



MySQL Enterprise High Availability

A Reference Guide

A MySQL® White Paper

© 2018, Oracle Corporation and/or its affiliates

ORACLE®



Table of Contents

| | |
|--|----------|
| MySQL Enterprise High Availability..... | 1 |
| A Reference Guide..... | 1 |
| 1 Introduction | 3 |
| 2 MySQL Enterprise Edition | 3 |
| 3 Oracle MySQL Cloud Service | 4 |
| 4 An Introduction to MySQL Enterprise High Availability..... | 5 |
| 5 Infrastructure Considerations | 8 |
| 6 Software Stack | 8 |
| 7 Reference Architectures | 9 |
| 8 Developer Best Practices | 12 |
| 9 Read Scale Out | 14 |
| 10 Write Scale Out | 15 |
| 11 Fencing off Failed Instances | 15 |
| 12 Backups..... | 15 |
| 13 Monitoring..... | 16 |
| 14 Oracle Premier Support | 19 |
| 15 Conclusion | 20 |
| 16 Additional Resources | 20 |



1 Introduction

Whether you are building high volume websites, enterprise and departmental applications, or advanced communications networks, your organization needs the tools to build and manage these business-critical database applications. This paper explores how you can confidently deploy and operate MySQL Enterprise HA clusters, on-premises or in the Cloud, as part of a cost-effective cross-platform solution for delivering high-performing, highly available, disaster tolerant, reliable, and scalable applications. It examines some of the challenges associated with building and supporting scalable, data-driven applications and provides a detailed overview of how MySQL Enterprise HA can address these challenges. MySQL Enterprise Edition combines the most secure, scalable, “always on” version of the MySQL database with online backup, monitoring, management and visual database design and SQL development tools, all backed by Oracle Premier Support, 24x7 global enterprise-class support services. Further, MySQL Enterprise Edition supports your use of MySQL in conjunction with many of the Oracle products and tools you may already be familiar with or are currently using. MySQL Enterprise Edition is specifically designed to help you bring secure, high performing, highly available, and scalable MySQL applications to market faster, mitigate risk, and ensure you meet customer and end-user Service Level Agreements (SLAs).

2 MySQL Enterprise Edition

MySQL is the world's most popular open source database. Whether you are a fast growing web property, software vendor, a large organization or an SMB, MySQL can cost-effectively help you deliver high performance, scalable, and highly available database applications. If you are currently using MySQL, you probably started with MySQL Community Edition. In fact, in many instances MySQL enters an organization via an application development project and makes its way into the data center when the application is promoted for production use. Challenges arise when these applications become vital to business revenues or key business functions. The most common challenges around running MySQL and other open source technologies are revealed by a simple line of questioning:

- How will you ensure you are using the most reliable, secure, scalable, up-to-date version?
- How will you know:
 - If a server or application is down?
 - If there is a clustering, data replication, or consistency issue?



- If something is affecting the performance of a server?
- Will you know:
 - If MySQL is configured to scale-out as your application data and customer base grows?
 - If MySQL can survive regional outages and disasters?
 - How to configure MySQL variables to ensure your applications run at their peak performance?
- How will you:
 - Identify security loopholes in MySQL servers?
 - Know when there have been security alterations on a MySQL server?
 - Seamlessly add auditing compliance to your new and existing MySQL applications?
- How will you:
 - Optimize your database designs and queries before they are migrated into your production environments or included in your products?
 - Ensure HA clusters are configured correctly for availability, performance, and scalability?
 - Monitor and tune poorly performing user/application SQL code?
- How will you integrate MySQL with your existing security standards and infrastructure?

To help you answer these and other critical questions with confidence, MySQL provides MySQL Enterprise Edition. MySQL Enterprise Edition is a commercial offering comprised of the MySQL database with high availability, security, encryption, auditing, scalability extensions, online backup, monitoring, management, and visual database design and SQL development tools. MySQL Enterprise Edition is backed by Oracle Premier support for organizations delivering highly available, business critical applications and services.

For MySQL deployments in the cloud, Oracle also provides [Oracle MySQL Cloud Service](#).

3 Oracle MySQL Cloud Service

Oracle MySQL Service Cloud delivers a secure, cost-effective and enterprise-grade MySQL database service—with HA and DR provided as basic features of the service. Built on MySQL Enterprise Edition and powered by the Oracle Cloud, it provides the best in class management tools to automate administrative tasks such as provisioning, scaling, patching, backup & recovery, monitoring, and tuning. Multi-layered security protects your data against external attacks while helping you achieve regulatory compliance. Self-service provisioning and automated scaling



enables users to elastically scale compute resources, storage resources, and MySQL services.

