



MySQL 8.4: Essentials

Student Guide
S1107033GC10

Learn more from Oracle University at education.oracle.com

O

Copyright © 2024, Oracle and/or its affiliates.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Trademark Notice

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

1009062024

Contents

1 Course Overview

- Target Audience 1-2
- Prerequisites 1-3
- Learning Outcomes 1-4
- Course Outline 1-5

2 MySQL Enterprise Edition

- Objectives 2-2
- What is MySQL? 2-4
- MySQL is Open Source 2-5
- MySQL Commercial Products 2-6
- The Oracle MySQL Business 2-7
- The Oracle MySQL Business: Licensing 2-9
- MySQL OEM Licensing 2-10
- The Oracle MySQL Business: Support 2-12
- Support: MySQL Enterprise 2-13
- The Oracle MySQL Business: Tools and Advanced Features 2-15
- MySQL Shell 2-16
- MySQL Server Process 2-17
- MySQL Enterprise Edition: Tools and Services 2-18
- MySQL Enterprise Edition: Plugins 2-19
- MySQL Enterprise Management Tools 2-21
- The Oracle MySQL Business: MySQL HeatWave in the Cloud 2-30
- MySQL HeatWave 2-31
- The Bottom Line 2-32
- Summary 2-35

3 Installation and Architecture

- Objectives 3-2
- The Perfect MySQL Server 3-4
- MySQL Downloads 3-5
- MySQL LTS and Innovation Releases 3-6
- MySQL Enterprise and Docker 3-8
- MySQL Documentation 3-9
- MySQL – Installation 3-11

| | |
|--|------|
| Package Manager for MySQL on Linux: Examples | 3-14 |
| Package Manager Configuration | 3-15 |
| MySQL by Using a Package Manager on Linux | 3-16 |
| Extend MySQL with Plugins and Components | 3-17 |
| MySQL User Accounts | 3-19 |
| MySQL Installation Deep Dive (Linux) | 3-20 |
| Create User That Runs MySQL Server Process | 3-21 |
| Create Directories Infrastructure | 3-23 |
| Initial Configuration | 3-24 |
| Initialize the Data Directory | 3-25 |
| Final actions | 3-26 |
| MySQL Upgrades and Downgrades | 3-28 |
| From Community to Enterprise | 3-29 |
| MySQL Shell Upgrade Checker | 3-30 |
| Summary | 3-31 |

4 Database Design

| | |
|--|------|
| Objectives | 4-2 |
| Storage Engine Features | 4-4 |
| MySQL InnoDB: The Default Storage Engine | 4-5 |
| MySQL InnoDB: Transactions | 4-6 |
| MySQL Database Design | 4-8 |
| MySQL Tables: CREATE TABLE Syntax | 4-9 |
| MySQL Data Types | 4-10 |
| MySQL Indexes | 4-13 |
| Indexes: Tips | 4-14 |
| Examples | 4-15 |
| Joining Tables | 4-17 |
| Examples: Joins (Outer) | 4-18 |
| EXPLAIN | 4-19 |
| MySQL Partitioning | 4-21 |
| Summary | 4-22 |

5 Database Security

| | |
|--|------|
| Objectives | 5-2 |
| Regulatory Compliance | 5-4 |
| Attack Vectors and Targets for Databases | 5-7 |
| Mitigating Database Vulnerabilities | 5-8 |
| Mitigating Database Attacks | 5-10 |
| Mitigating Database Malicious Actions | 5-11 |
| MySQL Database Hardening Best Practices | 5-12 |

| | |
|---|------|
| Hardening: Database and OS Installation, Configuration, and Maintenance | 5-13 |
| MySQL User Accounts | 5-15 |
| User Passwords | 5-16 |
| User Passwords – Examples | 5-17 |
| MySQL Enterprise Authentication | 5-19 |
| MySQL Enterprise Authentication: Native LDAP | 5-20 |
| MySQL Enterprise Authentication: PAM | 5-21 |
| MySQL Enterprise Authentication: Windows | 5-22 |
| MySQL Role-based Authentication | 5-23 |
| Example Syntax with Graph Result | 5-24 |
| MySQL Privileges | 5-26 |
| Resetting MySQL Privileges | 5-27 |
| Dynamic Privileges | 5-28 |
| Summary | 5-29 |

6 MySQL Enterprise Security Tools

| | |
|--|------|
| Objectives | 6-2 |
| MySQL Enterprise Audit | 6-4 |
| MySQL Enterprise Audit Workflow | 6-5 |
| Audit Filtering | 6-6 |
| MySQL Enterprise Firewall | 6-8 |
| MySQL Enterprise Transparent Data Encryption (TDE) | 6-11 |
| MySQL Enterprise TDE Architecture | 6-12 |
| MySQL Enterprise Masking and De-Identification | 6-13 |
| Summary | 6-15 |

7 MySQL Backup

| | |
|---|------|
| Objectives | 7-2 |
| Why back up? | 7-4 |
| Backup and Recovery Can Be Trickier Than It Appears | 7-5 |
| Backup Types: Logical | 7-6 |
| Backup Types: Physical | 7-7 |
| Backup Types | 7-8 |
| Full and Incremental Backups | 7-9 |
| Full and Differential Backups | 7-10 |
| Storage Snapshots | 7-12 |
| mysqldump | 7-13 |
| mysqldump examples | 7-14 |
| MySQL 8 Shell: Parallel Export/Import | 7-15 |
| MySQL Replication and Backups | 7-16 |
| MySQL Enterprise Backup | 7-17 |

Binary Logs 7-18
Global Transaction IDentifier (GTID) 7-19
GTID vs. Anonymous Binary Log 7-20
Point-in-time Recovery: Process 7-21
How to Save Binary Logs 7-22
MySQL Enterprise Backup 7-24
MySQL Enterprise Backup: Instance Support 7-26
MySQL Enterprise Backup: Integration with MySQL Enterprise Monitor 7-27
Backup Process Short Summary 7-28
MySQL Enterprise Backup Files 7-29
Restore Process 7-30
MySQL Enterprise Backup Examples 7-33
MySQL Enterprise Backup Options Just some of the many options that can be used 7-34
MySQL Enterprise Backup Deployment – Example 7-35
Summary 7-36

8 MySQL Replication

Objectives 8-2
MySQL Replication 8-4
MySQL Replication Use Cases 8-5
Replication Steps – Summary 8-6
MySQL Asynchronous Replication 8-8
MySQL Replication Communication 8-11
MySQL Source-Source 8-13
MySQL Chain 8-14
MySQL Replication – Ring Topology 8-15
Multi-Source Replication 8-16
Summary 8-17

9 Enabling High Availability in MySQL

Objectives 9-2
InnoDB Cluster 9-4
InnoDB Cluster: Setup 9-5
InnoDB Cluster: Application Access 9-6
InnoDB Cluster: Application Access after Failover 9-8
Hardware Configuration 9-9
InnoDB Cluster Components 9-11
MySQL Shell for Configuring InnoDB Cluster 9-12
MySQL Router 9-13
MySQL Group Replication 9-14

| | |
|--|------|
| Group Replication | 9-15 |
| Split-Brain Scenario: Working Cluster | 9-17 |
| Split-Brain Scenario: Network Split | 9-20 |
| Split-Brain Scenario: What happens if we do nothing? | 9-21 |
| Group Replication: Network Split | 9-23 |
| Group Replication: Split-Brain Prevention | 9-24 |
| Network Partitioning | 9-25 |
| Monitoring Group Replication | 9-27 |
| Group Replication Server States | 9-28 |
| InnoDB Cluster: Shared Data | 9-29 |
| Single-Primary Mode | 9-31 |
| Multi-Primary Mode | 9-32 |
| Important Notes for Multi-Primary | 9-33 |
| MySQL Replicaset | 9-35 |
| MySQL InnoDB ClusterSet Replication between InnoDB Clusters | 9-36 |
| MySQL InnoDB Read Replicas Improved Scalability and Redundancy | 9-37 |
| Kubernetes Vs. Virtual Machine (VM) | 9-38 |
| MySQL and Kubernetes | 9-39 |
| MySQL Operator for Kubernetes | 9-40 |
| Summary | 9-41 |

10 Monitoring MySQL

| | |
|--|-------|
| Objectives | 10-2 |
| MySQL DBA Challenges | 10-4 |
| Source of Database Performance Problems | 10-5 |
| MySQL DBA Checklist | 10-6 |
| Slow Query Log | 10-8 |
| Performance Schema | 10-9 |
| SYS Schema | 10-10 |
| Oracle Enterprise Manager for Monitoring MySQL | 10-11 |
| Summary | 10-12 |

11 MySQL HeatWave

| | |
|--|-------|
| Objectives | 11-2 |
| MySQL HeatWave Overview | 11-4 |
| Managed MySQL: Focus on Your Business | 11-5 |
| HeatWave Speeds Up Analytic Queries | 11-6 |
| HeatWave Is Available in Public Clouds | 11-7 |
| HeatWave Performance and Price Summary TPCH | 11-8 |
| HeatWave Single System for OLTP, Analytics, and ML | 11-9 |
| MySQL HeatWave Database Service Shapes | 11-11 |

| | |
|---|-------|
| MySQL HeatWave Shapes – Minimum Configuration | 11-12 |
| High Availability | 11-15 |
| Inbound Replication | 11-16 |
| Outbound Replication | 11-17 |
| MySQL Migration MySQL Shell Dump and Load | 11-18 |
| Migrating to MySQL HeatWave | 11-19 |
| A Short Reminder: What's MySQL? | 11-21 |
| Without HeatWave: Need Separate Systems for OLTP and OLAP | 11-22 |
| HeatWave for Real-Time Analytics | 11-23 |
| MySQL HeatWave Cluster | 11-26 |
| Machine Learning-Powered Automation for MySQL HeatWave | 11-28 |
| Auto Provisioning Machine learning prediction of memory usage to estimate cluster size | 11-29 |
| Auto Data Placement Machine learning prediction of optimal in-memory partitioning column | 11-30 |
| Auto Query Plan Improvement Optimizer learns and improves query plan based on queries executed earlier. | 11-31 |
| Machine Learning Overview | 11-33 |
| Machine Learning Overview: HeatWave AutoML | 11-34 |
| HeatWave Machine Learning | 11-35 |
| How is training in HeatWave ML fast? | 11-36 |
| HeatWave ML Explainability | 11-37 |
| Machine Learning Techniques | 11-38 |
| MySQL HeatWave Lakehouse | 11-41 |
| Summary | 11-42 |



Course Overview



Target Audience



Database
Administrators



Developers



IT Professionals



The target audiences for this course are:

- Database administrators or prospective database administrators who need to become familiar with MySQL.
- Developers working on data-driven applications.

IT professionals responsible for installing, configuring, and supporting MySQL.

Prerequisites



Basic computing infrastructure architecture and networking



Basic operating system usage and configuration

There is some prerequisite knowledge that you should know before continuing with this course.

You are expected to know the basics of computing infrastructure architecture and networking.

You will also need to have knowledge of basic operating system usage and configuration.

Learning Outcomes



Explain the core components of MySQL Enterprise Edition

Install and upgrade MySQL

Execute SQL commands

Configure backups, replication, and high availability

Set up and use MySQL HeatWave

After completing this course, you should be able to:

- Explain the core components of MySQL Enterprise Edition
- Install and upgrade MySQL
- Execute SQL commands in MySQL
- Configure backups, replication, and high availability in MySQL
- Set up and use MySQL HeatWave in Oracle Cloud Infrastructure

Course Outline



This course consists of eleven modules.

- The first module introduces this course.
- The second module introduces MySQL Enterprise Edition and how it relates to MySQL Community Edition.
- The third module covers MySQL architecture, where to find the installation software, and how to install and upgrade.
- The fourth module introduces database design, queries, and Structured Query Language (SQL).
- The fifth and sixth modules cover MySQL Security and tools for security that are available in MySQL Enterprise Edition.
- The seventh module introduces concepts involved in backing up your MySQL instances, and in particular covers the MySQL Enterprise Backup utility.
- Module 8 describes replication topologies and the details of configuring MySQL Replication.
- The ninth module explains how to configure a highly-available MySQL infrastructure.
- The tenth module shows how to monitor MySQL with the Enterprise Monitor utility.
- Finally, the eleventh module introduces MySQL HeatWave, based on MySQL Enterprise Edition but with many additional features and benefits only available in the cloud.



MySQL Enterprise Edition



Objectives



What is MySQL?

Licensing

Support

Tools and Advanced Features

MySQL HeatWave in the Cloud

What is MySQL?



What is MySQL?

- > It's a relational database.
 - Transactional, ACID
 - InnoDB storage engine
 - OLTP: low latency, high throughput
- > It enables replication.
 - Read scale-out, high availability/cluster
- > It's simple, solid, and secure.
 - Easy to use, proven at scale



MySQL is Open Source

- > MySQL is the most widely used database on the Web.
 - Relational Database Management System (RDBMS)
 - Document Store capabilities
 - Advanced security
 - Based on tablespaces
- > MySQL is licensed under the GNU General Public License (GPL).
- > MySQL is available in two editions:
 - **MySQL Community Edition** binaries and source code are available at no cost.
 - **MySQL Enterprise Edition** provides additional tools and services.



