MySQL Reference Architectures for High Availability

Introduction

Information Technology (IT) is critical for enterprises seeking to gain a competitive edge, reduce expenses, strengthen customer relationships, and improve decision-making. As businesses become increasingly reliant on their IT infrastructure, uninterrupted availability is paramount. The MySQL Reference Architectures for High Availability offers a framework of best practices for deploying High Availability (HA) and Disaster Recovery (DR) solutions, guaranteeing this essential availability.

MySQL Reference Architectures for High Availability

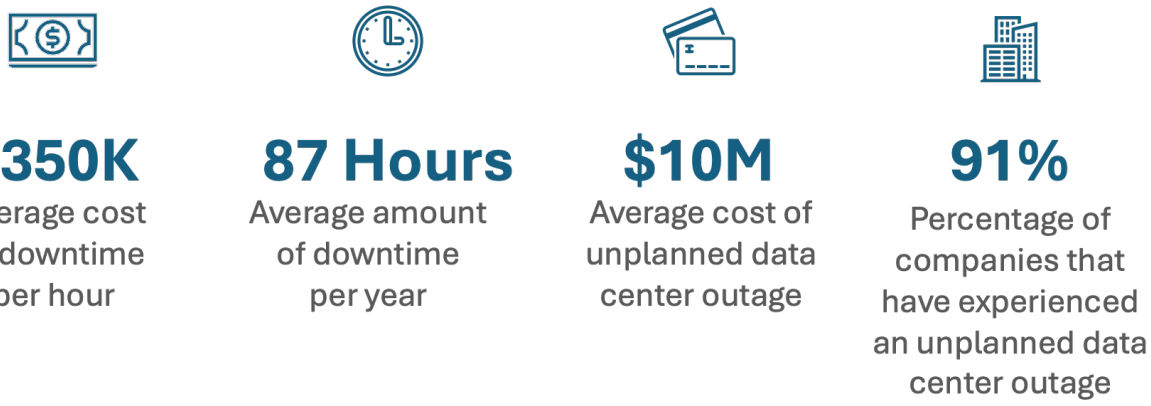
Blueprints for reduced planned and unplanned downtime for MySQL

The MySQL Reference Architectures for High Availability provides architecture, configuration, and lifecycle best practices for MySQL Databases and applications, enabling high-availability service levels for databases and applications residing in on-premises, cloud, or hybrid configurations. `

MySQL offers a choice of standard reference architectures --Bronze, Silver, Gold, and Platinum -- for high availability, data protection, and disaster recovery. Each reference architecture, or high availability tier, uses an optimal set of capabilities that, when deployed together, reliably achieve target service levels for unplanned outages and planned maintenance events.

The MySQL Reference Architecture for High Availability represents a tested architecture used to ensure that end-to-end application and database availability is preserved, or at its optimal levels, for any fault or maintenance event. Specifically, the MySQL Reference Architectures for High Availability introduces various faults and planned maintenance events to evaluate application and database impact.

Impact and Cost of Downtime



Downtime is common and very costly.

IT downtime represents a significant risk that businesses must mitigate through robust IT infrastructure, proactive monitoring, and effective disaster recovery plans.

MySQL Reference Architectures for High Availability and Chaos Engineering

Breaking things to ensure your peace of mind

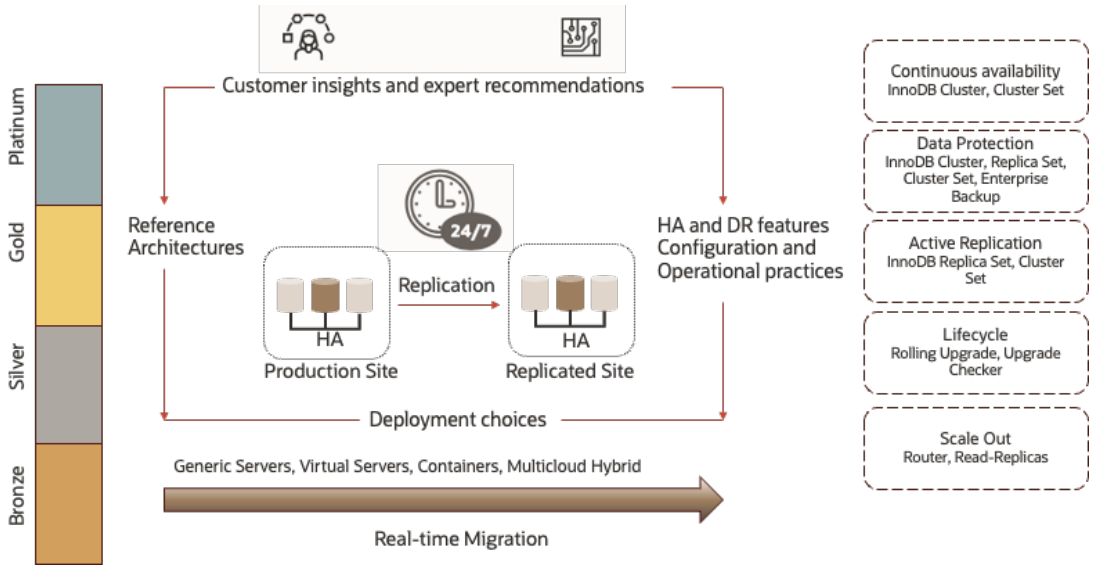
Chaos Engineering is the art form of experimenting (i.e. proactively breaking things) on a system to build confidence in a system's resilience to withstand turbulent events in production:

- Network, server & storage failures
- Human errors & data corruption
- Data corruption
- Power failures or site failures (i.e. Godzilla attack or hurricane)
- Application, database & server software updates
- Data reorganization or changes
- Application changes and optimizations

Investing in Chaos Engineering practices to test these diverse scenarios is a strategic imperative for organizations seeking to maintain continuous operation and minimize the impact of unforeseen events.