The Oracle MySQL Cloud Service video will provide you a rapid overview:
<https://www.youtube.com/embed/xXUIHvAYtiE?autoplay=1>

Oracle MySQL Cloud Service makes it extremely easy to rely on MySQL either to migrate existing on-premises applications to the cloud or deliver new ones. You can:

- **Increase Business Agility:** Focusing your resources on innovation, not on infrastructure management.
- **Ensure Security, Performance & Uptime:** relying on the most comprehensive MySQL cloud platform – straight from the source.
- **Reduce Total Cost of Ownership:** Saving on infrastructure and database management operations costs while improving uptime.

Get our eBook “Oracle MySQL Cloud Service; Propel innovation and time-to-market”:

https://cloud.oracle.com/opc/paas/ebooks/Oracle_MySQL_Cloud_Service.pdf

Learn more about Oracle MySQL Cloud Service and sign up for a free trial:

<https://cloud.oracle.com/mysql>

4 An Introduction to MySQL Enterprise High Availability

The MySQL Database is a fully integrated transaction-safe, ACID compliant database with commit, rollback, crash-recovery and row level locking capabilities. The MySQL Database is a cost-effective solution for:

- High-performance scalable Web/Cloud/SaaS and E-commerce applications
- Corporate Departmental OLTP and Data Mart applications
- Operational data stores integrated with Big Data repositories and Stream Processing systems
- Low administration, high performance, reliable embedded database applications

MySQL has always been famous for its very flexible Master/Slave replication. However, MySQL users have been looking for a high availability



solution built into the MySQL Server. Unlike other products, MySQL Group Replication delivers native, built-in HA for your MySQL databases:

- **Core part of the standard MySQL Server**
- **Leverages proven MySQL features:** InnoDB, Global Transaction IDs (GTIDs), binary logs, multi-threaded replication execution, multi-source replication, Performance Schema
- **Supported on all MySQL platforms**

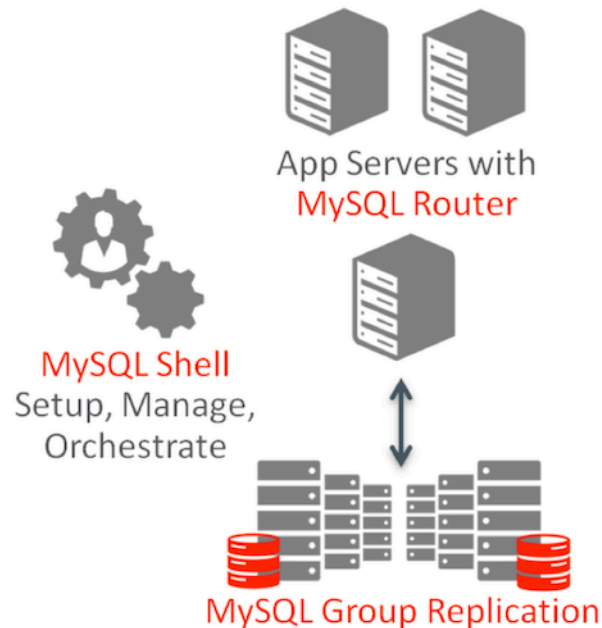


Figure 8: MySQL includes a native full stack, built-in HA solution

Group Replication implements both a [single-primary mode](#) with automatic leader election and a [multi-master update everywhere mode](#). By using a powerful new group communication system, which provides an in-house implementation of the popular Paxos algorithm, the group automatically coordinates on data replication, consistency, membership, and failure handling. This provides all of the built-in mechanisms necessary for making your MySQL databases highly available.

Elasticity

With Group Replication, a set of servers coordinate together to form a group. The group membership is dynamic and servers can leave, either voluntarily or involuntarily, and join at any time. The group will automatically reconfigure itself as needed and ensure that any joining member is synchronized with the group. This makes it easy to quickly scale your total database capacity up and down as needed. Router then hides all of this complexity from your applications and their developers, as the client connections are automatically routed to the correct server.



Failure Detection

Group Replication implements a distributed failure detector to find and report servers that have failed or are no longer participating in the group and the remaining members in the group coordinate to reconfigure the membership. Router then detects these changes and automatically adjusts the client connection routing.

Fault Tolerance

Group Replication builds on an in-house implementation of the popular Paxos algorithm to provide distributed coordination among servers. In order for a group to continue to function, it requires a majority of the members to be online and for them to form an agreement on every change. This allows your MySQL databases to safely continue to operate without manual intervention when failures occur, without the risk of data loss or data corruption.

Self-Healing

If a server joins the group, it will automatically bring itself up to date by synchronizing its state with an existing member. In the event that a server leaves the group, for instance it was taken down for maintenance, the remaining servers will see that it has left and will reconfigure the group automatically. When that server later rejoins the group, it will automatically re-synchronize with the group again.

Monitoring

MySQL Enterprise Monitor has full support for HA clusters; monitoring the configuration, health, and performance of each node and the cluster as a whole, providing best practice advice and alerts, along with easy to understand visual tools that allow you to easily monitor and manage all of your HA clusters.