MySQL Commercial Products

| | MySQL Standard Edition | MySQL Enterprise Edition | MySQL Cluster CGE |
|-------------------------------------|------------------------|---|-----------------------|
| Oracle Premier Support | ✓ | ✓ | ✓ |
| MySQL Database Server | ✓ | ✓ | ✓ |
| MySQL Connectors | ✓ | ✓ | ✓ |
| MySQL Replication | ✓ | ✓ | ✓ |
| MySQL Workbench | ✓ | ✓ | ✓ |
| Storage Engine: MyISAM, InnoDB | ✓ | ✓ | ✓ |
| MySQL Router | X | ✓ Available in Community Edition but fully supported in Enterprise Edition | ✓ |
| MySQL Partitioning | X | ✓ | ✓ |
| MySQL Document Store | X | ✓ | ✓ |
| InnoDB Cluster | X | ✓ | ✓ |
| Oracle Enterprise Manager for MySQL | X | ✓ | ✓ |
| MySQL Enterprise Monitor | X | ✓ | ✓ |
| MySQL Enterprise Backup | X | ✓ | ✓ |
| MySQL Enterprise Security | X | ✓ | ✓ |
| MySQL Enterprise Scalability | X | ✓ | ✓ |
| MySQL Enterprise High-Availability | X | ✓ | ✓ |
| Storage Engine: NDB | X | X | ✓ |
| MySQL Cluster Manager | X | X | ✓ Cluster CGE only |

The Oracle MySQL Business



Licensing



Support



Tools /
Advanced Features



Cloud

The Oracle MySQL Business: Licencing

MySQL is a relational database management system (RDBMS) used for structured data storage and retrieval.

It is open-source software developed by Oracle Corporation.

MySQL is widely used in web applications and is the most popular database for them.

MySQL is a relational database management system (RDBMS) used for structured data storage and retrieval.

It is open-source software developed by Oracle Corporation.

MySQL is widely used in web applications and is the most popular database for them.



The Oracle MySQL Business: Licensing



Licensing



Support



Tools /
Advanced Features



Cloud

MySQL OEM Licensing

- > ISV / OEM – Independent Software Vendor / Original Equipment Manufacturer
 - Embedded Database – Integrated Database and part of your product installation
- > ISVs, OEMs, VARs, and other distributors:
 - Combine and distribute commercially licensed software with MySQL
 - Do not wish to distribute the source code for the commercially licensed software under the GPL v. 2
- > ISVs, OEMs, and VARs must enter into a commercial license agreement with Oracle

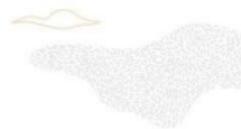
Note: This rule applies if you embed the whole MySQL or just components (such as connectors).

<http://www.mysql.com/about/legal/licensing/oem/>

The Oracle MySQL Business: Support



The Oracle MySQL Business: Support



Licensing



Support



Tools /
Advanced Features



Cloud

Support: MySQL Enterprise

- > <https://support.oracle.com>
- > Backed by the MySQL developers
- > Global scale and reach, as well as support in 29 languages
- > Hot fixes and maintenance releases
- > 24x7x365
- > Unlimited incidents
- > Consultative support



The Oracle MySQL Business: Tools and Advanced Features – Part 1



The Oracle MySQL Business: Tools and Advanced Features



Licensing



Support



Tools /
Advanced Features



Cloud

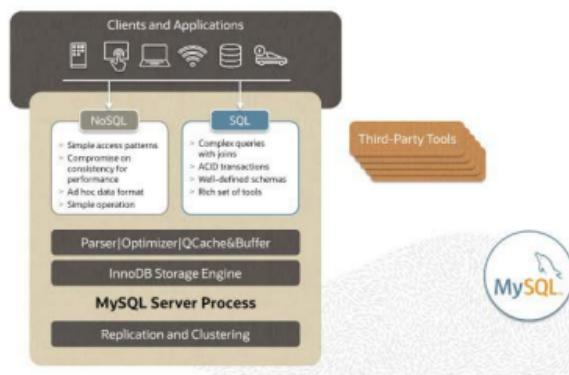
MySQL Shell

A single unified client for all administrative and operations tasks

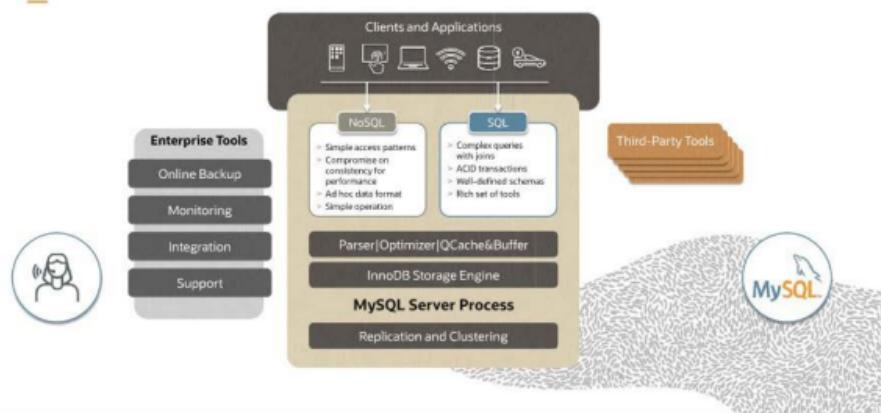
- > Multi-Language: JavaScript, Python, and SQL
 - Naturally scriptable
- > Supports both Document and Relational models
- > Exposes full Development and Admin API
- > Perform DBA operations



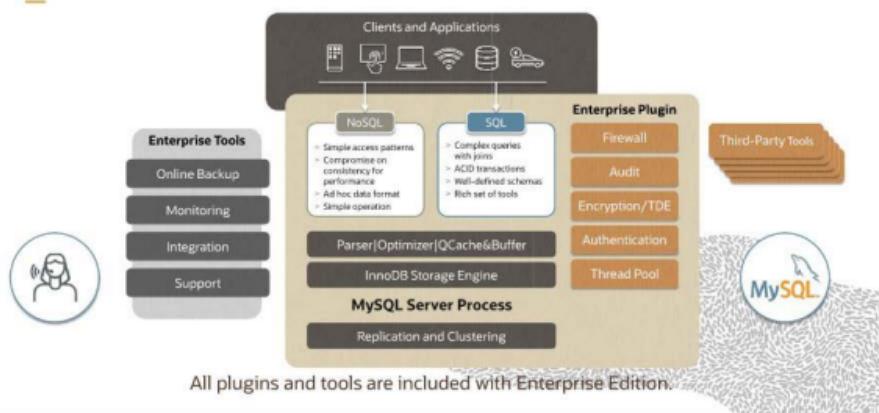
MySQL Server Process



MySQL Enterprise Edition: Tools and Services



MySQL Enterprise Edition: Plugins



The Oracle MySQL Business: Tools and Advanced Features – Part 2



MySQL Enterprise Management Tools

- > MySQL Enterprise Backup provides:
 - Online, non-locking backup, and recovery with best performances
 - Advanced features (incremental/differential backups, compression, encryption ...)
 - Integration with MMS (media management software)

MySQL Enterprise Management Tools

- > MySQL Workbench Enterprise provides:
 - Database development, design, and documentation
 - Database migrations
 - Database administration

MySQL Enterprise Management Tools

> MySQL Enterprise Monitor

- Provides real-time MySQL performance and availability monitoring
- Visually finds and fixes problem queries
- Provides MySQL best-practice advisors
- Is proactive and provides forecasting

MySQL Enterprise Management Tools

- > Oracle Enterprise Manager Plugin for MySQL provides:
 - Availability and performance monitoring
 - Configuration monitoring
 - Collection of all available metrics

MySQL Enterprise Management Tools

- > MySQL Enterprise **Thread Pool**
 - Improves scalability as concurrent connections grow
- > MySQL Enterprise **Authentication**
 - Integrates MySQL with existing security infrastructures
 - Linux PAM, LDAP, Windows
 - Kerberos
 - FIDO (passwordless authentication)

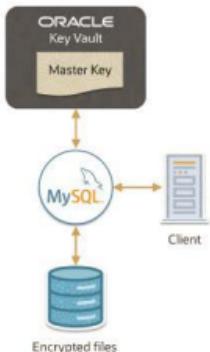
MySQL Enterprise Management Tools

- > MySQL Enterprise **Audit**
 - Out-of-the-box logging of connections, logins, and query (XML/JSON)
 - Simple- to fine-grained policies for filtering and log rotation
- > MySQL Enterprise **Firewall**
 - Out-of-policy database transactions detected and blocked



MySQL Enterprise Management Tools

- > MySQL Enterprise
Asymmetric Encryption
 - MySQL encryption libraries
 - Key management, sign and verify data
- > MySQL Enterprise
Transparent Data Encryption
 - Data-at-rest encryption in the database
 - Master Key stored outside the database (KMIP 1.1 Compliant Key Vault)



MySQL Enterprise Management Tools

> MySQL Enterprise **Masking**

- Data Masking: String Masking, Dictionary Replacement
- Random Data Generators: Range based, Payment Card, Email, SSN

Employee Table

| ID | Last | First | SSN |
|------|-----------|---------|-------------|
| 1111 | Smith | John | 555-12-5555 |
| 1112 | Templeton | Richard | 444-12-4444 |

Masked View

| ID | Last | First | SSN |
|------|-----------|---------|-------------|
| 2874 | Smith | John | XXX-XX-5555 |
| 3281 | Templeton | Richard | XXX-XX-4444 |

MySQL HeatWave in the Cloud



The Oracle MySQL Business: MySQL HeatWave in the Cloud



Licensing



Support



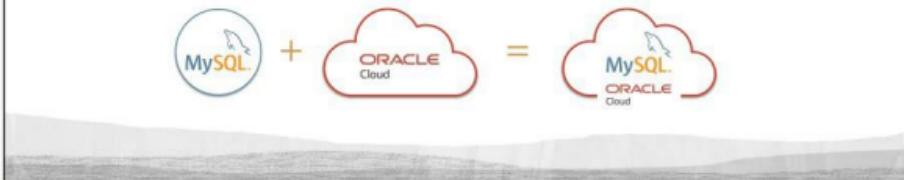
Tools /
Advanced Features



Cloud

MySQL HeatWave

- > Fully managed MySQL Service
 - Deploy, Backup and Restore, High Availability, Resize, Read Replicas, etc.
- > Powerful union of **Oracle Cloud Infrastructure** and **MySQL Enterprise Edition 8**
- > Always up to date with the latest security fixes
- > Both OLTP and analytics/ML use cases



The Bottom Line

Performance

- > Performance at scale
- > Monitoring and tuning



Peace of Mind

- > Insurance
- > Immediate help if/when needed



The Bottom Line

TCO

- > Reduced risk of down time
- > Improved productivity

Risks

- > Security and regulatory compliance
- > Contact with MySQL team



The Bottom Line

DevOps Agility

- > Automated scaling and management
- > Flexible, real-time backups



Customer Satisfaction

- > Application performance
- > Application uptime



Summary



What is MySQL?

Licensing

Support

Tools and Advanced Features

MySQL HeatWave in the Cloud



Installation and Architecture



Objectives



Preparing to Install MySQL

Installing MySQL

Initial User Accounts

Initial Configuration

Upgrading MySQL

Preparing to Install MySQL



The Perfect MySQL Server

- Some practical tips:
- Choice of operating system:
 - <https://www.mysql.com/support/supportedplatforms/database.html>
- Choice of host:
 - MySQL is supported on bare metal, VMs, containers, and in the Cloud.
- MySQL 8.0 scales very well on multiple cores.
 - MySQL 8.0 has the benefit of multi-threading and NUMA.
- InnoDB uses the available memory effectively by caching data and keys.
 - It assigns enough memory for the active data set + keys.
- Fast disks are preferred (SSD, FusionIO, etc.).
- A single point of failure should be excluded.



Multiple core performance: additional content at:

<http://dimitrik.free.fr/blog/posts/mysql-performance-over-18m-qps-with-80-ga-on-2s-skylake.html>

MySQL Downloads

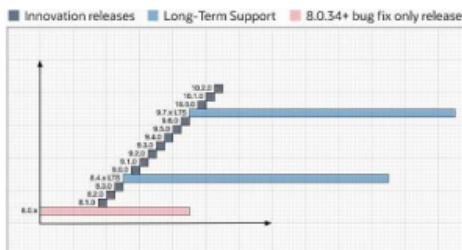
- > MySQL Community Edition:
 - <https://dev.mysql.com/downloads/>
- > MySQL Enterprise Edition:
 - <https://support.oracle.com>
- > MySQL Trial:
 - <https://edelivery.oracle.com>
 - The only most recent release of each MySQL product
- > MySQL source code is available at:
 - <https://dev.mysql.com/downloads/>
 - <https://github.com/mysql>



MySQL LTS and Innovation Releases

MySQL Long-Term Support (LTS) Releases

- Stable: bug fix and security patches only
- Backward compatibility
- Released every 2 years
- Support life cycle:
5 years premier + 3 years extended



MySQL Innovation Releases

- Leading-edge innovations
- Easy migration between LTS and Innovation
- Released every quarter
- Support life cycle: short term

MySQL LTS and Innovation Releases

- > LTS and Innovation releases are available for Community, Enterprise, and MySQL HeatWave Database Service.
- > LTS and Innovation releases are Generally Available and are recommended to be used in production environments.
- > LTS and Innovation releases are both commercially supported.
- > MySQL tools follow the same approach.
- > Connectors use the latest version:
 - They are considered Generally Available.
 - They remain compatible with all supported MySQL Server versions.



Details for Innovation and LTS Releases can be found here:

- <https://blogs.oracle.com/mysql/post/introducing-mysql-innovation-and-longterm-support-lts-versions>

MySQL Enterprise and Docker

- > MySQL is supported inside virtualization or containers:
 - KB 1383964.1
 - <https://dev.mysql.com/doc/refman/8.0/en/faqs-virtualization.html>

- > MySQL can be integrated in docker images:

- MySQL Community is available on Docker Hub.