MySQL Reference Architectures for High Availability

Standard Reference Architectures for Never-Down Deployments



Various MySQL availability solutions map to a business' availability requirements.

MySQL High Availability for Multiple Workloads

The MySQL Reference Architectures for High Availability serves as a best-practice framework for achieving the highest levels of database availability. It thoroughly examines various failure scenarios and proposes solutions leveraging MySQL's native high availability and disaster recovery capabilities.

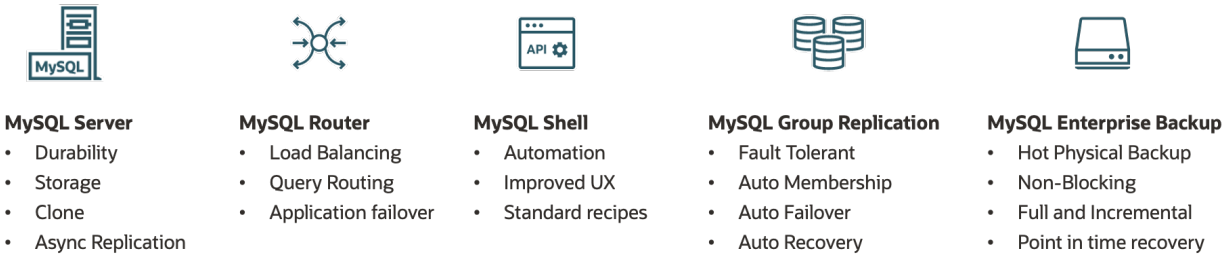
This framework is valuable for both organizations looking to improve existing MySQL deployments and those designing new systems, by providing insights into potential failures and maintenance requirements. Additionally, MySQL Reference Architectures for High Availability guides application developers on how to design applications to gracefully handle specific failures, while utilizing Application Continuity for transparent management of others.

Monitor, Operate, Automate

MySQL can handle many different high availability workload patterns. The MySQL Reference Architectures for High Availability provides references for a variety of deployment architectures. These architectures use the following native built-in MySQL high availability replication technologies to simplify deployment:

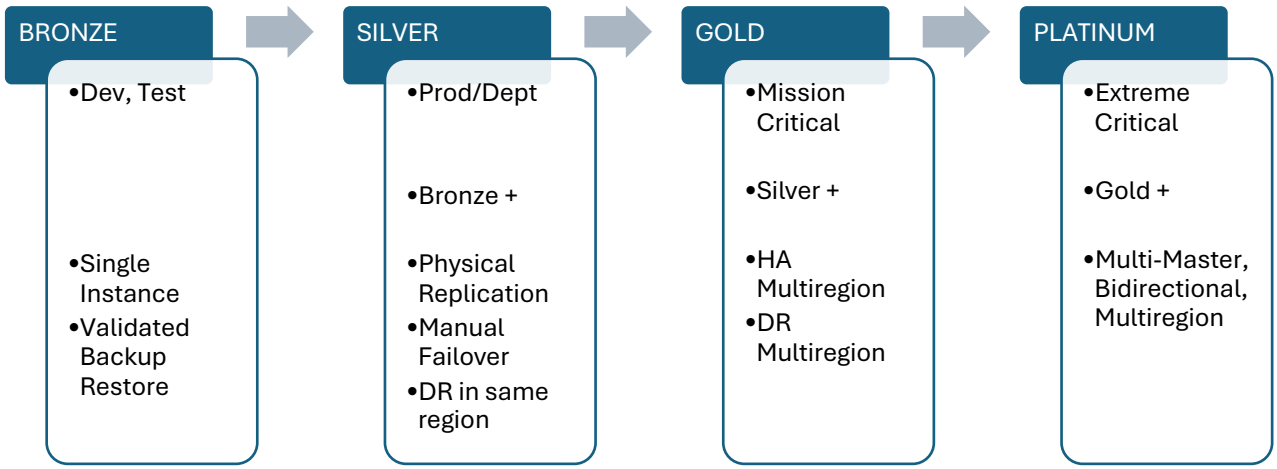
- **InnoDB Cluster** (Active/Active, Automatic Group Membership, Automatic Failover)
- **InnoDB ReplicaSet** (Primary & Replica Servers within a single region)
- **InnoDB ClusterSet** (Primary & Replica Clusters across multiple regions)

In addition, MySQL provides a powerful set of products and tools to configure, monitor and operate of these high-availability solutions.



A comprehensive set of integrated tools to manage high availability environments.

MySQL Reference Architectures for High Availability



Customers can map business use cases to reference architecture solutions.

Each tier uses a different reference architecture to deploy the optimal set of high availability capabilities that reliably achieve a given service level at the lowest cost and complexity. The tiers explicitly address all types of unplanned outages, including data corruption, component failure, and system and site outages, as well as planned outages due to maintenance, migrations, or other purposes.