HA for MySQL Client Applications using MySQL Router

MySQL Router allows you to easily migrate your standalone MySQL instances to natively distributed and highly available clusters without affecting existing applications. Router provides transparent client connection routing, load balancing, and failover into your HA clusters.

Easy and Accessible using MySQL Shell

MySQL Shell provides a single intuitive, flexible, and powerful interface for all MySQL related tasks. MySQL Shell makes it easy to create, monitor, and manage your HA clusters -- including MySQL Router instances -- in a self-describing natural language interface, without the need to understand low level concepts, configuration options, or other complex aspects. MySQL Shell makes HA clusters easy and accessible for everyone.

5 Infrastructure Considerations

- Hardware redundancy with 3-5 physical machines
 - Different motherboards
 - Different memory sub-systems
 - Different storage devices
- Reliable and fast network
 - Eliminate hubs, switches, and network hops
 - 10G+ Ethernet end-to-end between nodes
 - Dedicated network
 - Avoiding noisy neighbor issues
 - Avoiding general congestion issues
 - Providing more predictable performance
- Site redundancy
 - 1-3 data centers (sites/AZs/ADs)
 - Redundant cooling systems
 - Redundant internet backbone connections
 - Redundant power utility providers
- Regional redundancy (optional)
 - 1+ regional sites for disaster tolerance

6 Software Stack

- Operating System: Windows or Linux (see: [MySQL 8 Platforms](#))
- MySQL: MySQL 8 (Group Replication, Router, Shell)

7 Reference Architectures

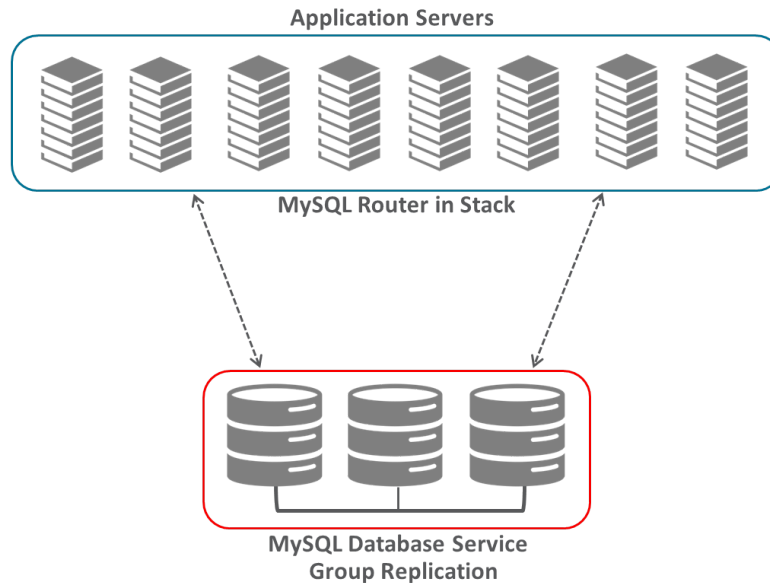


Figure 9: Shared Nothing Cluster – Single Data Center

This is the most basic production ready HA setup. This provides basic availability within a single data center and protects against any failure confined to a single machine. In order for this to be effective, ensure that the machines are not sharing physical resources. The OS images should be running on different physical boxes, those boxes should be in different racks, and the boxes would ideally have redundant systems for power, cooling, and networking.

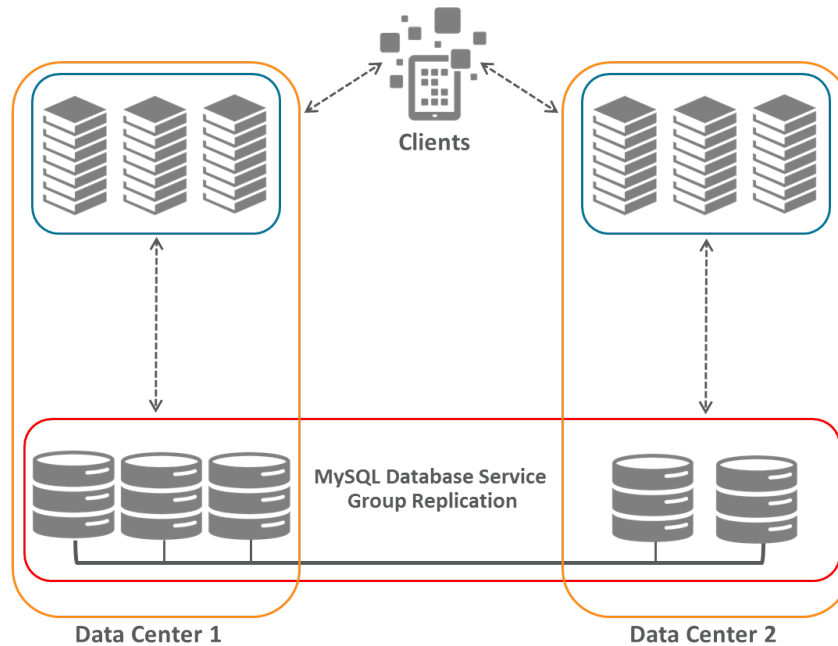


Figure 10: Shared Nothing Cluster – Cross Data Center

This deployment provides additional availability. Now even if we lose Data Center 1 (DC1) entirely, we can continue on with the cluster partition in Data Center 2 (DC2).

