Amazon RDS for MySQL and MariaDB Best Practices

Darin Briskman

Technical Evangelist, Amazon Web Services

briskman@amazon.com or @briskmad

AWS

Offering you the broadest choice of fully-managed MySQL compliant engines

Standard

The open source standard MySQL

Community

The popular community choice

Performance

The fastest MySQL compatible engine

If you host your databases on-premises

- App optimization
- Scaling
- High availability
- Database backups
- DB s/w patches
- DB s/w installs
- OS patches
- OS installation
- Server maintenance
- Rack & stack
- Power, HVAC, net

If you host your databases in Amazon EC2

- App optimization
- Scaling
- High availability
- Database backups
- DB s/w patches
- DB s/w installs
- OS patches

- OS installation
- Server maintenance
- Rack & stack
- Power, HVAC, net

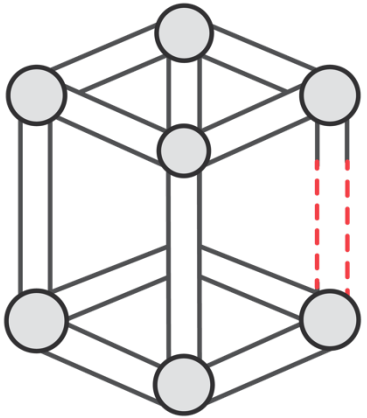
If you choose Amazon RDS

App optimization

- Scaling
- High availability
- Database backups
- DB s/w patches
- DB s/w installs
- OS patches
- OS installation
- Server maintenance
- Rack & stack
- Power, HVAC, net

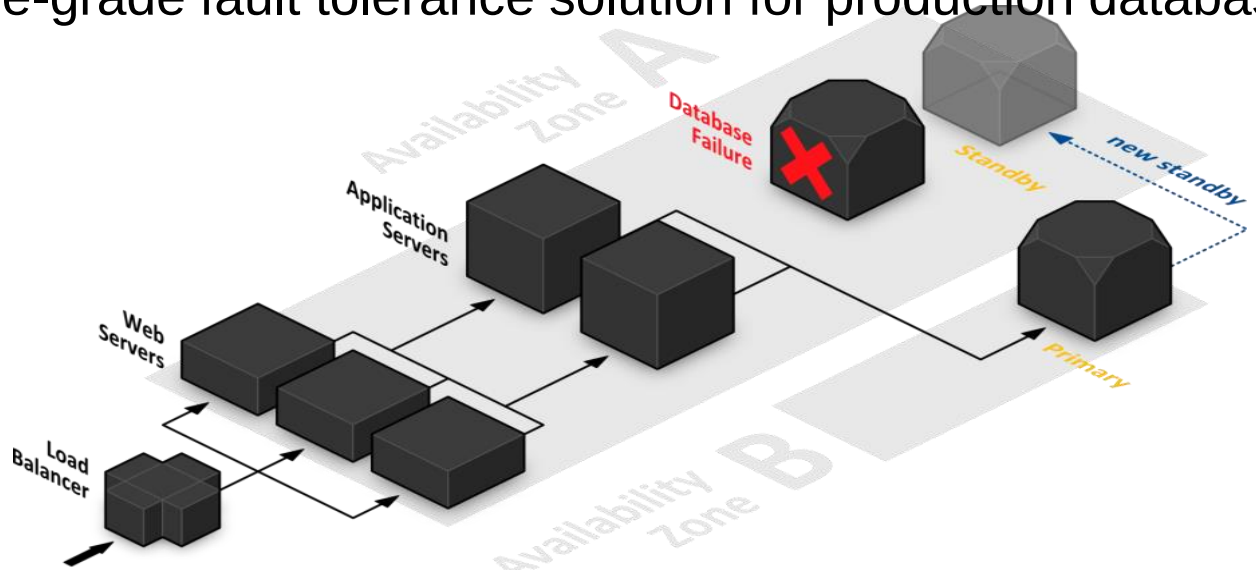
Designed to be better

- Designed to have built-in high availability and cross-region replication across multiple data centers
- Managed backup and point-in-time recovery
- Push button provisioning, automated instance and storage scaling, patching, security, restores, and general care and feeding
- Lower TCO because we manage the muck
  - ✓ Get more leverage from your teams
  - ✓ Focus on the things that differentiate you