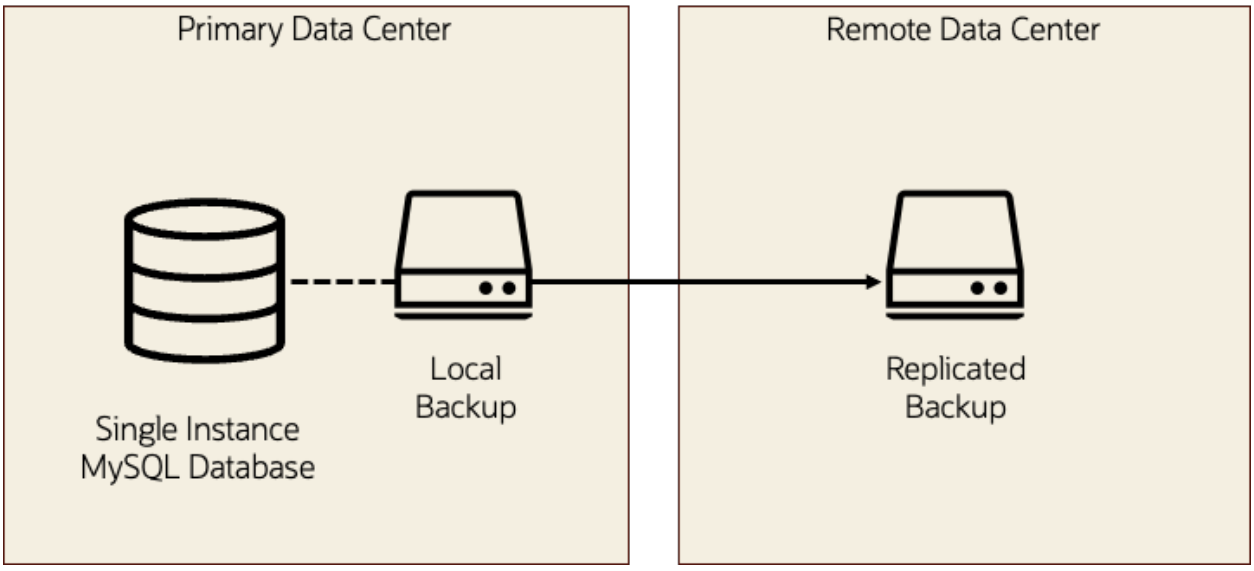
Reference Architecture	Unplanned Outages (Local Site)	Planned Maintenance	Data Protection	Unrecoverable Local Outages and Disaster Recovery
Bronze	Single Instance, auto-restart for recoverable instance and server failures.	Some online, most off-line	Basic runtime validation combined with manual checks	Restore from backup, potential to lose data generated since the last backup.
Silver	HA with simple manual failover for instance and server failures	Most rolling, some online, few offline	Basic runtime validation combined with manual checks	Promote standby Replica to Primary Restore from remote backup, potential to lose data generated since the last backup.
Gold	Comprehensive high availability and disaster recovery	All rolling or online	Comprehensive runtime validation combined with manual checks	Real-time failover, zero to near-zero data loss
Platinum	Zero application outage for Platinum ready applications	Zero application outage	Comprehensive runtime validation combined with manual checks	Zero application outage for Platinum-ready applications, with zero data loss.

BRONZE Reference Architecture

The Bronze tier is appropriate for databases where simple restart or restore from backup is "HA and DR enough."

Dev, Test, Prod - Single Instance Database with Backups

- Single Instance
- Enterprise Backup
- Binlog Backups with MySQL Binlog
- PITR Recovery
- In-place upgrade
- Automatic restart via system



Bronze – Local and Replicated Remote Backup Storage.

The MySQL Bronze reference architecture provides basic database service at the lowest possible cost. A reduced level of high availability and data protection is accepted in exchange for reduced cost and implementation complexity. This architecture may be suitable for databases used for test, development, and less critical production applications and databases.

Bronze uses the backup capabilities included in MySQL Enterprise Edition. For logical corruptions such as human error, you can use MySQL Binlog along with MySQL Enterprise Backup restore operations to “rewind” the database and roll it forward to a specific point in time. In the worst-case scenario of a complete site outage, there is additional time

required to restore and recover the system and database from remote backups which may result higher downtime.

In Bronze, a local backup within the same data center is always recommended for the fastest recovery. Additionally maintaining a second copy of backups in a remote data center to protect against site outages and disasters. You can use MySQL Enterprise Backup support for cloud object stores to maintain a cloud-based backup of on-premises databases.

The Bronze level of service illustrates the tradeoff between reduced implementation and maintenance costs and expected downtime during planned and unplanned outages.

Bronze Level of Service

Unplanned Outage	RTO/RPO Service Level Objectives [1]
Recoverable node or instance failure	Minutes to hour
Disasters: corruptions and site failures	Hours to days RPO since last backup or near zero with MySQL Binlog Backups
Planned Maintenance	
Software/hardware updates	Minutes to hour
Major database upgrade	Minutes to hour

Data Protection Tools

Product/Tool	Technology
MySQL Enterprise Backup	Physical Backup – Fast, Minimal to no Locking, Archival
MySQL Shell – Dump and Load	Logical Backup aka Export – Logical, Flexible
File System Specific	File System Snapshots – Fast, Rigid, Non-Archival
MySQL InnoDB Instance Cloning	Database Storage Snapshot (Cloning)
MySQL Binlog	Large, Continuous, Roll forward recovery

Bronze Backup Strategies

Method	Backup Factors	Recovery Factors
1. Full Backups	Longest backup times Largest storage space • Save space with compression	Easy to recover Faster restore time RPO – data loss to last Full Backup
2. Full + Incremental	Shortest backup time Reduced storage requirements Requires additional production storage	Finer-grained recover • End of last Incremental Backup Slower restore times • First restore full • Then restore Incremental RPO – data loss to last Incremental
3. Full + Incremental + Log	Added Storage requirements	Finest-grained recovery Slowest Restore times • First restore full • Then restore Incremental • Then apply logs RPO – data loss near 0, end of binlog
4. Replicated backup copy	Copy files to remote storage	RPO based on method and last copy time

SILVER Reference Architecture

The Silver tier provides an additional level of high availability for databases that require minimal or zero downtime in the event of database instance or server failure, as well as most common planned maintenance events, such as hardware and software updates.