- **Note:** If DC1 fails, a [manual step is required to unblock the minority partition](#) in DC2
- **Note:** The network connection between DC1 and DC2 should be reliable and have low latency in order to avoid node evictions.

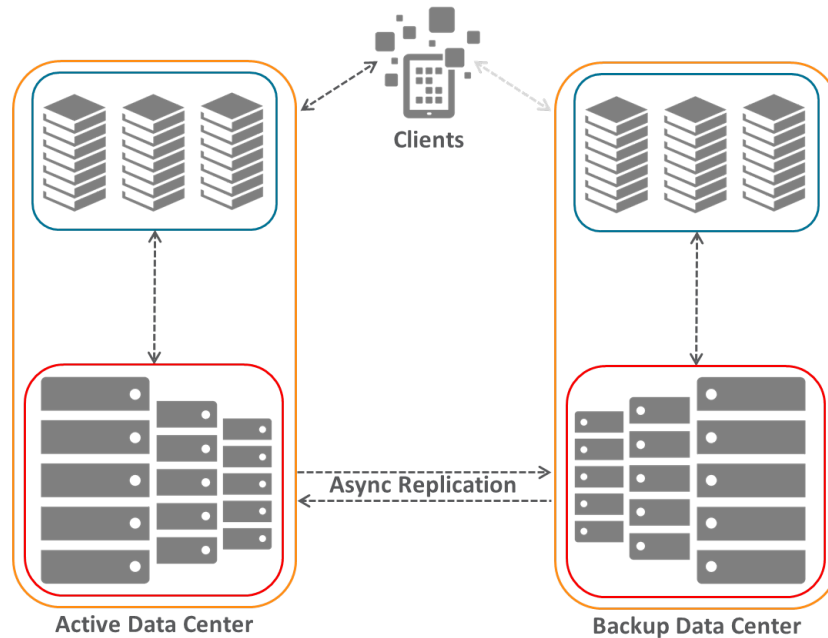


Figure 11: Geographically Redundant Cluster

This deployment provides disaster tolerance as the data centers are a great distance from each other on the earth. So your database service can survive major regional disasters without interruption and without user intervention.

- **Note:** because these datacenters are far apart and communicate over a high latency connection, you would use asynchronous master/slave replication between each site and cluster
- **Note:** because asynchronous replication is in use, there is some potential for losing your most recent writes when losing a regional data center.

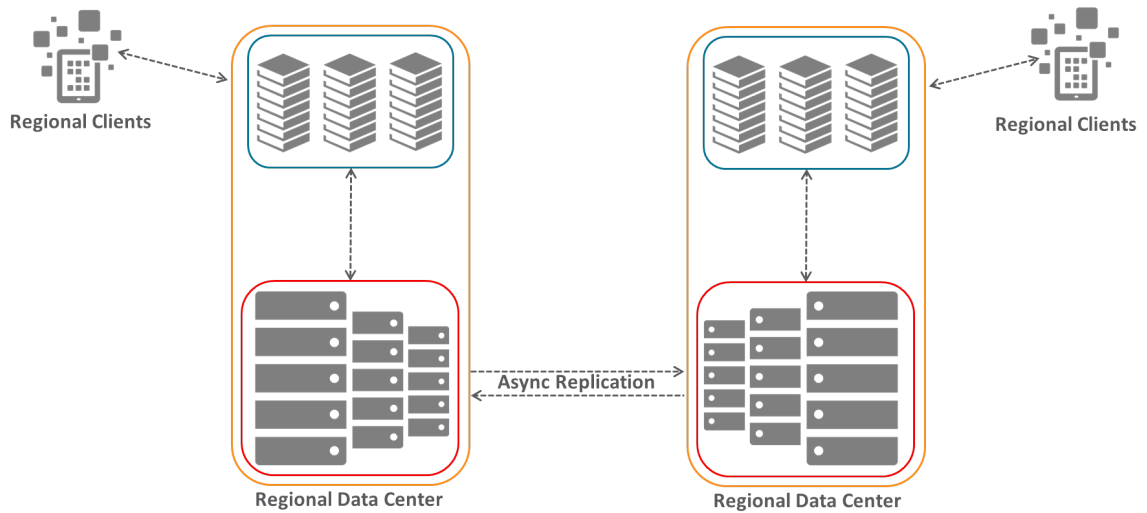


Figure 12: Active/Active Regional Multi-Data Center Setup

This setup provides high availability, disaster tolerance, and low latency access to users around the globe.