Designed for high availability

Enterprise-grade fault tolerance solution for production databases

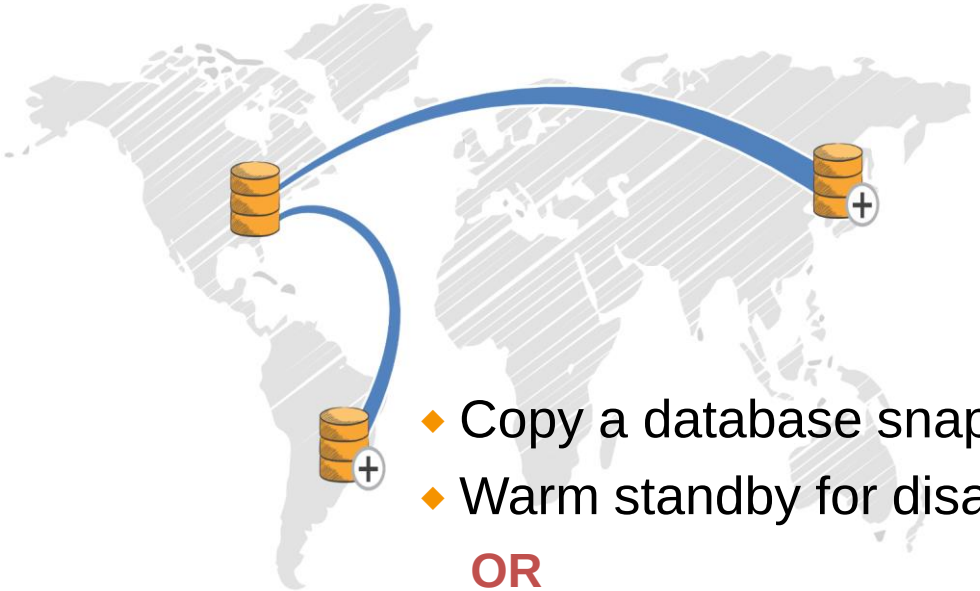


- An Availability Zone is a physically distinct, independent infrastructure
- Your database is synchronously replicated to another AZ in the same AWS region
- Failover occurs automatically in response to the most important failure scenarios

Cross-region read replicas

- Promote read-replica to a master for faster recovery in the event of disaster
- Bring data close to your customer's applications in different regions
- Promote to a master for easy migration

Cross-region snapshot copy



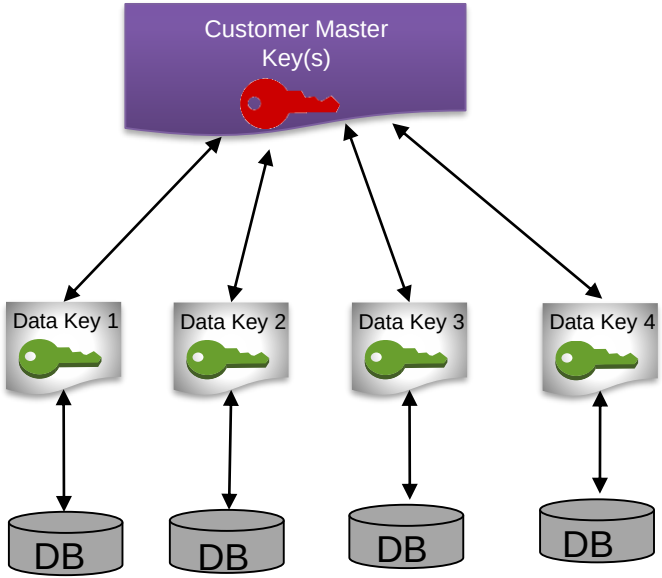
- Copy a database snapshot to a different AWS region
- Warm standby for disaster recovery