Production/Departmental

- MySQL InnoDB ReplicaSet
- Hot Standby
- Manual Failover
- Enterprise Backup
- Binlog Backups with MySQL Binlog
- PITR Recovery
- In-place upgrade
- Automatic restart via system

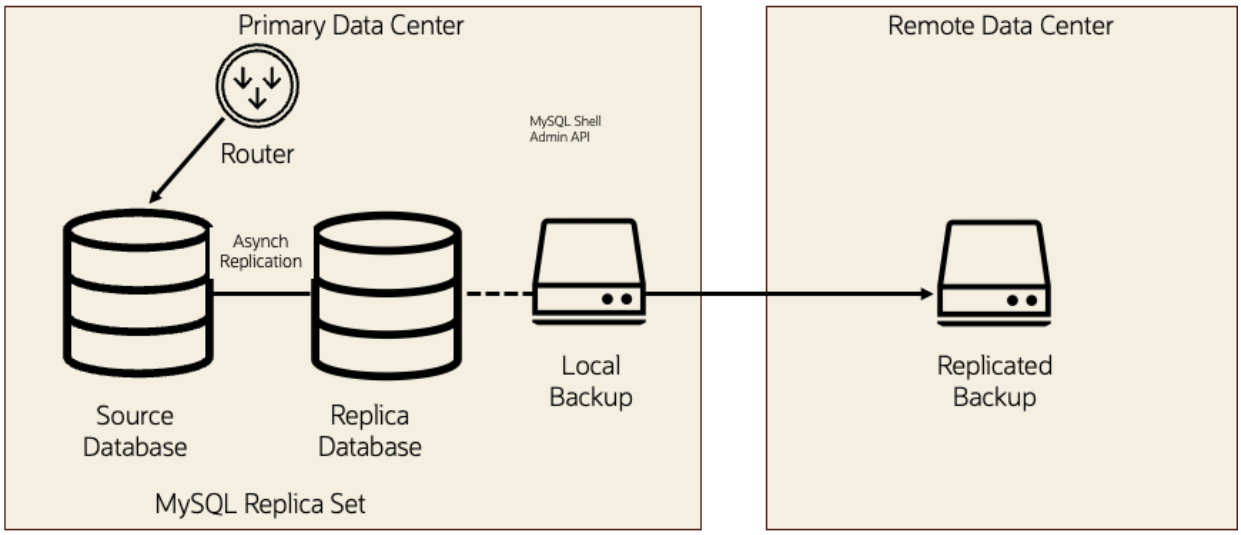
BRONZE +

- MySQL InnoDB Replica Set
- Primary Data Center Replica
- Read-Only Scale Out (Optional)

MySQL InnoDB Replica Set
Fully integrated solution leverages

- MySQL Shell
- MySQL Router
- MySQL Server
- MySQL Async Replication

SILVER
Hot Standby with Remote Backup



Silver – Local Standby Replica with Local and Replicated Remote Backup Storage.

Silver is designed for databases that can't afford to wait for a cold restart or a restore from backup, should there be an unrecoverable database instance or server failure. This architecture may be suitable for production applications that are business critical and need to reduce downtime for local failures and most common planned maintenance activities.

Silver is built on the foundation of the Bronze architecture and adds MySQL InnoDB Replica Set for minimal downtime in the event of database instance or server failure, as well as minimal database downtime for most common planned maintenance events.

Just like in Silver, MySQL Enterprise Backup provides database-optimized backups to restore availability should there be a complete Replica Set outage or disaster. This level of service lets you dramatically decrease expected downtime for hardware failures and brings most planned downtime due to software and hardware upgrades down to a minimum, when compared to the Bronze level of service.

Silver Level of Service

Unplanned Outage	RTO/RPO Service Level Objectives [1]
Recoverable node or instance failure	Minutes – Requires Manual Failover
Disasters: corruptions and site failures	Hours to days RPO since last backup or near zero with MySQL Binlog Backups
Planned Maintenance	
Software/hardware updates	Zero [2]
Major database upgrade	Minutes to hour

[1] RPO is zero unless explicitly specified
[2] To achieve zero downtime or lowest impact for online processing, apply application checklist best practices. For long running transactions such as batch operations, it's best to defer outside the planned maintenance window.

Silver Backup Strategy

Method	Backup Factors	Recovery Factors
All Bronze Plus		
Offload backup to replicas	Frees master from backup workload	Fast Failover to Replica Backups are last resort In the event of site failure, need to recover from backups and apply logs.

GOLD Reference Architecture

Mission Critical

The Gold reference architecture is well suited for service level requirements that cannot tolerate long periods of downtime and data loss. This set of architecture patterns provides

high availability and comprehensive data protection for all types of unplanned outages, including data corruptions, database failures, and site outages. Mission critical production applications that require quick recovery time and zero or minimal data loss for all database and system outages and planned maintenance activities will benefit from the capabilities included in the Gold reference architecture.

Gold, building on Silver, provides you with **three architecture patterns.**

The architecture patterns vary from a single remote active standby to multiple standby database configurations and standby reader farms.