- **Note:** because these datacenters are far apart and communicate over a high latency connection, you would use asynchronous master/slave replication between each site and cluster
- **Note:** because master/slave connections are used between clusters—which have no built-in support for automatic conflict prevention, detection, and handling—you will need to ensure that the writes are conflict free at the application layer.

8 Developer Best Practices

- Catch all client connection errors, and re-connect. This is because Router only does any routing today for *new* connections. So to be safe you can catch all client errors and re-connect whenever you get one. In particular, be prepared to encounter and handle the following errors:
 - Error: 2006 ([CR_SERVER_GONE_ERROR](#)) -- “MySQL server has gone away”
 - Description: this means that the backend connection was lost. Most likely the result of a failure.
 - Action: re-connect so that Router will automatically connect you to the current PRIMARY
 - Error: 2013 ([CR_SERVER_LOST](#)) -- “Lost connection to MySQL server during query”

- Description: this means that the backend connection was lost. Most likely the result of a failure.
 - Action: re-connect so that Router will automatically connect you to the current PRIMARY
- Error: 2055 ([CR_SERVER_LOST_EXTENDED](#)) -- “Lost connection to MySQL server at '%s', system error: %d”
 - Description: this means that the backend connection was lost. Most likely the result of a failure.
 - Action: re-connect so that Router will automatically connect you to the current PRIMARY
- Error: 1836 SQLSTATE: HY000 ([ER_READ_ONLY_MODE](#)) -- “Running in read-only mode”
 - Description: this means that we’re *not* currently connected to a valid PRIMARY instance. You could see this if you have an open connection to a node that *was* the PRIMARY but has recently been ejected from the group due to an ephemeral issue (e.g. temporary network hiccup).
 - Action: re-connect so that Router will automatically connect you to the current PRIMARY
- Error: 3101 SQLSTATE: 40000 ([ER_TRANSACTION_ROLLBACK_DURING_COMMIT](#)) – “Plugin instructed the server to rollback the current transaction.”
 - Description: this means that a transaction failed the group certification process, due to one or more members detecting a potential conflict, and was thus rolled back.
 - Action: re-try your transaction
- Error: 3635 SQLSTATE: HY000 ([ER_APPLIER_LOG_EVENT_VALIDATION_ERROR](#)) – “The table in transaction %s does not comply with the requirements by an external plugin.”
 - This will happen if you attempt to execute DML against a table which fails to meet the [requirements](#)—most commonly the table is using a storage engine other than InnoDB or it lacks a defined Primary Key.
 - Action: examine [the table](#) noted in the error log
- If you're using any sort of client connection pool, then use a custom pool "healthcheck" query and have that run every second on each connection in the pool and/or executed on a connection before handing it out from the pool. Most connection pools (e.g. JDBC app server

connection pools) provide these options, and if the healthcheck query fails then one or all of the connections in the pool are closed. It's common to use `'SELECT 1'`—which just ensures that you have a database listening and responding on the other end—but in our case we want to know that the database on the other end is also currently a PRIMARY ONLINE member of the HA cluster. Two (5.7 compatible) examples of specific healthcheck queries that can be used are:

```
-- Is the node an ONLINE R/W member of a single or multi-primary
-- cluster with a write quorum
SELECT IF( CONCAT(
  (SELECT
    IF( MEMBER_STATE='ONLINE' AND ((SELECT COUNT(*) FROM
      performance_schema.replication_group_members WHERE MEMBER_STATE != 'ONLINE') >=
      ((SELECT COUNT(*) FROM performance_schema.replication_group_members)/2) = 0),
      'YES', 'NO' ) FROM performance_schema.replication_group_members JOIN
      performance_schema.replication_group_member_stats USING(member_id)
    WHERE member_id=(SELECT @@server_uuid)
  ),
  IF( (SELECT (SELECT GROUP_CONCAT(variable_value) FROM
    performance_schema.global_variables WHERE variable_name IN ('read_only',
    'super_read_only')) != 'OFF,OFF'), 'YES', 'NO'
  )
) = 'YESNO', 1, 0) as endpoint_status;

-- Is the node an ONLINE PRIMARY member of a cluster
SELECT IF(
  MEMBER_STATE='ONLINE' AND
  (member_id=(select variable_value from performance_schema.global_status
    WHERE variable_name="group_replication_primary_member")), 1, 0
) as endpoint_status
FROM performance_schema.replication_group_members INNER JOIN
performance_schema.replication_group_member_stats USING(member_id)
WHERE member_id=(SELECT @@server_uuid);
```

9 Read Scale Out

MySQL HA clusters are not only about availability, they can also be leveraged for read scale out. Router makes this easy for application developers—for your R/O workloads you connect to the R/O port and it transparently load balances these connections across the SECONDARY nodes in the cluster.