  OR

- Use it as a base for migration to a different region

Levels of security difficult to achieve on-premises

- RDS offers transparent encryption at rest and SSL protection for data in transit
- Amazon RDS gives each database instance IP firewall protection
- Amazon VPC lets you isolate and control network configuration and connect securely to your IT infrastructure
- AWS Identity and Access Management (AWS IAM) provides resource-level permission controls



Two-tiered key hierarchy using envelope encryption

AWS Database Migration Service

- Move data to the same or different database engine
- Keep your apps running during the migration
- Start your first migration in 10 minutes or less
- Replicate within, to, or from Amazon EC2 or RDS

RDS MySQL

- Most popular open-source database engine
- Support for MySQL Community Edition versions 5.5, 5.6 and 5.7
- InnoDB and MyISAM storage engines
- Version 5.7 - New Features
  - **JSON support**
  - Query optimizer improvements
  - GIS extensions
  - Improved parallel replication
  - Dynamic buffer pool resizing
- Version 8.0 coming soon

RDS MariaDB

- Support for versions 10.0 and 10.1
- Same instance, regions, pricing as RDS MySQL (including free tier)
- Differences from RDS MySQL
  - XtraDB and Aria storage engines only
  - Thread Pooling
  - GTID
- Version 10.1 - New Features
  - XtraDB page compression, data scrubbing, and defragmentation
  - Optimistic in-order parallel replication
  - ORDER BY optimization
- Version 10.2 coming soon

MariaDB Audit Plug-in for MariaDB and MySQL

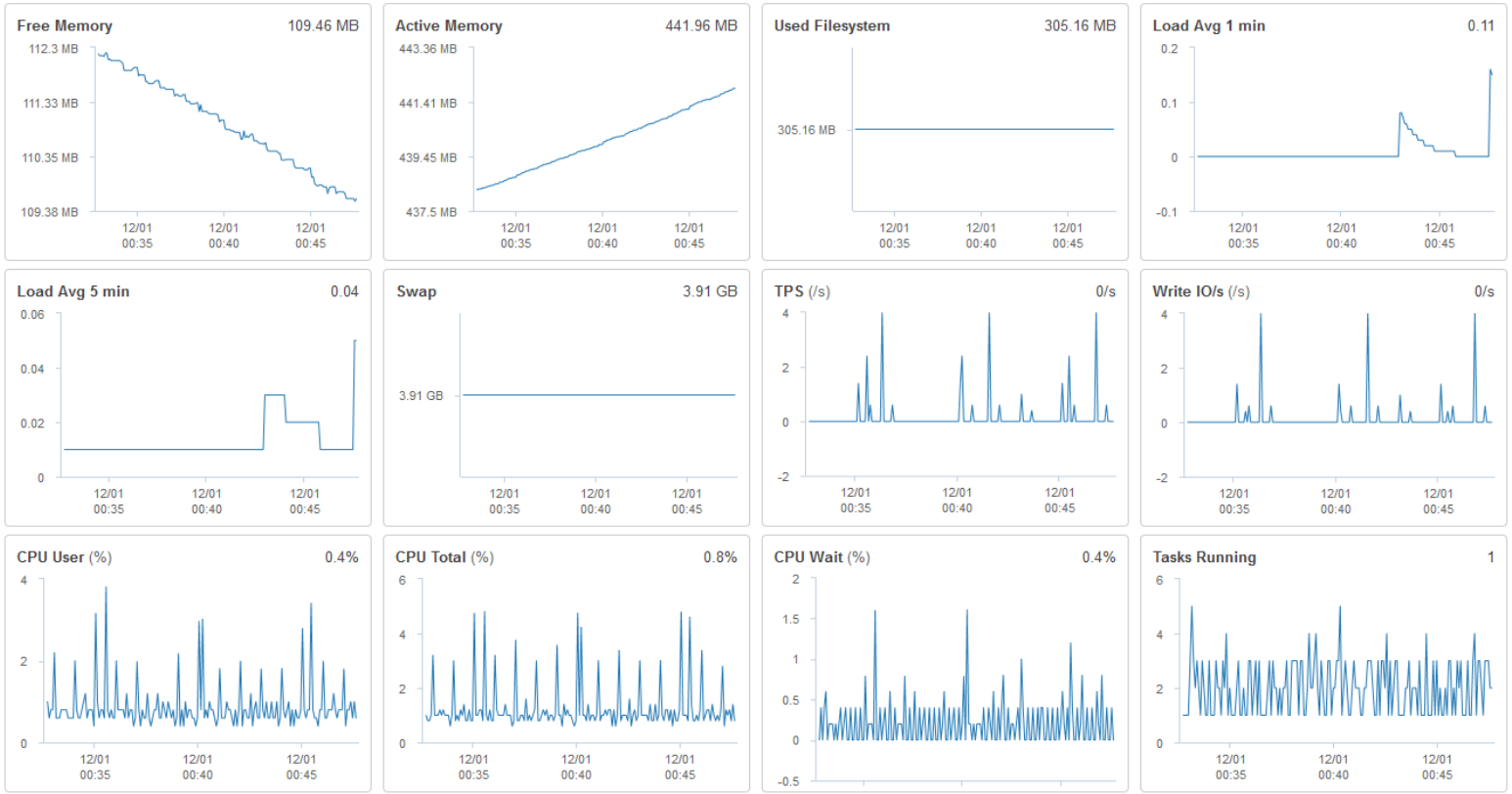
- Provides customer configurable event logging for database activity
  - Auditable events include logins, queries, and tables accessed
  - Individual users can be included or excluded from the audit
- The MariaDB audit plug-in is supported on the following RDS versions
  - MySQL 5.6.29
  - MySQL 5.7.11
  - MariaDB 10.0.24
  - MariaDB 10.1.14
- Available via RDS option group
- Can impact server performance (up to 10% penalty for full logs)