- > MySQL Enterprise docker images are available on:

- Docker Store
<https://store.docker.com/>
 - My Oracle Support
<https://support.oracle.com>
 - Oracle Container Registry
<https://container-registry.oracle.com>



MySQL Documentation

- > Manuals (community and commercial):
 - <https://dev.mysql.com/doc/>
- > Knowledge Base (partners and customers only):
 - <https://support.oracle.com>
- > Blogs from MySQL engineers:
 - <https://blogs.oracle.com/mysql/>
 - Blogs moved to <https://dev.mysql.com/blog-archive/>
 - <https://insidemysql.com>
 - <https://mysqlserverteam.com/>
 - <http://mysqlhighavailability.com/>
- > The MySQL Community evangelist:
 - <https://lefred.be/>
- > Performance expert:
 - <http://dimitrik.free.fr>

Installing MySQL



MySQL – Installation

- > Many operating systems are supported and documented, such as:
 - Unix/Linux
 - Microsoft Windows
 - OS X
 - Unbreakable Linux Network (ULN)
 - Solaris and OpenSolaris
 - FreeBSD 9
- > Documentation describes installation, post installation setup and test, upgrade, and downgrade.

<https://dev.mysql.com/doc/refman/8.0/en/installing.html>

MySQL – Installation

- > Windows:
- > MySQL Installer

- It includes most MySQL Packages for Windows in one single package.
- It is used for MySQL Database, MySQL Connectors, MySQL Workbench, mysqlslap (Load generation tool), and sample models.
- It installs documentation and sample databases.
- Initial setup is performed by an installation wizard.

- > Binary zip ("Noinstall Archive")
 - For binaries only
 - Is configured manually

MySQL – Installation

- > Linux
- > Installation is done with repositories, packages (yum, rpm, deb), or binaries (tarball and zip).
 - The choice is made based on:
 - Knowledge
 - Company standards
 - Flexibility
- > Some Linux distributions provide third-party forks of MySQL.
 - It's easy to reinstall MySQL over forks.
- > MySQL Linux dependencies include:
 - libaio
 - /lib64/libtinfo.so.5 (usually contained in ncurses-compat-libs)

Package Manager for MySQL on Linux: Examples

Package managers simplify MySQL installation by installing package dependencies.

- > On RPM-based systems (Oracle Linux, Red Hat, ...), use `yum install` to install Community Edition.

Examples:

- `yum install mysql-community-server`
- `yum install mysql-workbench`

- > On APT-based systems (Ubuntu, Debian, ...), use `apt install` to install Community Edition.

Examples:

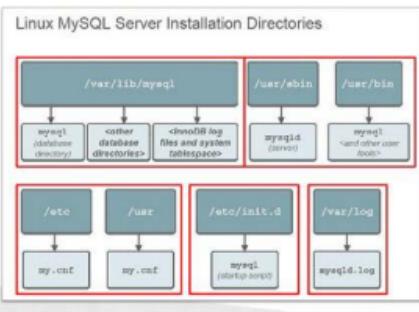
- `apt install mysql-community-server`
- `apt install mysql-workbench`

Package Manager Configuration

- > Package managers have their own default settings:
 - Example: The location of configuration files such as `/etc/my.cnf` and `/etc/mysql/my.cnf`
- > Enterprise Edition has no public repository, so download the appropriate .rpm or .deb package from the <https://support.oracle.com> website.
 - There is a dedicated package (MySQL Database/Components Yum Repository TAR) to set up a self-hosted Yum repository.

MySQL by Using a Package Manager on Linux

Example from Oracle Linux



Extend MySQL with Plugins and Components

- > Plugins and components dynamically extend MySQL functionality.
 - Plugins implement APIs and may call plugin service APIs exposed by the server.
 - They also have access to all the server binary global symbols.
 - Components are self-contained code containers that interact by implementing and consuming services.
- > MySQL can also be extended with:
 - Stored procedures and functions
 - User-defined functions (UDF)



<https://dev.mysql.com/doc/refman/8.0/en/function-loading.html>

Initial User Accounts

MySQL User Accounts

- > Within MySQL, accounts comprise three elements:
 - username@hostname + password
 - "hostname" identifies the host from where the mysql client connects.
 - A hostname value of '%' indicates an account that can connect from any IP address.
- > Initial installation creates only the administrative account 'root'@'localhost'.
 - The initial root password is displayed on screen during installation or added to the error log file.
 - You must change the password before executing any MySQL command.

<https://dev.mysql.com/doc/refman/8.0/en/user-account-management.html>

MySQL Installation Deep Dive (Linux)

- The basic steps are the same for all platforms.
- We use the generic binaries distribution here.
- MySQL installation steps include:
 - Create a user that runs MySQL server process
 - Create the directories infrastructure
 - First setup of configuration file (my.cnf)
 - Install and initialize the instance
 - Start mysql server
 - Final actions:
 - Create start/stop script
 - Tune the configuration file

<https://dev.mysql.com/doc/refman/8.0/en/installing.html>

Create User That Runs MySQL Server Process

- > Generic system user
- > Does not require system privileges
- > Does not require shell
- > Might require permissions for startup and backup/recovery
 - For example, if you use sudo at the shell to perform backups

Initial Configuration



Create Directories Infrastructure

- There are only two mandatory variables: 'basedir' and 'datadir'.
- Structure can be complex, for example:

| Folder | Usage |
|---------------------|---|
| [...]/mysql | Root folder Sometimes inside /usr/local/ or /opt/ |
| [...]/mysql/mysql-X | Directory with binaries (<code>basedir</code>) Sometime referred to with a symbolic link to simplify management |
| [...]/mysql/data | Directory with database files (<code>datadir</code>) In some Linux distributions, the default is /var/lib/mysql/ |
| [...]/mysql/logs | Directory where the store logs In some Linux distributions, the default is /var/log/mysql/ |
| [...]/mysql/temp | Directory for temporary files |
| [...]/mysql/uploads | Directory where the server can read files to upload (LOAD DATA) |
| [...]/mysql/etc | Directory with configuration file(s) (my.cnf) In some Linux distributions, the default is /etc/mysql |
| [...]/mysql/binlogs | Directory with binary logs used for replication and backups |

Initial Configuration

- > my.cnf
 - my.ini on Windows
- > It's a text file split into sections.
- > It's read at startup.
 - Errors prevent server startup.
- > What is a good my.cnf?
 - Don't tune too much!
 - Windows has defaults.
 - Guidelines are available in documentation and whitepapers.

Socket connection
[mysqld]
General
port=3306
socket=/tmp/mysql.sock

File locations
basedir=/usr/local/mysql
datadir=/var/lib/mysql
tmpdir=/tmp
log_error=/var/log/mysqld_error.log
log=/var/log/mysqld.log
pid_file=/var/run/mysqld/mysqld.pid
log_error_use_symlink=1
log_error_use_symlink=1

Maximum limits
max_connections=100

InnoDB
innodb_buffer_pool_size=8M
innodb_flush_method=O_DIRECT

MySQL
log_buffer_size=128K
#Other settings
innodb_log_group_home_dir=/var/lib/mysql/innodb_log_group_home_dir

Threadpool settings
pluggable-threadpool_size=0

Initialize the Data Directory

- > Package initialization creates the data dictionary, redo logs, undo logs, and required files.
- > If you install MySQL from binary archive, you must initialize the data directory.
 1. Change the location to the top-level directory of your MySQL installation.
 2. Grant directory user and group ownership to your "mysql" user and "mysql" group.
 3. Initialize the data directory, e.g.:
 - `$> bin/mysqlld --initialize --user=mysql`
- > (Optional) With `--initialize-insecure`, no root password is generated.
 - This is insecure; it is assumed that you intend to **assign a password** to the account in a timely fashion **before putting the server into production use**.

<https://dev.mysql.com/doc/refman/8.0/en/data-directory-initialization.html>

Final actions

- > Choose how to start the mysqld service:
 - Automate startup/shutdown/respawn
 - mysqld or mysqld_safe, systemd, Windows service ...
 - `--mysqld --help` and `mysqld --help --verbose` [check for option order]
 - Special-purpose startup options:
 - `--defaults-file=/mysql/my.cnf`
 - `--bind-address`
 - `--skip-grant-tables`
- > Refine my.cnf/my.ini, for example, to:
 - Disable binary logs to reduce database creation time
 - To use a default configuration only at creation time
 - To provide particular settings for your application

<https://dev.mysql.com/doc/refman/8.0/en/automatic-start.html>

Upgrading MySQL



MySQL Upgrades and Downgrades

- > Upgrades and Downgrades are offline operations.
- > Best practice: Back up your MySQL environment.
 - Triggers, stored procedures, configs, binaries
- > Options include:
 - In-Place Upgrade: Involves shutting down the old MySQL version, replacing the old MySQL binaries or packages with the new ones, and restarting MySQL in the existing data directory
 - Logical Upgrade: Involves exporting existing data from the old MySQL version using mysqldump/MySQL Shell, installing the new MySQL version, and loading the dump file into the new MySQL version.
- > Don't forget MySQL Connectors and libraries.
 - You may need to recompile any C or C++ programs that link statically to the ".lib" or ".a" forms of MySQL libraries.
- > Replication may help to reduce application down time.

From Community to Enterprise

To change from MySQL Community Edition to MySQL Enterprise Edition, replace the Community Edition binaries with Enterprise Edition binaries of the same version.

1. Download the Enterprise software, exactly the same version as Community.
2. Stop the Community binaries.
3. Remove the Community binaries.
4. Install the Enterprise binaries.
5. Start the instance using the same configurations files, datafiles, etc. (no changes).

MySQL Shell Upgrade Checker

- > Run MySQL Shell before upgrade:

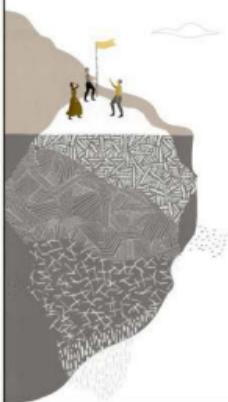
```
MySQL [JS] > util.checkForServerUpgrade("user@127.0.0.1:3306",
  {"targetVersion": "8.4.9", "configPath": "/etc/my.cnf"})
```

- > MySQL Shell detects:

- Reserved keywords
- Removed datatypes
- Changed parameter default values

- > Upgrade checker will not identify problems such as GIS changes or deprecated MySQL functions used by your application.
- > The user account requires RELOAD, PROCESS, and SELECT privileges.

Summary



Preparing to Install MySQL

Installing MySQL

Initial User Accounts

Initial Configuration

Upgrading MySQL



Database Design



Objectives



Data Storage

Databases and Tables

Indexes

Joins

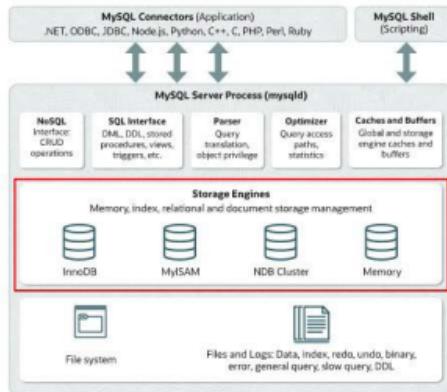
Partitioning

Data Storage



Storage Engine Features

- > Concurrency
- > Transaction Support
- > Referential Integrity
- > Physical Storage
- > Index Support
- > Memory Caches
- > Performance Aids



MySQL InnoDB: The Default Storage Engine

- > Balances high reliability and high performance.
- > ACID compliant and **fully transactional** with ROLLBACK/COMMIT/Foreign Keys
- > Supports encryption and compression
- > Built-in HA clustering with Group Replication
- > Supports large and complex data sets:
 - > Maximum tablespace size is 64TB
 - > Serves more than 1000 columns per table
 - > Contains up to 64 **secondary indexes**
 - > Supports row size up to 64Kb (less BLOBs)

MySQL InnoDB: Transactions

- > By default, autocommit=ON.
 - > The statements START TRANSACTION and BEGIN are synonyms.
 - > A "start transaction" statement temporarily disables autocommit (for DML).
- Example:

```
START TRANSACTION; {begin}  
INSERT INTO t1 ('name') VALUES ('Jonas');  
INSERT INTO t1 ('name') VALUES ('Lisa');  
ROLLBACK;  
INSERT INTO t1 ('name') VALUES ('Beth');  
SELECT * FROM t1;
```

Databases and Tables

MySQL Database Design

- > In MySQL, "Schema" and "Database" are synonyms.
- > On the file system:
 - A database is an OS directory.
 - Tables are files or portions of files.
- > There are some internally created schemas.
 - Data dictionary: `mysql`, `information_schema`
 - Performances: `performance_schema`, `sys`
 - InnoDB Cluster metadata: `mysql_innodb_cluster_metadata`

MySQL Tables: CREATE TABLE Syntax

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...)
[table_options] [partition_options]
```

Example:

```
CREATE TABLE `ppt_test`.`ppt_table_test`
(`id` INT NOT NULL AUTO_INCREMENT,
`lastname` VARCHAR(45) NULL,
`forename` VARCHAR(45) NULL,
`birthday` DATE NULL,
PRIMARY KEY (`id`));
```

MySQL Data Types

- > Numeric Exact Value
 - INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT
 - DECIMAL, NUMERIC
- > Numeric Approximate Value
 - FLOAT, DOUBLE
- > Bit-Value Type
 - BIT
- > Date and Time Types
 - DATE, DATETIME, TIMESTAMP, YEAR
- > String Types
 - CHAR, VARCHAR
 - BINARY, VARBINARY
 - ENUM
 - SET

| Type | Storage (Bytes) | Signed Range | Unsigned Range |
|-----------|-----------------|--|-------------------------|
| TINYINT | 1 | -128 to 127 | 0 to 255 |
| SMALLINT | 2 | -32768 to 32767 | 0 to 65535 |
| MEDIUMINT | 3 | -8388608 to 8388607 | 0 to 16777215 |
| INT | 4 | -2147485488 to 2147485487 | 0 to 4294967295 |
| BIGINT | 8 | -2 ⁶³ to 2 ⁶³ -1 | 0 to 2 ⁶⁴ -1 |

| Value | CHAR(4) | Storage Required | VARCHAR(4) | Storage Required |
|-----------|---------|------------------|------------|------------------|
| " | ' ' | 4 bytes | " | 1 byte |
| 'ab' | 'ab' | 4 bytes | 'ab' | 3 bytes |
| 'abcd' | 'abcd' | 4 bytes | 'abcd' | 5 bytes |
| 'abcdefg' | 'abcd' | 4 bytes | 'abcd' | 5 bytes |

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Each data type has different ranges, defaults, and storage requirements

MySQL also provides dedicated functions.

MySQL Data Types

- > BLOB
 - TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB
- > TEXT
 - TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
- > Spatial
 - GEOMETRY, POINT, LINESTRING, POLYGON
 - MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION
- > JSON Data Type

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Each data type has different ranges, defaults, and storage requirements

MySQL also provides dedicated functions.

Indexes



MySQL Indexes

- > Optimize queries and data manipulation operations
- > Store data in a sorted order
 - Non-Unique - May refer to many records
 - Unique or Primary - Must refer to only one record
- > Are available in multiple types
 - BTREE
 - HASH
 - RTREE (spatial only)

<https://dev.mysql.com/doc/refman/8.0/en/innodb-indexes.html>

Indexes: Tips

- > Create a Primary Key (PK) on each table.
- > PK is replicated in each secondary index.
 - Keep the Primary Key simple.
- > Too many indexes can slow down inserts/deletes.
 - Use only the indexes you must have.
 - Check often if they are used or not (use the sys report or MEM).
- > Reduce indexes using prefix and compound keys.

<https://dev.mysql.com/doc/refman/8.0/en/optimization-indexes.html>
<https://lefred.be/content/about-mysql-and-indexes/>

Examples

```
CREATE TABLE address (
    id INT unsigned NOT NULL PRIMARY KEY,
    email VARCHAR(100),
    lname CHAR(30),
    fname CHAR(30),
    INDEX name (lname, fname)
);
```

Example queries that can use indexes:

```
> select * from address where lname='smith';
> select * from address where lname='smith' and fname='john';
> select * from address where lname='smith' and id > 100;
> select fname from address where lname='smith';
```

Joins



Joining Tables

Join (Inner Join)

- Returns rows from either table to appear in the result if and only if both tables meet the conditions specified in the ON clause

Left Join (Outer Join)

- Returns all records from the left table and the matched records from the right table

Right Join (Outer Join)

- Returns all records from the right table and the matched records from the left table

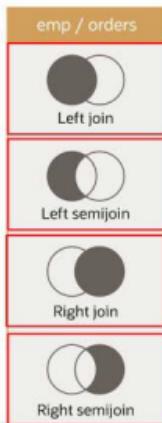
| Diagram | SQL Statement | Description |
|---------|--|---|
| | <pre>SELECT EmployeeID, LastName FROM Employees INNER JOIN Orders ON A.EmployeeID = B.EmployeeID</pre> | Employees who have registered orders (All employees who have registered at least one order) |
| | <pre>SELECT EmployeeID, LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Employees RIGHT JOIN Orders ON EmployeeID = Order.EmployeeID GROUP BY Employee.LastName</pre> | All employees and orders they have registered (There are employees who have not registered any orders) |
| | <pre>SELECT EmployeeID, LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Employees LEFT JOIN Orders ON Employee.EmployeeID = Order.EmployeeID GROUP BY Employee.LastName</pre> | All orders and employees who registered them (There are orders that have not been registered by any employee) |

For more information on joins, see the documentation at

<https://dev.mysql.com/doc/refman/8.0/en/join.html>

Examples: Joins (Outer)

- > All employees and their registered orders:
 - `SELECT * FROM emp LEFT JOIN orders ON emp.id=orders.emp_id;`
- > Employees who have not registered any orders:
 - `SELECT * FROM emp LEFT JOIN orders ON emp.id=orders.emp_id WHERE orders.emp_id IS NULL;`
- > All orders and employees who have registered them:
 - `SELECT * FROM emp RIGHT JOIN orders ON emp.id=orders.emp_id;`
- > All orders that do not have an employee:
 - `SELECT * FROM emp RIGHT JOIN orders ON emp.id=orders.emp_id WHERE emp.id IS NULL;`



EXPLAIN

- > EXPLAIN shows how MySQL will process a statement
 - Includes information about which indexes will be used
 - Includes information about how tables are joined and in which order.
 - Works with SELECT, DELETE, INSERT, REPLACE, and UPDATE statements.
 - Example:

```
EXPLAIN SELECT * FROM emp
LEFT JOIN orders
ON emp.id=orders.emp_id;
```

References

- <https://dev.mysql.com/doc/refman/8.0/en/explain.html>
- <https://dev.mysql.com/doc/refman/8.0/en/explain-output.html>

Partitioning



MySQL Partitioning

- > The partition plugin enables the distribution of portions of individual tables across a file system according to rules.
 - Ranges, lists, hash, key
- > MySQL allows distribution of partitions across an instance/physical storage.
- > It explains return info on partition usage in a query.
 - `explain select * from world_innodb.City_part;`
- > Support for partitioning and HA is not part of the MySQL Standard version (only Enterprise Edition).



For more information on partitioning, see the documentation at

<https://dev.mysql.com/doc/refman/8.0/en/partitioning.html>

Summary



Data Storage

Databases and Tables

Indexes

Joins

Partitioning



Database Security



Objectives



Regulatory Compliance

Mitigating Risks

Authentication

Authentication in MySQL Enterprise Edition

Authorization

Regulatory Compliance

Regulatory Compliance

Regulations include:

- GDPR, General Data Protection Regulation: Protection of Personal Data
- PCI – DSS: Payment Card Data
- HIPAA: Privacy of Health Data
- Sarbanes Oxley, GLBA, The USA Patriot Act:
- Financial Data, NPI "personally identifiable financial information"
- FERPA – Student Data
- Data Protection Act (UK): Protection of Personal Data



Sarbanes-Oxley



Data Protection Act 1998

Regulatory Compliance



► Requirements include:

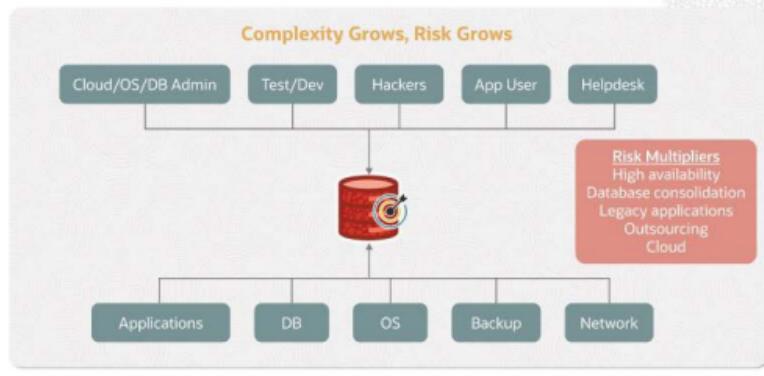
- Continuous Monitoring (Users, Schema, Backups, etc.)
- Data Protection (Encryption, Privilege Management, etc.)
- Data Retention (Backups, User Activity, etc.)
- Data Auditing (User Activity, etc.)

Mitigating Risks

Identify potential risks and develop strategies to prevent or reduce their impact.



Attack Vectors and Targets for Databases



Mitigating Database Vulnerabilities

- Poor Configurations
 - Set controls and change default setting
- Over-Privileged Accounts
 - Use privilege policies
- Weak Access Control
 - Use dedicated administrative accounts
- Weak Authentication
 - Use strong password enforcement
- Weak Auditing
 - Enforce compliance and audit policies



Mitigating Database Vulnerabilities

- Lack of Encryption
 - Use data, backup, and network encryption
- Proper Credential and Key Management
 - Protect passwords, use key vaults
- Unsecured Backups
 - Encrypt backups
- No Monitoring
 - Implement security monitoring for users and objects
- Poorly Coded Applications
 - Use database firewall



Mitigating Database Attacks

- SQL Injection
 - DB firewall (allow list), input validation
- Buffer Overflow
 - Frequently apply database software updates, DB firewall (allow list), input validation
- Insider Abuse
 - Tight access controls, user-specific authentication (no general accounts), auditing, monitoring, encryption
- Brute Force Attack
 - Lock out accounts after a defined number of incorrect attempts
- Network Eavesdropping
 - Require SSL/TLS for all connections and transport
- Malware
 - Tight access controls, limited network IP access, change default settings, encryption



Mitigating Database Malicious Actions

- Information Disclosure: Obtain credit card and other personal information
 - Encryption – Data and network, implement tighter access controls
- Denial of Service: Run resource-intensive queries
 - Resource usage limits – Set various limits, e.g., maximum connections, sessions, timeouts, etc.
- Elevation of Privilege: Retrieve and use administrator credentials
 - Stronger authentication, access controls, auditing
- Spoofing: Retrieve and use other credentials
 - Stronger account and password policies
- Tampering: Change data in the database and delete transaction records
 - Tighter access controls, auditing, monitoring, backups



MySQL Database Hardening Best Practices



| | | |
|---|---|--|
| Installation <ul style="list-style-type: none">➢ Keep MySQL up to date➢ MySQL Installer for Windows➢ Yum/Apt/other repositories | Authentication <ul style="list-style-type: none">➢ Password policies➢ Multifactor authentication➢ External authentication➢ X.509 | Authorization <ul style="list-style-type: none">➢ Remove extra accounts➢ Grant minimal privileges➢ Audit users and privileges |
| Configuration <ul style="list-style-type: none">➢ Firewall➢ Auditing and logging➢ Limit network access➢ Monitor changes | Encryption <ul style="list-style-type: none">➢ SSL/TLS for secure connections➢ Data encryption (AES, RSA)➢ TDE➢ Masking and de-identification | Collateral <ul style="list-style-type: none">➢ Encrypt backups➢ Secure replications➢ Secure HA and DR |

Hardening: Database and OS Installation, Configuration, and Maintenance

- › Build on a secure platform.
 - OS patched, firewall, etc.
- › Use the Secure build.
 - <https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/>
- › Keep the **operating system** and MySQL security patches up to date.
- › Follow OS vendor-specific hardening guidelines.
- › Follow Center for Internet Security (CIS) Benchmarks.

When you patch your operating system or MySQL, your system might require a restart.

For more information on hardening an Oracle Linux Server, see

<http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-harden-oracle-linux-1695888.html>

Authentication



MySQL User Accounts

- Within MySQL, accounts are composed of three elements:
 - “**username@host**” identifier, and a **password**
- The account identifier comprises two distinct values: username and host.
 - “host” is the network ID of the machine from where the mysql client connects.
 - A host value of ‘%’ indicates that the client can connect from any IP address.
 - An identifier with no host value indicates that the client can connect from any host.
 - Examples: ‘user_name’@‘myhost.oracle.com’, ‘db_user’@‘192.168.1.1’
- Example:
 - `CREATE USER 'test'@'%' IDENTIFIED 'VeryComplex1!';`

<https://dev.mysql.com/doc/refman/8.0/en/user-account-management.html>

User Passwords



- Passwords are stored in an encrypted form
 - Using a secure one-way hashing algorithm implemented in an authentication plugin
- The default authentication plugin is `caching_sha2_password`.
- To change the authentication plugin:
 - Add in the `my.cnf` configuration file, under [`mysqld`] section:

```
default_authentication_plugin=mysql_native_password
```
 - Specify at the command line when you invoke the server:

```
--default_authentication_plugin=caching_sha256_password
```

User Passwords – Examples

```
> mysql> CREATE USER 'jeffrey'@'localhost'  
        IDENTIFIED BY 'password_value';  
  
> mysql> SET PASSWORD FOR jeffrey'@'localhost' = 'password_value';  
  
> mysql> SET PASSWORD = 'mypass';  
  