Gold Level of Service

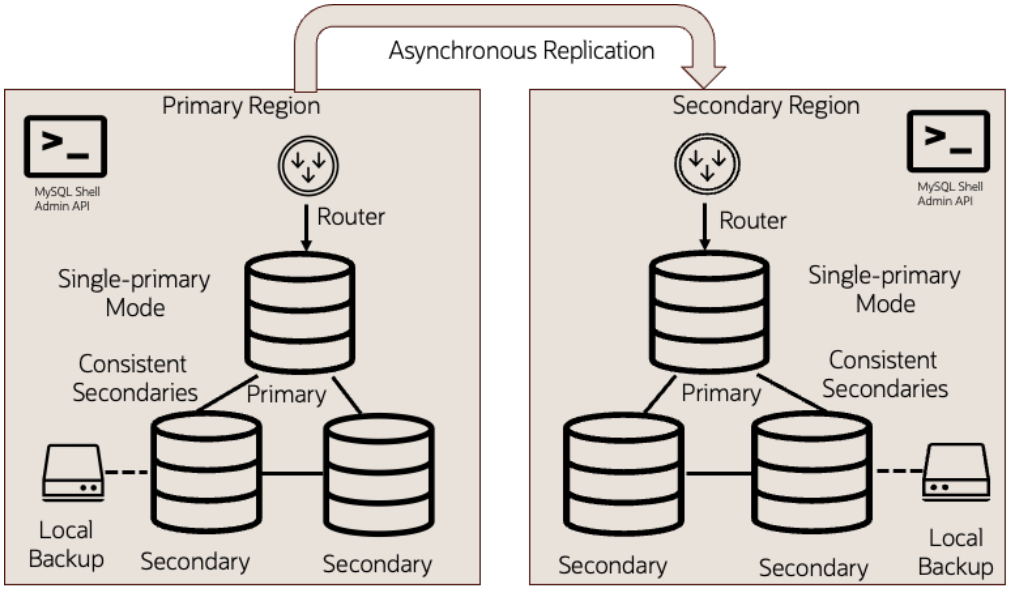
Unplanned Outage	**RTO/RPO Service Level Objectives [1]**
Recoverable node or instance failure	Single Digit seconds [2]
Disasters: corruptions and site failures	Hours to days RPO since last backup or near zero with MySQL Binlog Backups
Planned Maintenance	
Software/hardware updates	Zero [2]
Major database upgrade	Minutes to hour

[1] RPO is zero unless explicitly specified
[2] To achieve zero downtime or lowest impact for online processing, apply application checklist best practices. For long running transactions such as batch operations, it's best to defer outside the planned maintenance window.

Gold – Pattern 1 - Remote Standby

The Gold Remote Standby pattern includes a remote synchronized copy of the production database (Standby Database) using MySQL ClusterSet to eliminate single point of failure. The active standby database provides a high level of protection from unplanned outages and reduces downtime for planned maintenance activities, such as database upgrades.

The most notable attribute is that the standby provides low RTO (recovery time) and RPO (data loss potential) in the case of a disaster such as a database, cluster, or site failure.



Gold – Local HA with Replicated HA in secondary region.

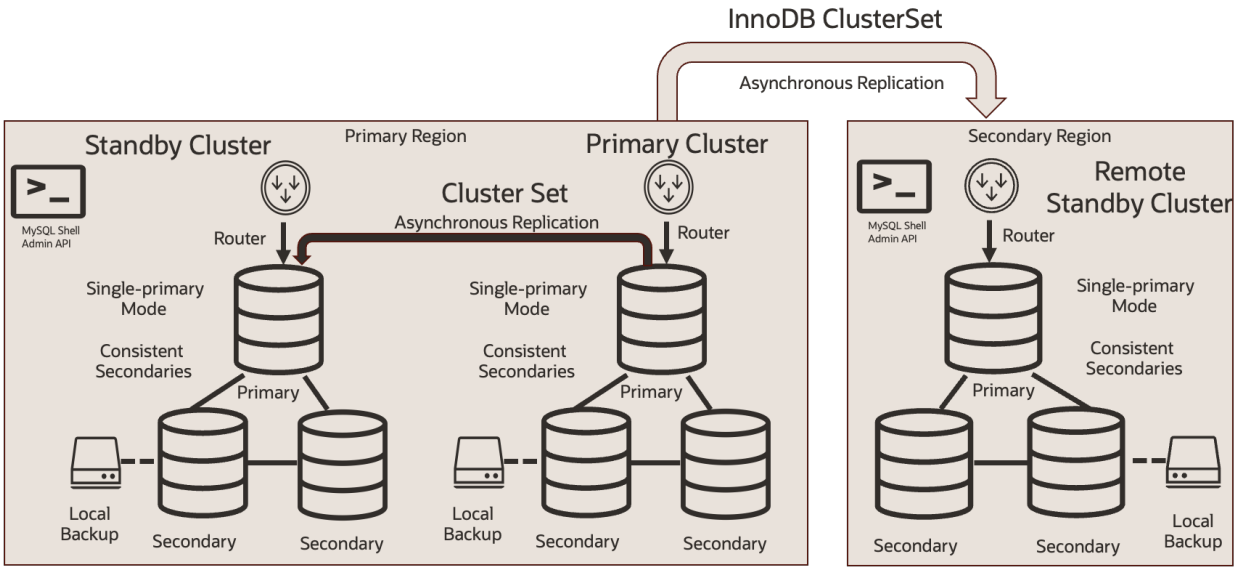
Gold – Pattern 2 - Multiple Stand-by Databases

Automatic failover to a local standby in the same region provides you with significant local disaster isolation and application failover simplicity. The local standby can be located in a separate fault domain or availability domain from the primary database within the same region. Application failover in this architecture pattern follows the recommendations described in Continuous Availability - Application Checklist for Continuous Service for MAA Solutions.

The business value of a local standby HA database is seen in zero data loss failover and application downtime reduced to seconds. By enabling synchronous HA, a zero data loss configuration becomes more viable due to the lower latency between primary and standby database systems. Applications pointing the MySQL Router automatically and transparently fail over to the local standby, maintaining the same latency between application servers and the database, which is particularly important for OLTP applications and package applications, because higher latency can significantly impact throughput and overall application response time.