MySQL / MariaDB / All Engines Best Practices

- Leverage Multi-AZ Configurations
  - Pros: High availability for planned and unplanned events
  - Cons: Additional hourly charge, increased latency
- Leverage Read Replicas
  - Pros: Read scalability, disaster recovery, upgrades
  - Cons: Application must handle asynchronous behavior, watch log storage on master
- Leverage Enhanced Monitoring
  - Pros: Additional information for tuning and troubleshooting
  - Cons: CloudWatch Logs charge (small)

* Excluding micro instance classes

Enhanced Monitoring (All Engines)



50+ system/OS metrics | sorted process list view | 1-60 sec granularity

alarms on specific metrics | egress to CloudWatch Logs | integration with 3rd-party tools

MySQL / MariaDB Best Practices

- Use RDS provided stored procedures for managing sessions, queries, and read replicas
  - *mysql.rds_kill(processID)* to terminate a connection
  - *mysql.rds_kill_query(queryID)* to terminate a query
  - *mysql.rds_stop_replication* manually stop replication
  - *mysql.rds_skip_repl_error* to skip the current replication error
  - *mysql.rds_next_master_log* to change the master log position on the replica

MySQL / MariaDB Logging Best Practices

- Enable MySQL general and slow query logs for troubleshooting
  - The logs can be written to a CSV table or file
  - The *long_query_time* database parameter sets the threshold for slow queries
  - Avoid large CSV table logs by using stored procedures to rotate the logs
    - *mysql.rds_rotate_general_log*
    - *mysql.rds_rotate_slow_log*
  - Disable the general and slow query logs if they are not actively being used for diagnosis or troubleshooting
    - The general and slow query logs can impact database performance, especially if *long_query_time* is set to a low value