> mysql> CREATE fred@localhost  
        IDENTIFIED WITH mysql_native_password  
        BY 'HisPa55wOrd!';
```

Authentication in MySQL Enterprise Edition



MySQL Enterprise Authentication

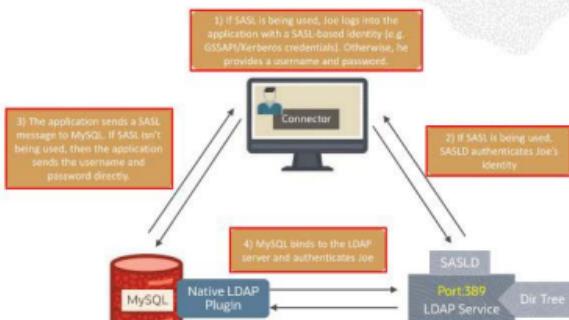
- Integrates with Centralized Authentication Infrastructure
 - Centralized Account Management
 - Password Policy Management
 - Groups and Roles
- Pluggable Authentication Modules (PAM)
 - Linux PAM Standard interface (Unix, LDAP, Kerberos, others)
- Kerberos
- FIDO interface (passwordless authentication)
- Single Sign-On (SSO)
 - Plugin is available to access native LDAP service for authentication.
 - Plugin is available to access native Windows service.



Integrates MySQL with existing security infrastructures

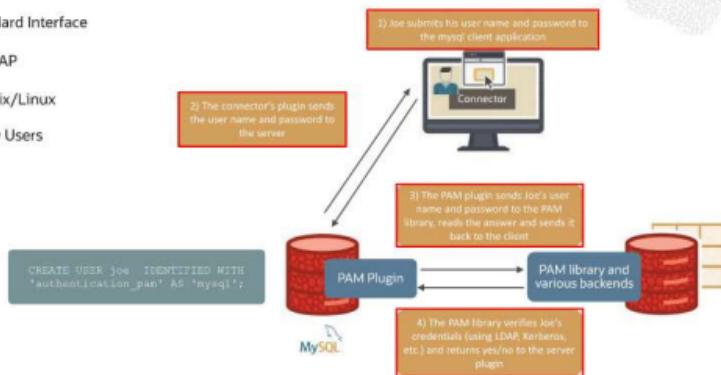
MySQL Enterprise Authentication: Native LDAP

- Direct Connection over LDAP Protocol/Ports
- Authentication with:
 - User and password or
 - SASL
- Customizable for users and groups



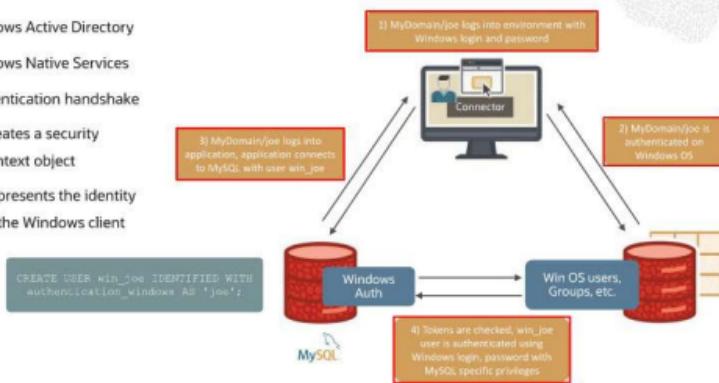
MySQL Enterprise Authentication: PAM

- Standard Interface
 - LDAP
 - Unix/Linux
- Proxy Users



MySQL Enterprise Authentication: Windows

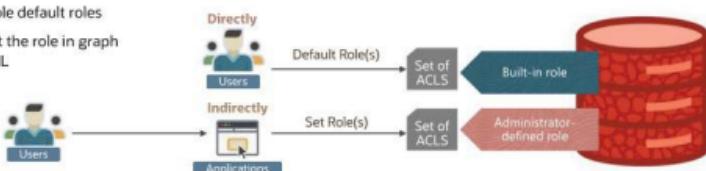
- Windows Active Directory
- Windows Native Services
- Authentication handshake
 - Creates a security context object
 - Represents the identity of the Windows client



MySQL Role-based Authentication



- Enable group-based management of user and applications rights
- Has multiple default roles
- Can export the role in graph in GraphML



Note: A role is both a *principal* (can be granted permissions, like a user) and a *securable* (to which principals can be granted permissions, like a table)



Example Syntax with Graph Result

```
CREATE ROLE 'r1', 'admin-db1', 'admin-db2', 'admin-dbtl1', 'admin-db2tl1', 'app-updater';

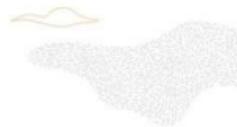
CREATE USER 'app-middleware-db1'@localhost,
    'app-middleware-db2'@localhost, 'app'@localhost;

GRANT 'admin-db1', 'app-updater' TO 'app-middleware-db1'@localhost;
GRANT 'r1' TO 'app-middleware-db1'@localhost WITH ADMIN OPTION;
GRANT 'admin-db2tl1' TO 'admin-db1';
GRANT 'admin-dbtl1' TO 'admin-db1' WITH ADMIN OPTION;
GRANT 'admin-dbtl1' TO 'admin-db2', 'app'@localhost;
GRANT 'admin-db2' TO 'app-middleware-db2'@localhost;
```



Authorization

MySQL Privileges



- Grants X privileges on Y to Z (with grant options):
 - Administrative Privileges: Global privileges, enabling users to manage the server
 - Database Privileges: Privileges specific to database(s)

```
Grant all privileges on *.* TO 'test'@'%' with grant option;
```
- Fine-grained controls over user privileges:
 - Create, alter, and delete databases
 - Create, alter, and delete tables; create or delete indexes
 - Create, execute, or delete stored procedures and with what rights
 - Execute INSERT, SELECT, UPDATE, DELETE queries



Resetting MySQL Privileges

- If you forget the password to administrative accounts, you are unable to administer MySQL.
- Disable access control by launching MySQL with --skip-grant-tables.
 - This option also enables --skip_networking to prevent networked users from logging in while security is disabled.
- When you disable access control, every user that logs in has root privileges.
 - After resetting the root password, enable the privilege system immediately.

For more details on resetting permissions, see the documentation at
<https://dev.mysql.com/doc/refman/8.0/en/resetting-permissions.html>

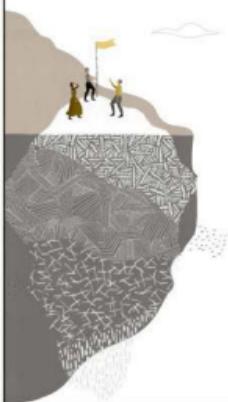
Dynamic Privileges



- Granted by the server at runtime, or by administrators using GRANT commands
- Include administrative privileges
 - Examples: SYSTEM_USER, BINLOG_ADMIN, SYSTEM_VARIABLES_ADMIN,
- Can be defined and granted by plugins and components
 - They grant privileges associated with those plugins
- Examples: FIREWALL_ADMIN granted by the mysql_firewall plugin, AUDIT_ADMIN granted by the audit_log plugin

For more details on dynamic privileges, see the documentation at
<https://dev.mysql.com/doc/mysql/en/privileges-provided.html>

Summary



Regulatory Compliance

Mitigating Risks

Authentication

Authentication in MySQL Enterprise Edition

Authorization



MySQL Enterprise Security Tools



Objectives



MySQL Enterprise Audit

MySQL Enterprise Firewall

MySQL Enterprise Encryption

MySQL Enterprise Audit

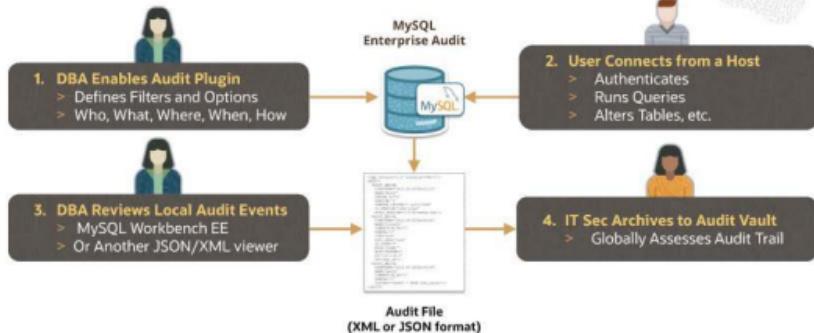


MySQL Enterprise Audit

- > Out-of-the-box logging of connections, logins, and query
 - Auditing for Security and Compliance: GDPR, PCI DSS, Sarbanes-Oxley, HIPAA, FIPS, ...
 - You see who has done what: Connections, logins, and queries.
- > Secure SQL access to audit events
- > Simple to fine-grained policies for filtering and log rotation
- > Dynamically enabled, disabled: No server restart
- > XML and JSON audit stream formatting options
- > Compression and encryption
- > Implemented as component (recommended) or plugin

Adds regulatory compliance
to MySQL applications
(GDPR, HIPAA, PCI DSS, etc.)

MySQL Enterprise Audit Workflow



Audit Filtering

- > Allows DBAs to “custom” design the audit process
- > Uses very fine-grained rules:
 - Reduces the audit log file size
 - Reduces File System IO and Storage/increases performance (less items are logged)
 - Increases the audit log post processing efficiency – less data to process for immediate answers
 - Is defined using JSON
- > Completes audit data:
 - Who, What, When, How, Status
 - From Where
 - DB version, OS version, options
 - And more

MySQL Enterprise Firewall

MySQL Enterprise Firewall

- > Provides Real-Time Protection
 - Queries checked against Allow List and Firewall Rules
- > Blocks SQL Injection Attacks
 - Blocks out-of-policy transactions
- > Detects Intrusion
 - Detects and alerts out-of-policy transactions
- > Creates Automated Allow List
 - Learning mode for creation of approved list of SQL command patterns on a per user or profile (multiple users) basis
- > Is Transparent
 - No changes to application required
- > Combines Firewall/Audit Rules
 - Create more general allow/deny firewall rules using JSON syntax

MySQL Enterprise Firewall

- > Use it to block SQL Injection Attacks:
 - Allow SQL Statements that match Whitelist.
 - Block SQL statements that are not on Whitelist.
- > Use it as an Intrusion Detection system:
 - Detect SQL statements that are not on Allowlist.
 - SQL statements execute and alert administrators.



MySQL Enterprise Encryption

MySQL Enterprise Encryption is a feature of MySQL Enterprise Server that provides transparent encryption for data at rest and in transit.

MySQL Enterprise Encryption uses industry-standard encryption algorithms and key management to protect sensitive data.

MySQL Enterprise Encryption is available as a separate component or as part of the MySQL Enterprise Server license.

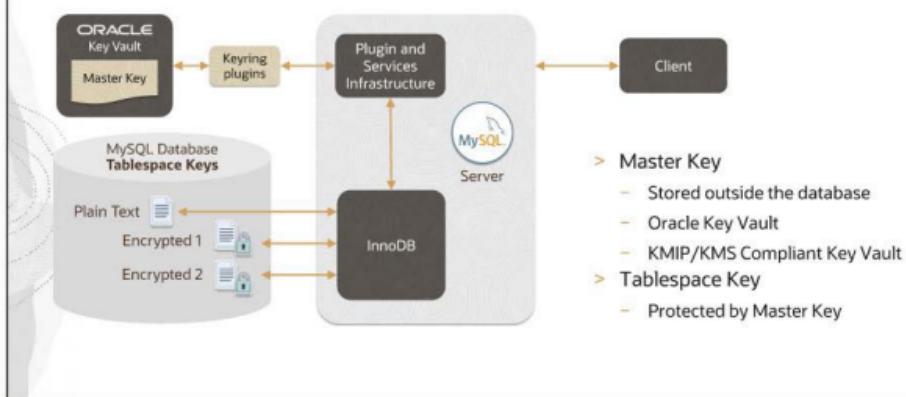


MySQL Enterprise

Transparent Data Encryption (TDE)

- > Protects against attacks on database files
- > Data at Rest Encryption
 - [System | General | Data Dictionary] Tablespaces, Undo/Redo, and Binary/Relay logs
 - Strong encryption – AES 256
- > Is transparent to applications and users
 - No application code, schema, or data type changes
- > Is transparent to DBAs
 - Keys are hidden from DBAs – no configuration changes.
- > Requires key management
 - Protection, rotation, storage, recovery
 - Integrates with KMIP 1.1 and KMS

MySQL Enterprise TDE Architecture



MySQL Enterprise Masking and De-Identification

- > It is implemented as a component (recommended) or a plugin.
- > Data de-identification helps database customers **improve security** and helps **accelerate compliance**.



Production Database



Dev Database
Test Database
Analytics Database

Employee Table

| ID | Last | First | SSN |
|------|-----------|---------|-------------|
| 1111 | Smith | John | 555-12-5555 |
| 1112 | Templeton | Richard | 444-12-4444 |

Masked View

| ID | Last | First | SSN |
|------|-----------|---------|-------------|
| 2874 | Smith | John | XXX-XX-5555 |
| 3281 | Templeton | Richard | XXX-XX-4444 |

<https://dev.mysql.com/blog-archive/exporting-masked-and-de-identified-data-from-mysql/>

MySQL Enterprise Masking and De-Identification

> Data Masking

- String
- Dictionary based
- Specific
 - Social Security Number
 - Payment card (strict/relaxed)

> Random Data Generator

- Random number within a range
- Email addresses
- Payment card (Luhn check compliant)
- Social Security Number
- Dictionary-based generation

```
mysql> SELECT mask_ssn('909-63-6922'),  
mask_ssn('abcdefghijkl');  
+-----+-----+  
| mask_ssn('909-63-6922') | mask_ssn('abcdefghijkl') |  
+-----+-----+  
| XXX-XX-6922           | XXX-XX-hijkl          |  
+-----+-----+  
mysql> SELECT mask_pan('378282246310005');  
+-----+  
| mask_pan('378282246310005') |  
+-----+  
| XXXXXXXXXXXXXXX005        |  
+-----+
```