If a regional disaster occurs, making the primary and local standby systems inaccessible, the application and database can fail over to the remote standby. Even though database downtime is still very low when regional disaster occurs, the application downtime can be higher due to additional orchestration required for failover operations to the secondary region.

To make the secondary region symmetric, you can add another standby in that region. Another variation is to add additional standby databases for reporting.

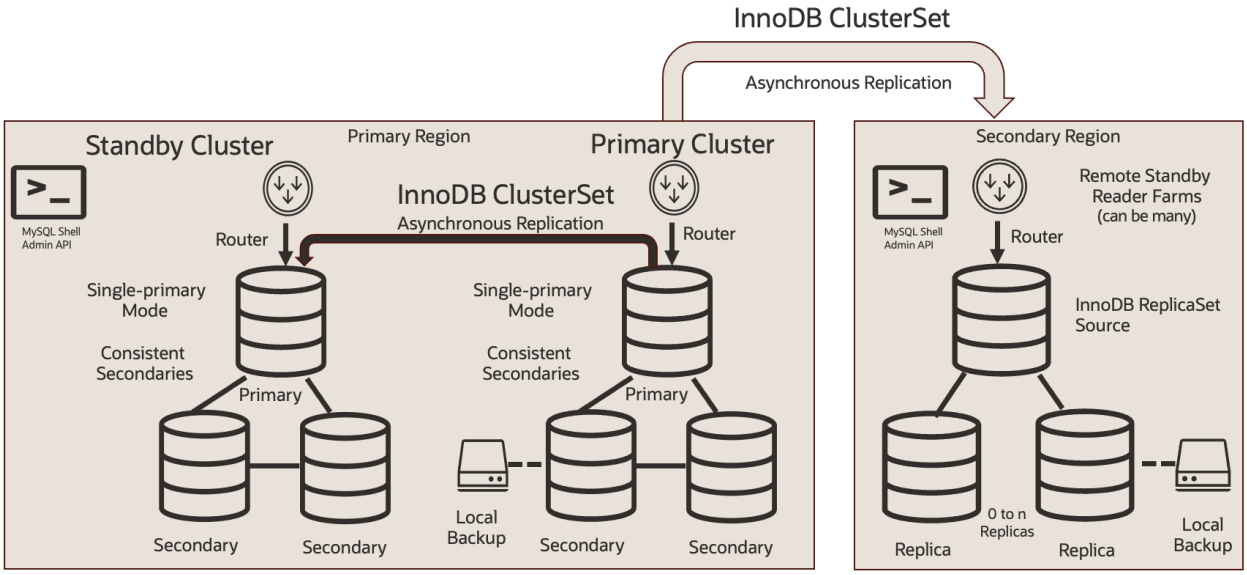


Gold – Local HA with Replicated HA in Primary Region and Replicated HA in Secondary Region.

Gold – Pattern 3 - Remote Standby Reader Farm

The Gold Standby Reader Farm pattern provides all the benefits of the Gold Multiple Standby Databases pattern, plus it allows read-only operations to scale across many standby databases for local and regional reader farm scalability.

The Gold Remote Standby pattern includes a remote synchronized copy of the production database (Standby Database) using MySQL Cluster Set to eliminate single point of failure. The active standby database provides a high level of protection from unplanned outages and reduces downtime for planned maintenance activities, such as database upgrades. The most notable attribute is that the standby provides low RTO (recovery time) and RPO (data loss potential) in the case of a disaster such as a database, cluster, or site failure.



Gold – Local HA with Replicated HA in Primary Region. Replicated HA in Secondary Region with Scale-our Read Replicas.

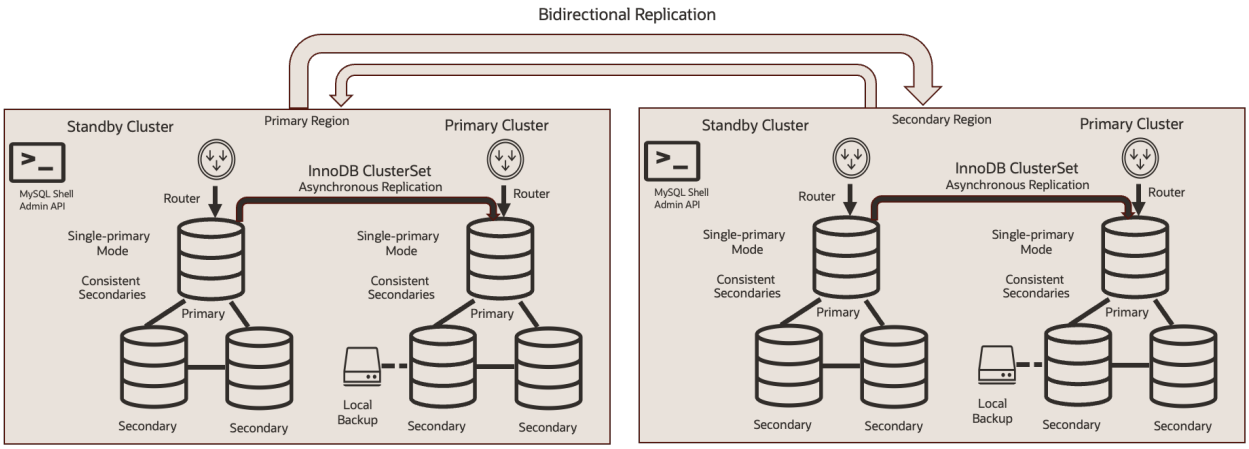
PLATINUM Reference Architecture

Extremely Critical

MySQL InnoDB ClusterSet
HA with Remote Replica(s)
Same DC – Automatic Failover
Remote DC – Manual Failover

Multiple Stand-by Databases

The Platinum reference architecture has the potential to provide zero downtime for outages and planned maintenance activities that are not achievable with Gold. Platinum builds on Gold by adding MySQL bi-directional asynch replication to eliminate downtime for migrations, application upgrades, and database upgrades. Each MySQL database is protected by a standby database to enable zero or near data loss in case of database, cluster, or site failure.