- **Note:** in a default configuration, the system is write optimized so as to reduce the write latency on the PRIMARY node. This means, however, that there's a chance that reads on SECONDARY nodes could be stale due to a (generally small) lag in the application of transactions.
- **Note:** you can choose to make the system read optimized—using the [group_replication_synchronous_writes](#) option—and you pay a penalty in write latency and throughput on the PRIMARY, but in return you have read-your-writes guarantees across the cluster. So your



application developers need not be concerned about any potential for stale reads.

10 Write Scale Out

MySQL HA clusters are primarily about ensuring that your database applications are always available. As noted, they can also be used for read scale out. However, for write scale out more is needed. Partitioning your data across a set of HA clusters—where each cluster manages a subset of the data—is needed. Then you can scale your writes as needed by further partitioning your dataset. Today this type of setup is often called a sharded cluster. While MySQL Enterprise HA does not have native support for sharded clusters today, it is on our roadmap. So stay tuned!

11 Fencing off Failed Instances

This is one area where you need to choose between availability and consistency using the [group_replication_exit_state_action](#) option. Group Replication provides two options for how to handle failed nodes:

1. Make the MySQL instance read-only
2. Kill the MySQL instance (which in some cases will allow the node to re-join the cluster automatically)

Option 1 will allow the cluster to continue serving reads in spite of multiple cluster failures, such as a total loss of write quorum, but it will also allow a failed node to provide stale reads. By default, Group Replication will kill an instance that has failed (entered the ERROR state) and exited the cluster in order to ensure read consistency.

12 Backups

MySQL Enterprise Backup performs online "Hot", non-blocking backups of MySQL HA clusters. Full backups can be performed while the cluster and all its MySQL instances are online, without interrupting queries or updates. Incremental and partial backups are supported, along with consistent Point-in-Time Recovery (PITR), compression, and direct cloud native storage.

- **Note:** Backups of an HA cluster can be performed via a PRIMARY or SECONDARY member of the cluster. But since SECONDARY members are read-only, for backup jobs executed on a SECONDARY no information about that backup job is recorded in the



local backup status and history tables and thus that backup job will not be visible to MySQL Enterprise Monitor.

13 Monitoring

MySQL developers, DBAs, and SysAdmins often find themselves having to manage dynamic, high growth applications that require continuous uptime. The MySQL Enterprise Monitor helps them automate the management of their MySQL infrastructure and improve the performance and availability of their applications.

The MySQL Enterprise Monitor is a web-based application that can manage MySQL HA clusters within the safety of a corporate firewall or remotely in a public cloud. MySQL Enterprise Monitor provides:

- **Performance & Availability Monitoring** - Continuously monitor MySQL health and performance related metrics
- **Visual Query Analysis** – Monitor query performance and pinpoint SQL code that could cause a slow-down
- **InnoDB Monitoring** – Monitor key InnoDB metrics that impact MySQL performance
- **Replication Monitoring** – Gain visibility into the performance, and health of all MySQL Masters and Slaves
- **HA Cluster Monitoring** – Monitor and manage your distributed and highly available MySQL setups with ease
- **Backup Monitoring** – Ensure your online, hot backups are running as expected
- **Firewall Monitoring** – Monitor MySQL Enterprise Firewall activity and protect your MySQL servers against database attacks
- **Audit Monitoring** – Enforce MySQL Enterprise Audit usage across all of your MySQL servers
- **Disk Monitoring** – Forecast future capacity requirements using trend analysis and projections
- **Security Monitoring** – Identify and resolve security vulnerabilities across all MySQL servers
- **Operating System Monitoring** – Monitor operating system level performance metrics such as load average, CPU usage, RAM usage and swap usage
- **Access Control Lists (ACLs)** – Role and group based access controls provide an easy way for MySQL DBAs to define fine-grained security policies for their MySQL servers



One Enterprise Dashboard for Monitoring all MySQL Assets

The Enterprise Dashboard provides real-time visibility into the performance and availability of all your MySQL installations. You can monitor MySQL related metrics for a single server or groups of servers, and manage all forms of replication and clustered topologies. The Enterprise Dashboard is designed to give you an overall view of the health of your MySQL installations and point you to potentially critical issues that may impact the performance and availability of your applications. A rich set of real time and historical graphs allow you to drill down into relevant statistics and details.