```
mysql> SELECT genRndEmail();  
+-----+  
| genRndEmail() |  
+-----+  
| a10cv_mevrhuf@example.com |  
+-----+  
mysql> SELECT mask_pan_relaxed(genRndPan());  
+-----+  
| mask_pan_relaxed(genRndPan()) |  
+-----+  
| 398403XXXXXX9547          |  
+-----+
```

Summary



MySQL Enterprise Audit

MySQL Enterprise Firewall

MySQL Enterprise Encryption



MySQL Backup



Objectives



Backup Types

Backup Types: Advantages and Disadvantages

MySQL Enterprise Backup: Features and Benefits

Backup Types

Why back up?

Data is your most valuable IP. You must protect it from loss.

Backup:

- Ensures complete business continuity for disaster recovery
- Helps in case of hardware, software, developer, or DBA errors
- Is used in migration and upgrades (software and hardware)
- Can be used to create replicas
- Archives data (e.g., regulations that require to keep data for X number of years)
- Moves chunks of data
- Can be used to set up test environments

Backup and Recovery Can Be Trickier Than It Appears

- > Requires some forethought and planning
 - How do you back up your critical production systems?
 - Backups consume resources: When is the best time?
 - Down time is not acceptable
 - What is the proper backup storage?
- > Plan your backups, monitor backups, and verify their consistency
- > Plan to perform tests of your restore process
- > How integrate with existing document backup policies

Backup Types: Logical

- > Converts database and tables to SQL statements
 - Portable
 - Server must be running for export/import
 - Generally slower than physical backups



Tools:

- mysqldump
- MySQL Shell
- Data export

Backup Types: Physical

- > Based on the "raw image" of the DB file(s)
- > Require file stability during the backup.
 - DB files must not be changed during the backup.
 - Otherwise, the restored server behaves as if it is restarting after a crash.



Tools:

- MySQL Enterprise Backup
- File system copies
- Snapshot
 - (LVM, HW, Virtualization)

Backup Types

> Others

- Activate and save binary logs for Point-In-Time Recovery (PITR)
- Use MySQL Replication for offline backups
- Use MySQL Replication with replica lag
- Use transportable tablespaces



Full and Incremental Backups



Full Backups

- A full image (physical/logical) of the database
- Restores a full copy of the DB

Incremental backups

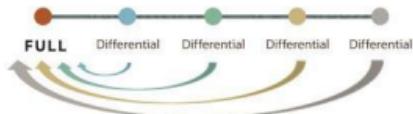
- A backup image with the difference from the last point of backup that can be incremental or a full backup
- Restores **the last full backup and in the right sequence all the previous incremental backups (strict dependencies)** to the required restore point



Full and Differential Backups

Differential backups

- A backup image with the difference from the last FULL backup
- Restores **the last full and the differential backups that you need**



Backup Types: Advantages and Disadvantages

Storage Snapshots



Advantages

- Quick
- Feature of Linux LVM, SAN, NAS, and virtualized environments
- Good to use in conjunction with backups

Disadvantages

- For consistency, it requires a service stop (or put it in read only)
- It's a snapshot.
 - Still, a backup copy is required to be made – which can be "full" in size.
- Performance may degrade with each concurrent snapshot.
 - Snapshots need to be released.
- It has cross-file system limitations.

mysqldump

<https://dev.mysql.com/doc/refman/8.0/en/mysqlpump.html>

- > Create a SQL file that rebuilds the instance.
- > Advantages
 - **Good for small databases or tables**
 - **Produce a text file:** flexible and portable
- > Disadvantages
 - **Cannot back up an active instance:** requires MVCC or table locks
 - **Not consistent across the backup:** unless you use `--single-transaction`
 - **Not incremental or differential:** requires a full backup every time
 - **Slower restore times than with physical backups**
- > Use mysql client to restore the backup.

mysqldump examples

- > Export of all databases for replication, including events and procedures:

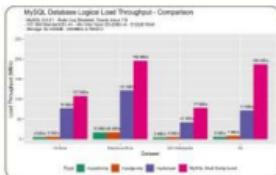
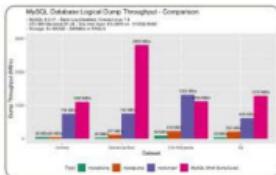
```
mysqldump [...] --all-databases \
  --single-transaction --routines \
  --events --master-data > full_backup.sql
```

- > Import of the dump:

```
mysql -uroot -p -S /tmp/mysql_restore.sock < full_backup.sql
```

MySQL 8 Shell: Parallel Export/Import

- > Supports the dump and import of schemas or tables
 - Data streaming from remote storage
 - Parallel loading of tables or table chunks
 - Progress state tracking
 - Resume and reset capability
 - Add primary keys to tables
 - ...
- > Commands:
 - `util.dumpInstance()` : Dump an entire database instance, including users
 - `util.dumpSchemas()` : Dump a set of schemas
 - `util.loadDump()` : Load a dump into a target database



<https://dev.mysql.com/blog-archive/mysql-shell-dump-load-part-2-benchmarks/>

MySQL Replication and Backups

- > A replica is not a backup, but it can improve the backup policy.
 - Doesn't protect against human error if the error is propagated to the replica
- > Advantages
 - Rolling "snapshot"
 - Non-blocking
 - You can back up the replica to avoid overhead on the source instance
- > Disadvantages
 - Only latest point in time (point in time keeps moving forward)
 - Not historical
 - Not for archival purposes

MySQL Enterprise Backup

Advantages

- Physical backup and restore operations are very fast compared to logical backups
- Flexible - many options
- Archival
- Scalable
- Consistent
- Supported

Disadvantages

- Cannot be used to perform upgrades
- Not available in Community Edition

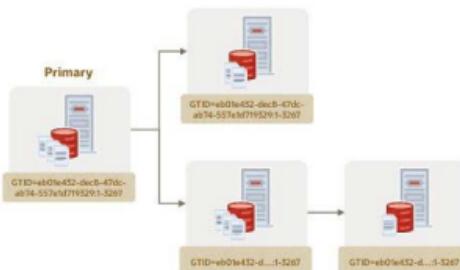
Binary Logs

- > Set of numbered files containing changes
 - For example, binlog.000001, binlog.000002, etc.
- > Three possible formats
 - Statement-based replication (SBR): Propagation of SQL statements
 - Row-based replication: Logs column changes to individual rows
 - Mixed-format replication: SBR is default, but switches to row-based
- > Rotate/flush logs with FLUSH BINARY LOGS
 - Closes the logs but does not delete the logs
- > Purge can be manual or automatic
 - SHOW BINARY LOGS;
 - PURGE BINARY LOGS TO 'mysql-bin.000010';
 - PURGE BINARY LOGS BEFORE '2019-04-02 22:46:26';
 - binlog_expire_logs_auto_purge, expire_log_days, binlog_expire_logs_seconds
- > Can be non-transactional (legacy) or transactional (GTID)

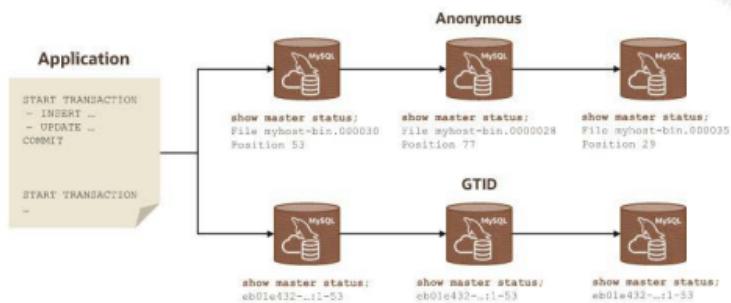


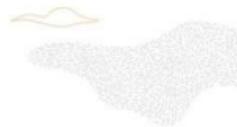
Global Transaction IDentifier (GTID)

- > A unique identifier created and associated with each transaction committed on the server of origin (the source)
 - <Server UUID>: <seq. number>
 - The transaction has the same GTID in every node involved in replication topology.
- > Unique across all servers in a given replication topology



GTID vs. Anonymous Binary Log





Point-in-time Recovery: Process

- Restore last backup.
- Check the actual position (GTID/LSN).
 - mysql> SHOW MASTER STATUS;
- Execute events recorded in the binary using mysqlbinlog utility.
 - Specify all the binary logs in one command.
 - If possible, specify whenever recovery has to start or stop in terms of time or event position.
 - --start-datetime and --stop-datetime
 - --start-position and --stop-position
 - mysqlbinlog binlog_files | mysql -u root -p
 - As an alternative to mysqlbinlog, use replication to apply all available binary logs.
 - (KB 2009693.1 and KB 2277457.1)

How to Save Binary Logs

- > Make a physical copy of the files on a remote system:
 - Requires scripting: Ensure that the binary log file is closed when copied
- > Create a replica that has a copy of the binary logs using mysqlbinlog or replicas.
 - mysqlbinlog is not a daemon: monitor it!
 - <https://dev.mysql.com/doc/refman/8.0/en/mysqlbinlog-backup.html>
 - [KB 2180573.1](#)

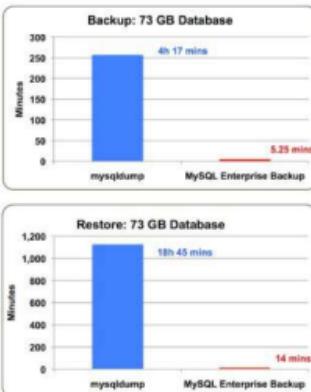


MySQL Enterprise Backup: Features and Benefits



MySQL Enterprise Backup

- > Online, non-locking backup, and recovery
 - Complete MySQL instance backup (data and config)
 - Partial backup and restore
- > Supports direct-to-cloud storage backups
 - Oracle Storage Cloud, S3, etc.
- > Incremental backups
- > Point-in-time recovery
- > Supports compression and encryption
- > Supports *System Backup to Tape API* (SBT)
- > Optimistic backups



MySQL Enterprise Backup

- > Cross-platform (Windows, Linux, Unix)
- > Available as part of MySQL Enterprise Edition
 - Either with MySQL Server Enterprise Edition or as a separate package.
- > Download from:
 - <http://edelivery.oracle.com> (trial, only most recent release)
 - <http://support.oracle.com> (all releases)



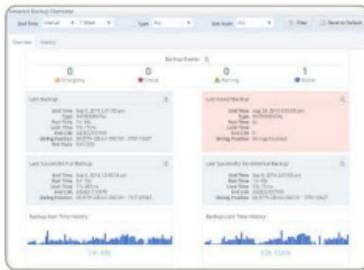
MySQL Enterprise Backup: Instance Support

- > MySQL Enterprise Backup comes with both Long-Term Support (LTS) and Innovation Releases.
 - LTS: Supports only instances of the same LTS release
 - The last version available of MySQL Enterprise Backup is recommended.
 - Example: MySQL Enterprise Backup 8.4 (LTS) can back up MySQL Enterprise Edition 8.4, but not 8.0.37.
 - Innovation: Supports only instances within the same LTS release
 - Example: MySQL Enterprise Backup 8.7 (Innovation) can back up MySQL Enterprise Edition 8.4, 8.5, but not 8.0.



MySQL Enterprise Backup: Integration with MySQL Enterprise Monitor

- > MySQL Enterprise Monitor provides a backup dashboard.
- > Monitors the backup usage and health
 - Across your entire datacenter
- > Drills into backup job details
 - Allowing for easy backup recovery
- > Supports all backup types
- > Provides alerts on significant events
 - Poor backup performance
 - Backup job failures
 - Out-of-date backups



Backup Process

Short Summary

- > The InnoDB data files, redo log, binary log, and relay log files (except for the log files currently in use) are copied into the backup, while the database server operates as usual.
- > A backup lock is applied on the server instance.
- > The FLUSH TABLES ... WITH READ LOCK statement is applied on all non-InnoDB tables.
- > A brief blocking of logging activities on the server is applied, to collect logging-related information such as the current InnoDB LSN, binary log position, GTID, and so on.
- > The read lock on the non-InnoDB tables is released.
- > Using information from step 4, the relevant portion of the binary or the relay log file currently in use is copied.
- > The backup lock on the server instance is released.
- > The redo log files not yet copied before, as well as all the metadata files for the backup, are copied or created.
- > The backup operation is completed.



A **backup lock** is applied on the server instance.

- It blocks DDL operations (except for those on user-created temporary tables), but not DML operations (except for those not captured by the binary log, such as administrative changes to the database) on InnoDB tables.

MySQL Enterprise Backup Files

- > MySQL Enterprise backup saves the datadir content.
- > Raw files are backed up with mysqlbackup.
 - InnoDB data
 - ibdata* files : Tablespace files
 - .ibd files : Per Table data files
 - ib_logfile* files : Log files
 - All other files
 - .MYD / .MYI : MyISAM data and index files
 - others : relay log, binlog, config files, etc. ...



Restore Process



1

Clean the destination directory.

2

Restore the database using mysqlbackup with:



- > `copy-back-and-apply-log` or
- > `copy-back`

3

Depending on how you are going to start the restored server, you might need to adjust some files.

- > Ownership of the restored data directory
- > my.cnf/mi.ini configuration files
- > mysqld-auto.cnf
- > auto.cnf

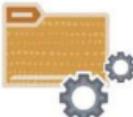
Configuration Files

- > It doesn't save the original my.cnf or my.ini and creates copies of the configuration file.
 - Use a standard OS backup for the original my.cnf and my.ini
- > It creates configuration files for restoring.
 - backup-my.cnf: Records the crucial configuration parameters that apply to the backup
 - **server-my.cnf**: Contains values of the backed-up server's global variables that are set to non-default values
 - server-all.cnf: Contains values of all the global variables of the backed-up server



Configuration Files

- > Restore rename mysqld-auto.cnf to backup-mysqld-auto.cnf
 - It contains persisted variables.
 - It's important to restore the original name when you want to restore the original instance or create a replica.
- > Restore rename auto.cnf to backup-auto.cnf
 - It contains server UUID.
 - It's important when you want to restore the original instance.
 - Don't rename to create a new replica.



MySQL Enterprise Backup Examples

> Full backup:

```
mysqlbackup --defaults-file=/etc/my.cnf --port=3306 --host=127.0.0.1 --protocol=tcp  
--user=root \  
--backup-dir=/home/mysql/backup/full backup
```

> Restore:

```
mysqlbackup --defaults-file=/etc/my.cnf --backup-dir=/home/mysql/backup/full \  
copy-back-and-apply-log
```

> Incremental backup (InnoDB):