Platinum – Multi-master, bidirectional HA across multiple regions.

Benefits:

- Uni-directional or bi-directional replication allowing reads and updates in any replica if desired activities (note that the applications must be customized for Mult-master to avoid conflicts).
- Replicas can run on different platforms, database versions, or database or application configurations, allowing for online migration and database and application upgrades.
- Replicas are online, so applications can switch over with zero downtime during outages and planned maintenance activities (note that the application switchover must be customized, rather than built in as it is with Application Continuity).

Unlike the other reference architectures, application considerations are required to integrate into this architecture, especially if there's a need to switch over to other replicas. Custom application service management may be required to achieve zero or minimum application downtime for activities such as migration, database upgrade, or site switch when one replica is down. Also, if multiple replicas are updated concurrently at any point, conflict detection and resolution must be configured.

To address zero downtime application upgrade, the best solution is to make application changes on an alternative primary database replica and then switch over from the primary database to the alternative primary database replica when all transactions are synchronized.

As shown in the table below, the Platinum level of service addresses the most mission critical requirements and delivers zero data loss and highest uptime potential.

Platinum Level of Service

Unplanned Outage	RTO/RPO Service Level Objectives [1]
Recoverable node or instance failure	Zero to Seconds [2,3]
Disasters: corruptions and site failures	Zero [3]
Planned Maintenance	
Software/hardware updates	Zero [2]
Major database upgrade	Zero [3]

[1] RPO is zero unless explicitly specified
[2] To achieve zero downtime or lowest impact for online processing, apply application checklist best practices. For long running transactions such as batch operations, it's best to defer outside the planned maintenance window.
[3] Application failover is custom

Conclusion

Not explicitly listed, every database needs to run on a reliable system platform. Monitoring databases and systems is critical to proactively detect, prevent, and recover from issues before they have an availability impact. Both Oracle Enterprise Manager and OCI Observability and Management tool are the MySQL strategic monitoring platforms. Lastly, Oracle Cloud HeatWave works collaboratively and continuously to incorporate all the MySQL reference architectures, configuration best practices, and life cycle operations.

Appendix: Terminology, Technology, Tools, Guidelines

Terminology - Faults, Failures and Errors

Types of Failure

High Availability
- Single Server Failure, Network Partition

Disaster Recovery
- Full Region/Network Failure

Human Error
- Little Bobby Tables

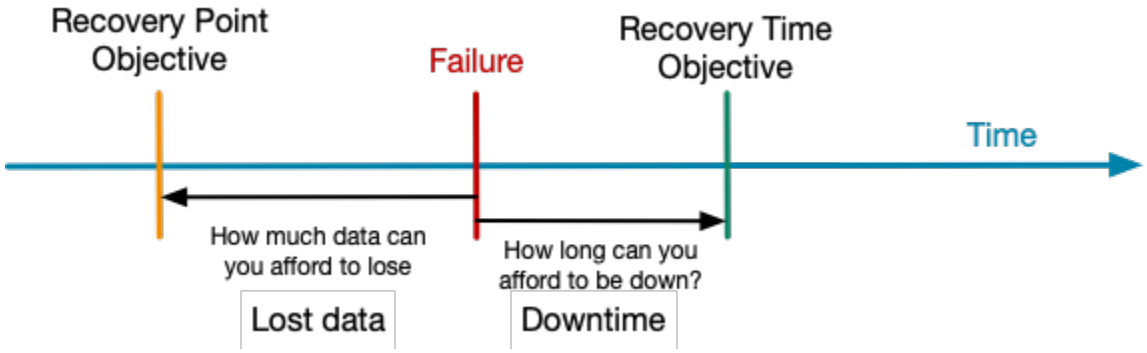
Terminology - RPO, RTO and SLAs

Concepts

RTO: Recovery Time Objective
- How long does it take to recover from a single failure

RPO: Recovery Point Objective
- How much data can be lost when a failure occurs

SLA: Service Level Agreement
- RTO and RPO heavily influence SLAs



RPO, RTO relative to Failure and Data Loss.

Technology: MySQL Asynchronous Replication

Replication enables data from one MySQL database server (known as a source) to be copied to one or more MySQL database servers (known as replicas). Replication is asynchronous by default; replicas do not need to be connected permanently to receive updates from a source. Depending on the configuration, you can replicate all databases, selected databases, or even selected tables within a database.

Advantages of replication in MySQL include:

- **Scale-out solutions** - spreading the load among multiple replicas to improve performance. In this environment, all writes and updates must take place on the source server. Reads, however, may take place on one or more replicas. This model can improve the performance of writes (since the source is dedicated to updates), while dramatically increasing read speed across an increasing number of replicas.

- **Data security** - because the replica can pause the replication process, it is possible to run backup services on the replica without corrupting the corresponding source data.

- **Analytics** - live data can be created on the source, while the analysis of the information can take place on the replica without affecting the performance of the source.

- **Long-distance data distribution** - you can use replication to create a local copy of data for a remote site to use, without permanent access to the source.

Documentation: https://dev.mysql.com/doc/refman/en/replication.html

Technology: MySQL Group Replication

MySQL Group Replication enables you to create elastic, highly-available, fault-tolerant replication topologies. Groups can operate in a single-primary mode with automatic primary election, where only one server accepts updates at a time. Alternatively, groups can be deployed in multi-primary mode, where all servers can accept updates, even if they are issued concurrently.

