology (e.g., a failed database instance). In addition to its advanced monitoring and dynamic query routing, MariaDB MaxScale maintains connection, session and transaction state on behalf of applications – all to ensure applications are not disrupted by a failure.

### AUTOMATIC FAILOVER

MariaDB MaxScale monitors all of the database instances within a primary/replica deployment. If the primary fails, and automatic failover is enabled, it will promote the most up-to-date replica (based on its global transaction ID), wait for it to execute any transactions in its relay log and begin routing queries to it – all within seconds.



*Note: MariaDB MaxScale should be deployed in an active/passive configuration with two instances to ensure high availability and prevent it from being a single point of failure. Further, semi-synchronous replication should be used to prevent data loss in the event of an automatic failover.*



## ZERO DISRUPTION

While most databases include basic high availability features to minimize downtime, they lack the advanced features necessary to ensure failure is completely transparent to applications. MariaDB MaxScale removes the need for applications to create new connections or retry transactions after database availability has been restored.

### Connection migration

In the event of an automatic failover, rather than closing the client connection, MariaDB MaxScale can migrate backend server connections to the new primary so applications are not interrupted – and do not have to create a new connection.

### Delayed retry

If the primary fails, and before the automatic failover process has finished, MariaDB MaxScale can delay and retry writes rather than returning an error or closing the client connection. When combined with connection migration, this enables write requests sent during an automatic failover to be executed rather than returned with an error.

### Transaction replay

If the primary fails when there are transactions still in progress (i.e., in-flight transactions) and transaction replay is enabled, MariaDB MaxScale, after automatic failover, will replay the statements of any in-flight transactions on the new primary rather than rolling back the transaction and requiring the application to retry it.

### Read retry

If a replica fails after a read has been routed to it but before the results have been returned, and read retry is enabled, MariaDB MaxScale will automatically route the read to a different replica rather than return an error or close the client connection.

## CLUSTERING

There is no need for automatic failover when MariaDB Server is deployed as a multi-master cluster because every database instance is capable of executing reads and writes. However, the zero-interruption features in MariaDB MaxScale are still required in order to prevent applications from being interrupted by failures. Further, MariaDB MaxScale can assign primary and replica roles to database instances within a cluster to support read/write splitting, and thus remove write conflicts.

---

[mariadb.com](https://mariadb.com)

Americas: [sales-AMER@mariadb.com](mailto:sales-AMER@mariadb.com)  
Europe, Middle East, Africa: [sales-EMEA@mariadb.com](mailto:sales-EMEA@mariadb.com)  
Asia Pacific: [sales-APAC@mariadb.com](mailto:sales-APAC@mariadb.com)

© Copyright 2024 MariaDB plc  
699 Veterans Blvd.  
Redwood City, California USA