```
mysqlbackup --defaults-file=/etc/my.cnf  
--incremental --incremental-base=history:last_backup --backup-  
dir=/home/dbadmin/temp_dir \  
--backup-image=incremental_image1.bi backup-to-image
```

MySQL Enterprise Backup Options

Just some of the many options that can be used

> Encryption:

- --encrypt (--decrypt to restore)
- --key=<hex key string>
- --key-file=<keyfile>
- Generating key-file : e.g., `openssl rand 32 -hex > mykeyfile`
- It's only valid for images.

> Compression:

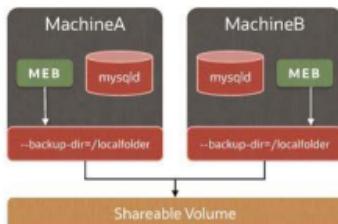
- --compress, --compress-level=[0 ~ 9]
- Only InnoDB data files are compressed.

> Using the SBT Interface:

- Back up directly to your company's media management software (MMS).

MySQL Enterprise Backup Deployment – Example

- > Multiple MySQL Instances deployment
- > Back up to local folder/image file
 - Local folder: Specified with `--backup-dir`
 - The local folder can be mounted to external NFS or shared volume.
- > Timestamp
 - option: `--with-timestamp`
 - Create a folder with a timestamp and store multiple copies of a backup based on timestamp



Summary



Backup Types

Backup Types: Advantages and Disadvantages

MySQL Enterprise Backup: Features and Benefits



MySQL Replication



Objectives



Overview

Asynchronous and Semisynchronous Replication

Replication Topologies

Overview

MySQL is a relational database management system (RDBMS) used for storing, retrieving, and managing data.

It is developed by Oracle Corporation and is one of the most popular RDBMSes in the world.

MySQL is open-source and can be used for both personal and commercial purposes.

It is a fast, reliable, and easy-to-use database system.

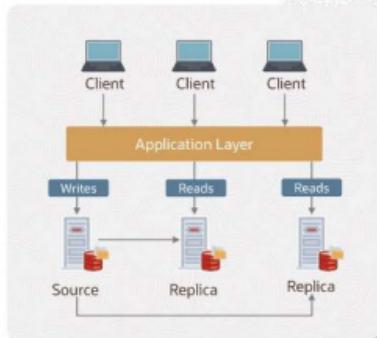
MySQL supports a wide range of programming languages and tools for interacting with the database.

It is widely used in web development, data warehousing, and other applications.



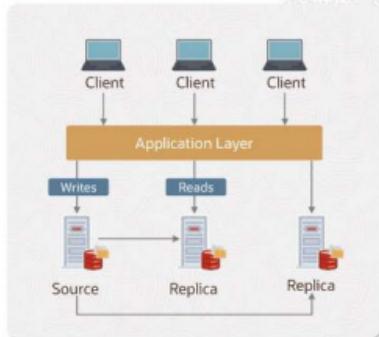
MySQL Replication

- Data is copied from source to replica.
- Replication is asynchronous by default.
 - Can be also semi-synchronous
- Multiple sources or replicas are possible.
 - Additional replicas don't require source configuration.



MySQL Replication Use Cases

- › Scale-out solutions for read-heavy environment
 - Scaling out doesn't involve editing the `my.cnf` file.
- › Disaster recovery
- › Simple high availability (requires scripts and/or external tools)
- › Analytics
- › Long-distance data distribution



Replication Steps – Summary

1

Create replication user on source

2

Check Source configuration
- SHOW BINARY LOG STATUS
- Confirm GTID is enabled

3

Configure replica server
- CHANGE REPLICATION SOURCE TO

4

Start replica
- START REPLICA
- SHOW REPLICA STATUS

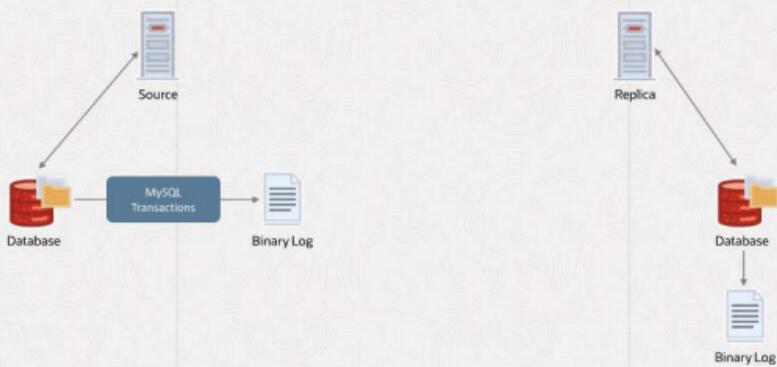
After both the source and target servers have the same dataset:

- Create replication user on source
- Check Source configuration
 - SHOW BINARY LOG STATUS
 - Confirm GTID is enabled
- Configure replica server
 - CHANGE REPLICATION SOURCE TO
- Start replica
 - START REPLICA
 - SHOW REPLICA STATUS

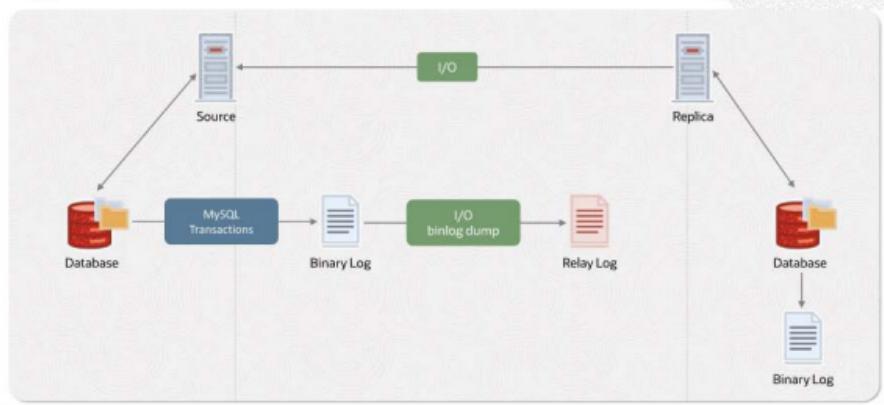
Asynchronous and Semisynchronous Replication



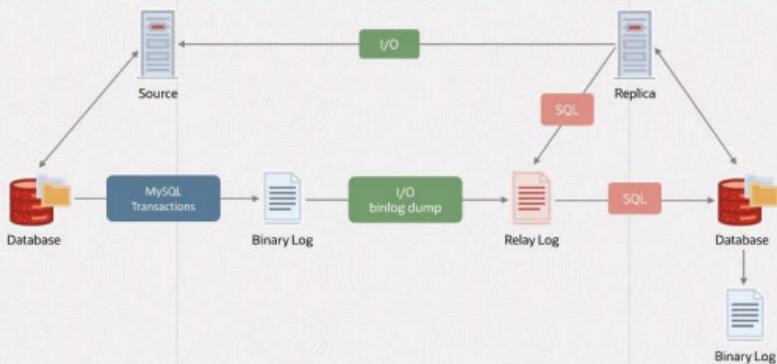
MySQL Asynchronous Replication



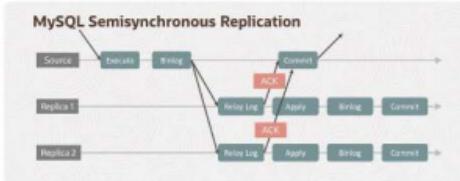
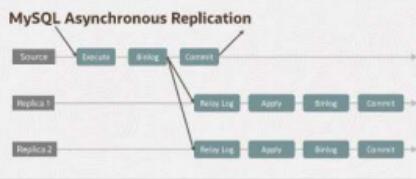
MySQL Asynchronous Replication



MySQL Asynchronous Replication

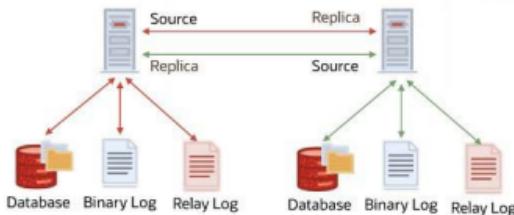


MySQL Replication Communication



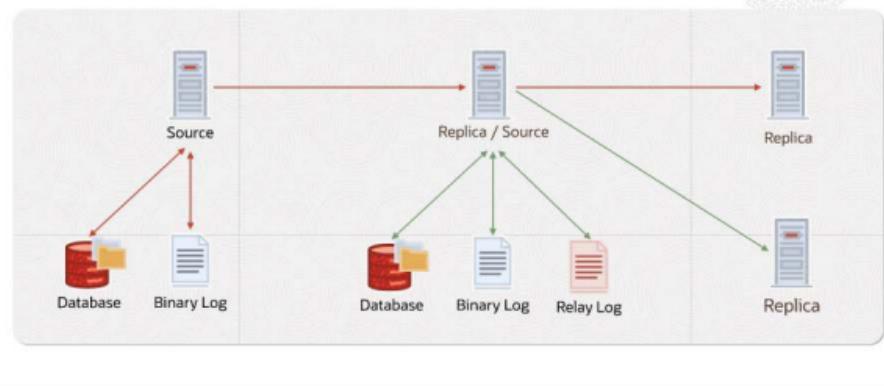
Replication Topologies

MySQL Source-Source

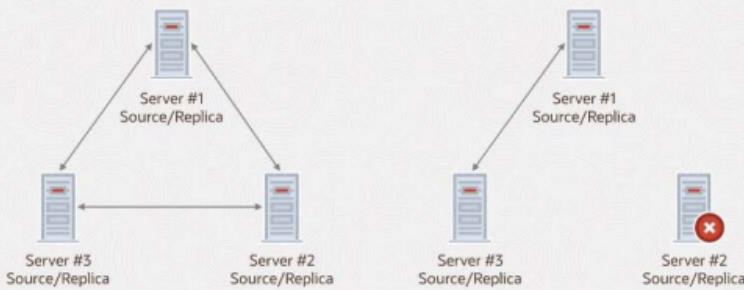


- No conflict resolution
 - High risk of corruption due to asynchronous changes
- Set auto_increment options to reduce risk of conflict, for example:
 - FIRST SERVER: `auto_increment_offset=1; auto_increment_increment=2`
 - SECOND SERVER: `auto_increment_offset=2; auto_increment_increment=2`

MySQL Chain

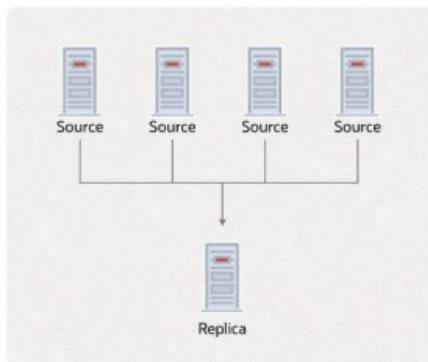


MySQL Replication – Ring Topology

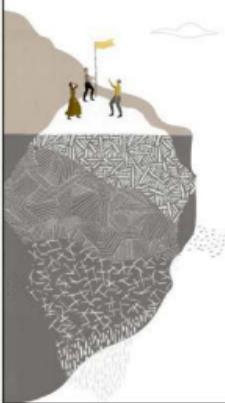


Multi-Source Replication

- Consolidate updates from multiple sources into one replica
- Compatible with semi-sync replication
- Online operations
 - dynamic replication filters, switch source



Summary



Overview

Asynchronous and Semisynchronous Replication

Replication Topologies



Enabling High Availability in MySQL



Objectives



Overview

Components

Split Brain and Network Partitions

Monitoring

Complex Topologies

Overview

MySQL is a relational database management system (RDBMS) used for storing, retrieving, and managing data.

It is developed by Oracle Corporation and is one of the most popular RDBMSes in the world.

MySQL is open-source and can be used for both personal and commercial purposes.

It is a fast, reliable, and easy-to-use database system.

MySQL supports a wide range of programming languages and tools for interacting with the database.

It is widely used in web development, data warehousing, and other applications.

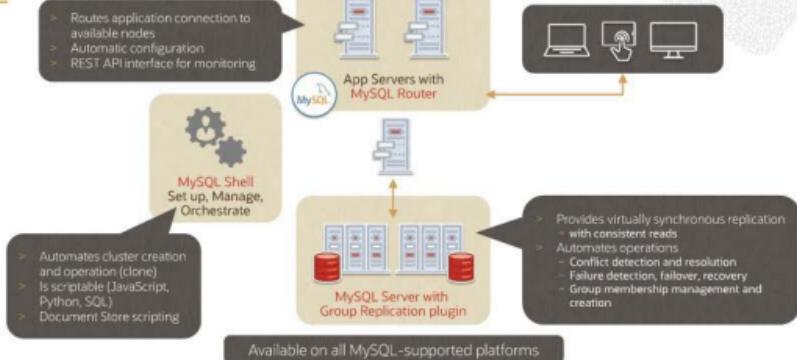
MySQL has a large user base and a strong community of developers and enthusiasts.

It is a powerful and flexible database system that can handle large amounts of data and complex queries.

MySQL is a great choice for anyone looking for a reliable and efficient database system.

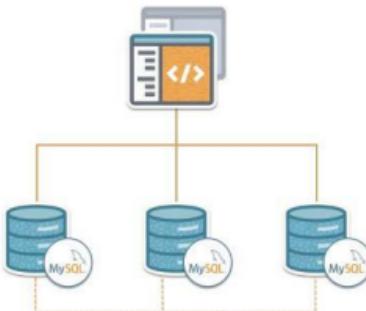


InnoDB Cluster



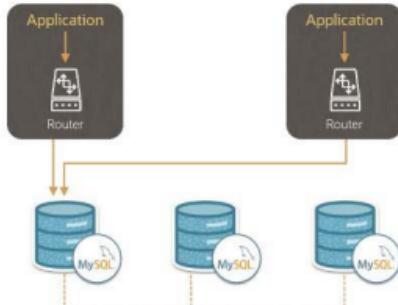
InnoDB Cluster: Setup

- > Management of the cluster is easy with MySQL Shell: Creation, status check, configuration changes, and cloning.
- > A cluster is set up with a remote connection to three instances.



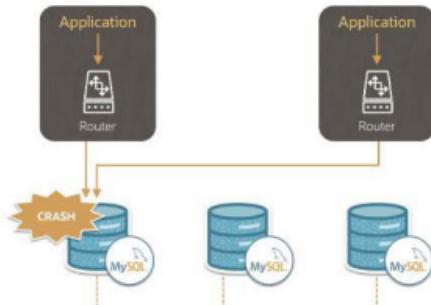
InnoDB Cluster: Application Access

- > Start Router with the “bootstrap” option that automatically creates the Router configuration.



InnoDB Cluster: Application Access

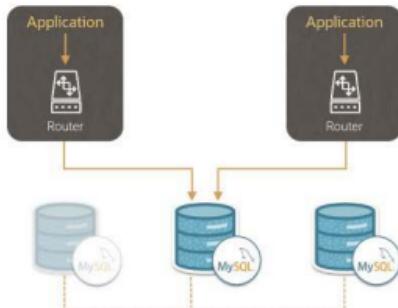
- > The application accesses the database through MySQL Router.
- > If the primary server fails, MySQL Router performs a *failover*.



<https://dev.mysql.com/doc/refman/8.0/en/group-replication-failure-detection.html>

InnoDB Cluster: Application Access after Failover

- > After failover, MySQL Router redirects application queries to an available node.
- > No application configuration changes are required; the failover happens automatically.



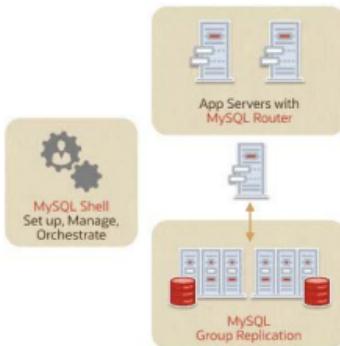
Hardware Configuration

- > All servers in a cluster should have similar specifications.
- > Server resources should be isolated for MySQL use.
- > The number of servers in the group must be 3, 5, 7, or 9.
- > Ideally, storage hardware should be SSDs.
- > Use redundant hardware where possible:
 - Storage
 - Power supplies
 - Network cards

Components

InnoDB Cluster Components

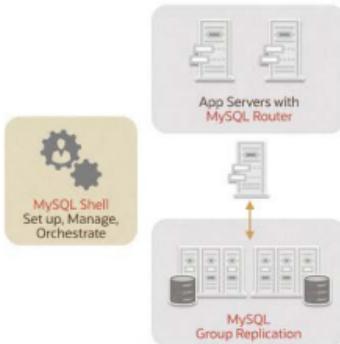
- **MySQL Shell:** User interface