There is a built-in group membership service that keeps the view of the group consistent and available for all servers at any given point in time. Servers can leave and join the group and the view is updated accordingly. Sometimes servers can leave the group

unexpectedly, in which case the failure detection mechanism detects this and notifies the group that the view has changed. This is all automatic.

Group Replication guarantees that the database service is continuously available. However, it is important to understand that if one of the group members becomes unavailable, the clients connected to that group member must be redirected, or failed over, to a different server in the group, using a connector, load balancer, router, or some form of middleware. Group Replication does not have an inbuilt method to do this. For example, see MySQL Router 8.4.

Technology: Hot Backup

MySQL Enterprise Backup component, lets you perform a “Hot backup” a running MySQL instance, including InnoDB tables, with minimal disruption to operations while producing a CONSISTENT SNAPSHOT of the database.

By Hot we mean that when mysqlbackup is copying InnoDB tables, reads and writes to InnoDB tables can continue.

In conjunction with the MySQL binary log, users can perform point-in-time recovery. For logical backups such as mysqldump or mysql shell dump, the backup will not be consistent without locking tables.

Technology: Distributed Database Fault Tolerance with Paxos

MySQL Group Replication builds on an implementation of the Paxos distributed algorithm to provide distributed coordination between servers. As such, it requires a majority of servers to be active to reach quorum and thus make a decision.

For more information see Group Replication and Paxos.

Technology - Fast restart

To reduce the warmup period after restarting the server, InnoDB saves a percentage of the most recently used pages for each buffer pool at server shutdown and restores these pages at server startup. The percentage of recently used pages that is stored is defined by the innodb_buffer_pool_dump_pct configuration option.

For more information see Saving and Restoring the Buffer Pool State

Tool Set: MySQL Shell

MySQL Shell processes code written in JavaScript, Python and SQL.

Interactive Mode

Type code at the MySQL Shell prompt and each entered statement is processed, with the result of the processing printed onscreen.

Batch Mode

Take code from different sources and process it. This method of processing code in a noninteractive way is called Batch Execution.

APIs

Includes the APIs implemented in JavaScript and Python which you can use to develop code that interacts with MySQL.

For more information see MySQL Shell

Tool Set: MySQL Shell Admin API

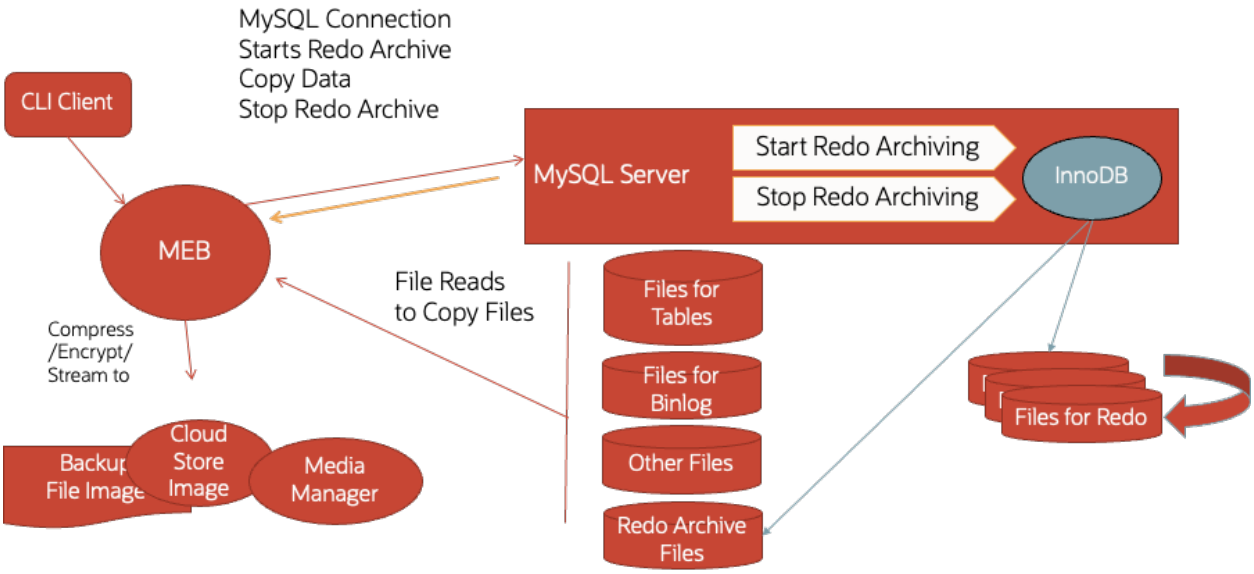
AdminAPI enables you to administer MySQL instances, using them to create InnoDB Cluster, InnoDB ClusterSet, and InnoDB ReplicaSet deployments, and integrating MySQL Router.

For more information see Using MySQL AdminAPI

Tool Set - MySQL Enterprise Backup (MEB)

For more information see MySQL Users Backup Guide

Architecture



MySQL Enterprise Backup and Restore Architecture Diagram

Guideline - Rolling Patching for Complex Environments

- For more information see InnoDB Cluster Upgrade

Guideline - Single Server Upgrades

- See For more information see MySQL In-place upgrades

Guideline – Security

- MySQL Maximum Security Architecture
- MySQL Center for Internet Security Benchmark
- MySQL DISA STIG
- MySQL Security Guidelines

Guideline – Replication from OnPrem to OCI HeatWave

- OCI documentation on inbound replication

Guideline – Load Balancing

- Connection Routing
- Using Replication for Scale-Out