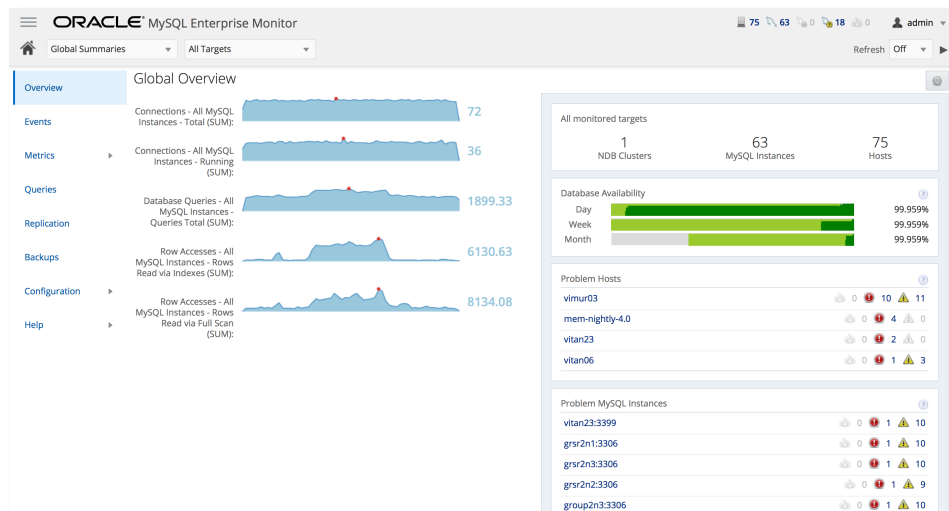


Figure 10: MySQL Enterprise Overview Dashboard

Monitoring of Replicated and Clustered Topologies

The Replication and Topology Dashboards make it easier to scale out using MySQL replication and clustering by providing industry-leading auto detection, grouping, documenting and monitoring of all combinations of MySQL Master/Slave Replication and MySQL HA clusters. Changes and additions to existing topologies are also auto detected and maintained providing you instant visibility into changes. This helps reduce the learning curve for anyone new to MySQL Replication and MySQL HA clusters or to scale-out and high availability environments in general.

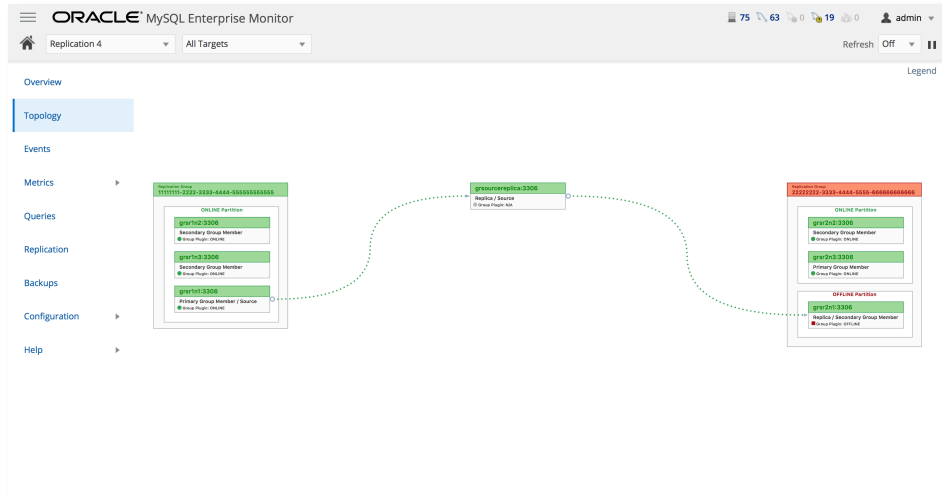


Figure 11: MySQL Enterprise Replication Topology view

Whether you use a single-source tree hierarchy, circular replication, HA clusters, or a complex, multi-level, multi-source hierarchy that mixes all of the above, the Topology view shows how your MySQL Topology is currently operating.

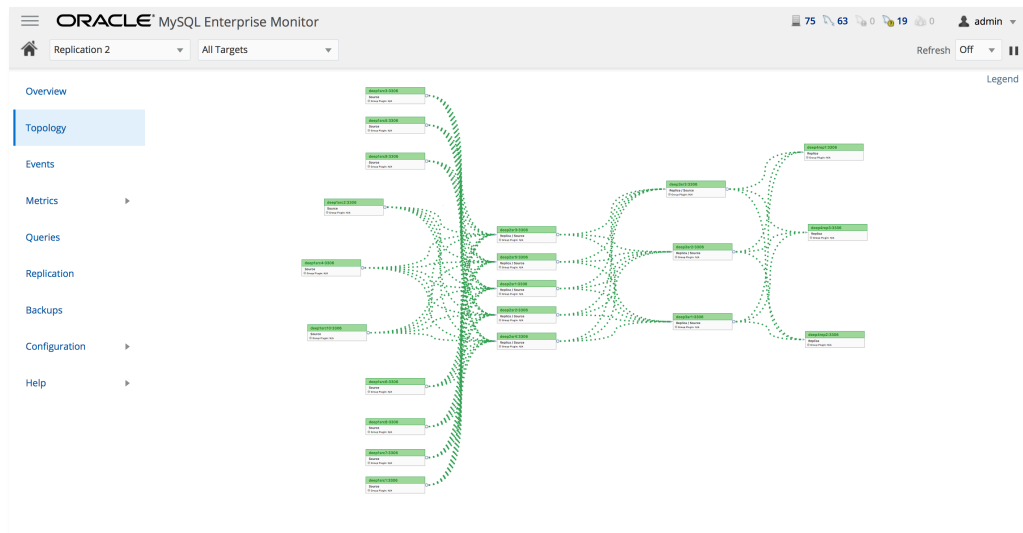


Figure 12: Multi-source Replication topology view



The Topology and Replication Dashboards show context-sensitive information for each node and node type, as well as HA. High-level identity and throughput metrics are always displayed. For each Replication Source, you'll see Binary Logging and GTID status, or, if enabled, statistics on Semi-Sync Replication or Group Replication performance. For each Replica, you'll see the status of each Replication Channel, statistics on Relay Logging, Multi-Threaded Replication and other source configuration details.