MySQL / MariaDB Replication Tips

- Replicas can be made writeable
  - Useful for changes like adding an index for reporting
  - Think of it as “breaking the warranty”
- RDS supports managed replication chaining (source > target > target)
  - Replica in one region and then a cross-region replica in another region
- RDS limits 5 replicas to per master (can be extended upon request)
- RDS instances support non-managed replication
  - Useful for on-premises to RDS, EC2 to RDS scenarios
  - Uses stored procedures rather than service API for managing
- RDS MariaDB provides crash-safe replication using Global Transaction Identifiers (GTIDs)

MySQL / MariaDB Storage Types

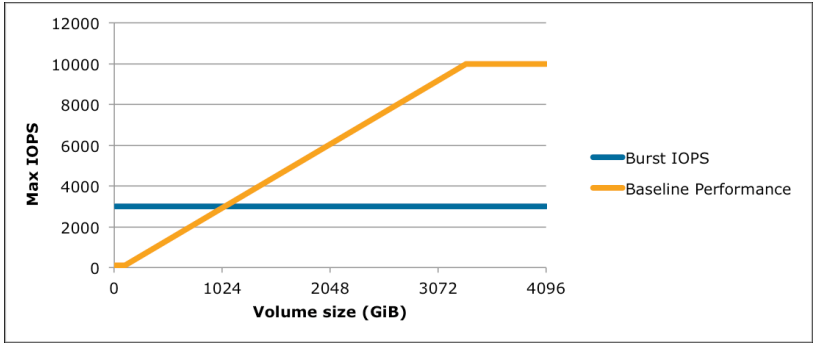
- The type of storage can make a big difference for I/O intensive operations
  - **Magnetic** – Low cost but IOPS and latency can vary
  - **General Purpose (GP2)** – SSD based storage with 3K IOPS burst capability then a baseline rate of 3 IOPS per GB. Throttled via a credit-based system. Great for storage below 1 TB especially when you do not deplete credits.
  - **Provisioned IOPS (PIOPS)** – SSD based storage with defined IOPS rates. Great when you need consistent performance or when you need very high performance.
  - For almost all use cases we recommend an SSD-based storage type
  - Magnetic storage has average IOPS and latencies that are an order of magnitude slower than SSD-based storage.
  - GP2 is the default and makes sense until you start to go above 1 TB.

MySQL / MariaDB Storage Best Practices

- Fast, consistent storage is important since many routine operations can be heavily I/O dependent
  - **Crash Recovery** – InnoDB/XtraDB needs to scan through certain files as well as rollback and roll-forward transactions.
  - **Engine Upgrades** – MySQL scans through all tablespaces during a minor or major engine version upgrade.
  - **DDL** – MySQL can do online DDL which means it is copying a potentially large table. Index creations and rebuilds also are I/O intensive.
- Multi-AZ failovers can be less than a minute with most of the downtime in DNS propagation
  - If MySQL needs to do a crash recovery or other I/O intensive operation before it can start, the speed of storage could mean downtime is several minutes or even hours

MySQL / MariaDB Storage Tips

- GP2 is a great choice but be careful about burst credits on volumes < 1TB
  - Hitting credit depletion results in IOPS drop. Latency and queue depth metrics will spike until credits are replenished.
  - Monitor read/write IOPS to see if average IOPS is greater than the baseline.
- Think of GP2 burst rate and PIOPS stated rate as maximum I/O rates.
  - Your application needs to actually run an I/O intensive workload to hit these rates
  - Multi-AZ introduces an extra commit latency so MAZ IOPS will typically be lower compared to SAZ
- Use EBS-optimized instance type



Getting started with Amazon RDS for MariaDB

Information
https://aws.amazon.com/rds/mariadb

Pricing
https://aws.amazon.com/rds/mariadb/pricing/

MariaDB user guide
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MariaDB.html

Getting started with Amazon RDS for MySQL

Information
https://aws.amazon.com/rds/mysql/

Pricing
https://aws.amazon.com/rds/mysql/pricing/

MySQL user guide
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html

Getting started with Amazon Aurora

Information
https://aws.amazon.com/aurora/

Pricing
https://aws.amazon.com/aurora/pricing/

Aurora user guide
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Aurora.html