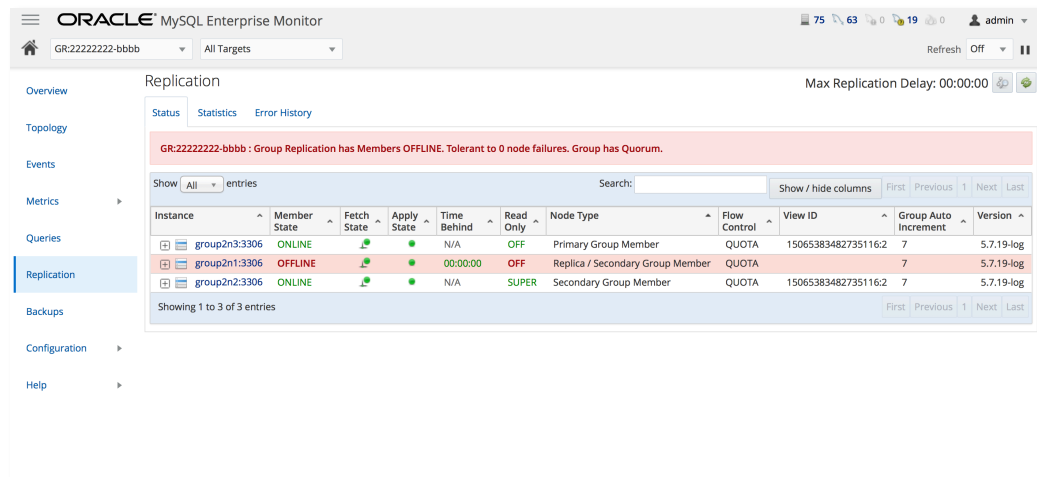


Figure 13: MySQL Replication Dashboard Status View

14 Oracle Premier Support

Oracle offers 24x7, global support for MySQL. The MySQL Support team is composed of seasoned MySQL developers, who are database experts and understand the issues and challenges you face. With Oracle Premier Support, you can lower the total cost and risk of owning your MySQL databases, improve the return from your IT investment, and optimize the business value of your IT solutions. MySQL support is included in the subscription for end users, and available separately from commercial licenses for ISVs and OEMs. Oracle Premier Support for MySQL includes the following features:

- 24 X 7 production support
- Unlimited support incidents
- Knowledge Base
- Maintenance releases, bug fixes, patches and updates
- MySQL Consultative support
- Staffed by the most experienced MySQL Engineers in the industry
- The ability to get MySQL support in 29 languages

MySQL Consultative Support service is included in Premier Support. MySQL Consultative Support is a proactive approach that is designed to



help you avoid critical outages. MySQL Support Engineers advise you on how to properly setup and tune your MySQL servers, schema, queries, and HA clusters to maximize performance and availability. Also, by taking the initiative to properly design and tune your MySQL database applications you can avoid having to purchase expensive hardware for your IT infrastructure.

Learn more about Oracle Premier Support:

<http://mysql.com/support/>

15 Conclusion

In this paper we explored the components that make up MySQL Enterprise High Availability. These components are designed to help you mitigate risk and meet Service Level Agreements (SLAs) as you implement highly available and disaster tolerant applications on MySQL. In addition to High Availability, MySQL Enterprise Edition extends the MySQL Database to include advanced Auditing, Security, Encryption, and Performance/Scale features. The MySQL Enterprise Backup performs online "Hot", non-blocking backups of your MySQL HA clusters. The Enterprise Monitor, with its smart Advisors and Query Analyzer, proactively notify you of potential problems and service improvement opportunities before issues impact your applications and customers.

You can also consume [MySQL Enterprise as a service in Oracle Cloud](#). Built on the proven MySQL Enterprise Edition and powered by the Oracle Cloud, [Oracle MySQL Cloud Service](#) provides a simple, automated, integrated and enterprise-ready cloud service that enables organizations to deliver MySQL-based applications globally at scale.

16 Additional Resources

<http://www.mysql.com/>



<https://cloud.oracle.com/mysql>

MySQL Enterprise Edition Demo:

<https://www.youtube.com/watch?v=ypQh9H9Rf9w>

To contact an Oracle MySQL Representative:

<http://www.mysql.com/about/contact/>