- **MySQL Router:** Directs queries to the correct server
- **MySQL Group Replication:** Manages the continuity of service



MySQL Shell for Configuring InnoDB Cluster

A single unified client for all administrative and operations tasks

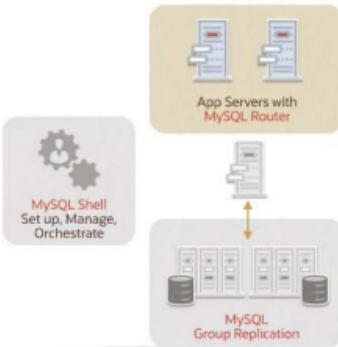
- > Multi-language: JavaScript, Python, and SQL
 - Naturally scriptable
- > Supports both Document and Relational models
- > Exposes full Development and Admin API
- > Performs DBA operations
- > Creates and manages MySQL InnoDB clusters, ReplicaSet and ClusterSet



MySQL Router

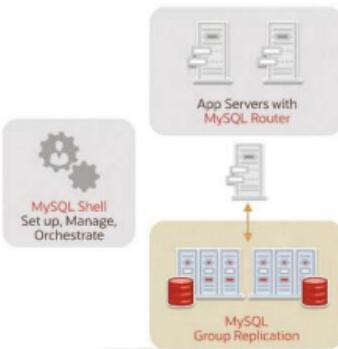
Transparent access to HA databases for MySQL Applications

- It offers transparent client connection routing.
 - Load balancing
 - Application connection failover
- Stateless design offers easy HA client routing.
 - A local router becomes part of the application stack (installed in every application server).
 - It can be installed on dedicated servers.
 - But HA has to be provided by third-party SW.



MySQL Group Replication

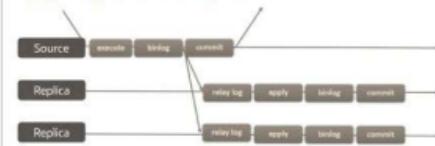
- > Group Replication library:
 - Provides *virtually synchronous* replication for MySQL
 - Is supported on all **MySQL platforms**
 - Linux, Windows, Solaris, OSX, FreeBSD
 - Automates operations
 - Conflict detection and resolution
 - Failure detection, failover, recovery
 - Group membership management and reconfiguration



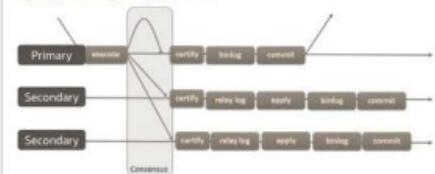
Group Replication

- Communication layer provides:
 - Atomic messaging
 - Total order message delivery
- Multi-source update everywhere
 - R/W transactions commit after group approval.
 - Conflicts are auto-detected and managed (first-in wins).
- Very large transactions that cannot be copied between group members over the network within a 5-second window can cause failures in the group communication.

MySQL Asynchronous Replication



MySQL Group Replication



Split Brain and Network Partitions

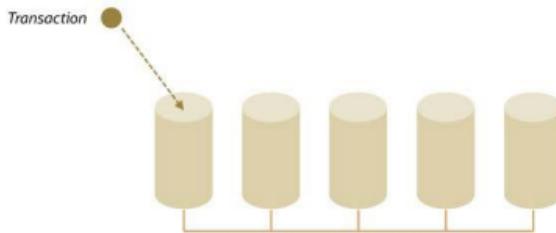
When a cluster is partitioned, it can lead to a split brain where two groups of servers each think they are the primary.

This can happen if a partition occurs between the master and slave replication threads.

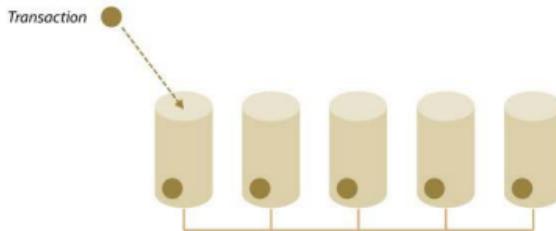
MySQL 8.0 includes a feature called "split brain detection" which can help prevent this from happening.



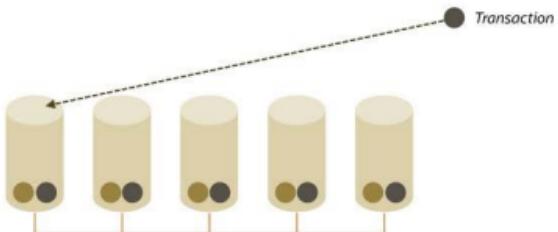
Split-Brain Scenario: Working Cluster



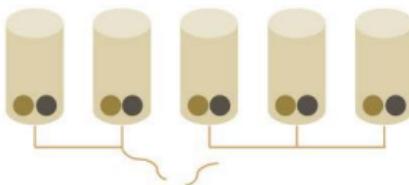
Split-Brain Scenario: Working Cluster



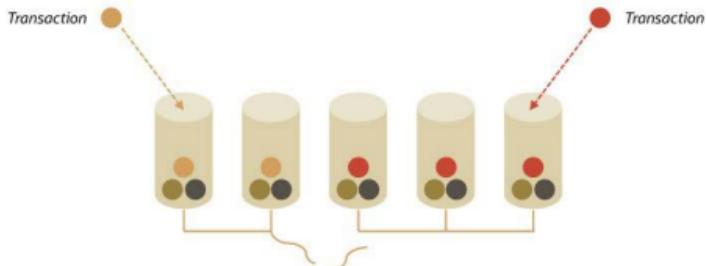
Split-Brain Scenario: Working Cluster



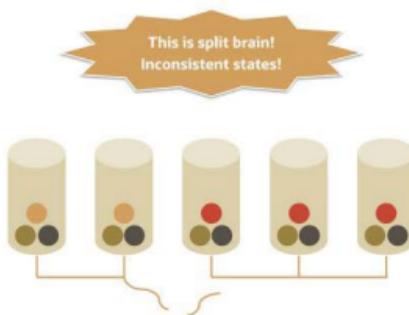
Split-Brain Scenario: Network Split



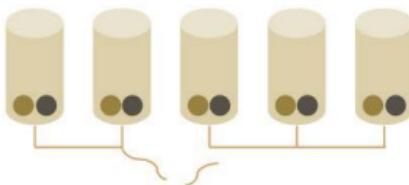
Split-Brain Scenario: What happens if we do nothing?



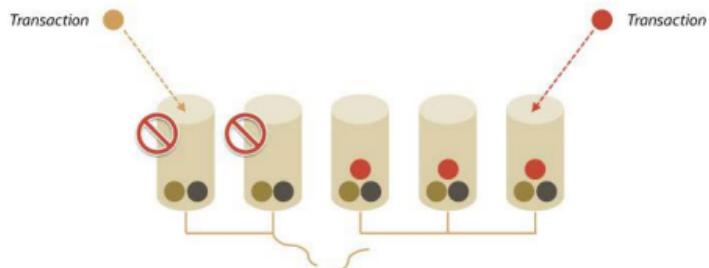
Split-Brain Scenario: What happens if we do nothing?



Group Replication: Network Split



Group Replication: Split-Brain Prevention



Network Partitioning

- > Group replication requires a majority of members (consensus) to agree on a given decision.
- > A loss of a majority of members results in shutdown.
- > Loss of a single server results in group reconfiguration.
- > The minimum number in a group is 3.
- > The maximum number in a group is 9.

| Group Size (n) | Majority | Failures Tolerated (f) |
|----------------|----------|------------------------|
| 1 | 1 | 0 |
| 2 | 2 | 0 |
| 3 | 2 | 1 |
| 4 | 3 | 1 |
| 5 | 3 | 2 |
| 6 | 4 | 2 |
| 7 | 4 | 3 |
| 8 | 5 | 3 |
| 9 | 5 | 4 |

Monitoring



Monitoring Group Replication

> MySQL Enterprise Monitor

- Automatic discovery of InnoDB Clusters
- Dedicated advisors
- Automatic update in case of topology changes

> Performance Schema Tables

- `performance_schema.replication_group_member_stats`
- `performance_schema.replication_group_members`
- `performance_schema.replication_connection_status`
- `performance_schema.replication_applier_status`

> Replication Channels Created:

- `group_replication_recovery` - changes to replication
- `group_replication_applier` - applies incoming transactions



Group Replication Server States



| STATUS | DESCRIPTION | GROUP SYNCHRONIZED |
|-------------|--|--------------------|
| ONLINE | The server is an active member of a group and in a fully functioning state. | Y |
| RECOVERING | The server has joined a group and is in the process of becoming an active member. Distributed recovery is currently taking place using a remote cloning operation or binary log. | N |
| OFFLINE | The Group Replication plugin is loaded, but the member does not belong to any group. | N |
| ERROR | The member is in an error state and is not functioning correctly as a group member. | N |
| UNREACHABLE | The local failure detector suspects that the member cannot be contacted, because the group's messages are timing out. | N |

InnoDB Cluster: Shared Data

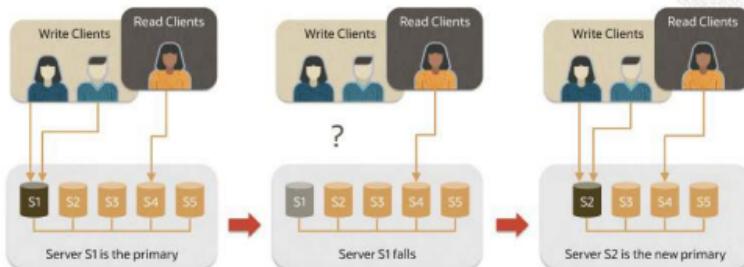
- > MySQL Shell saves metadata for InnoDB Cluster in the database `mysql_innodb_cluster_metadata`.
 - Metadata are replicated on every node.
- > The users used to manage replication are automatically generated.
 - `mysql_innodb_cluster_...`
 - `mysql_routerl_...`
 - Use `<cluster>.resetRecoveryAccountsPassword()` to reset the passwords for the internal recovery accounts created by InnoDB Cluster.
- > Administrative user for both cluster and router can be created and managed with MySQL Shell:
 - `dba.configureInstance()`, `<cluster>.setupRouterAccount()`, and `<cluster>.setupAdminAccount()`

29

Single-primary and Multi-Primary Configurations

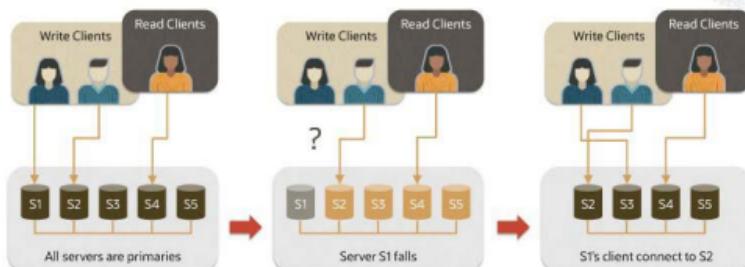


Single-Primary Mode



<https://dev.mysql.com/doc/refman/8.0/en/group-replication-single-primary-mode.html>

Multi-Primary Mode



Important Notes for Multi-Primary

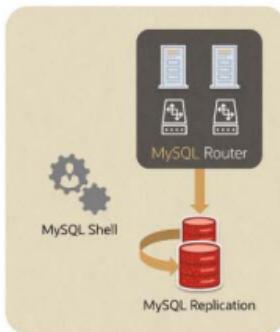
- Optimistic execution: Transactions may abort on commit if there are conflicts with concurrent transactions on other members.
- Group replication is not aware of table locks and named locks.
- Concurrent DDL and DDL+DML execute against the same object but are not supported on different servers.
- Foreign keys with cascading constraints are not fully supported.



Complex Topologies

MySQL Replicaset

- > 'Classic', 'asynchronous' Replication-based solution, fully integrated
- > Integrated MySQL Router:
 - Automatic routing
 - (asynchronous) Read scaleout
- > Ease of use with MySQL Shell:
 - Configuring, adding, removing members
 - Automatic Member Provisioning (CLONE)
- > Switchover and failover (manual)



<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-innodb-replicaset.html>

MySQL InnoDB ClusterSet

Replication between InnoDB Clusters



<https://dev.mysql.com/doc/mysql-shell/8.0/en/innodb-clusterset.html>

High Availability (Failure Within a Region)

- > Recovery point objective (RPO) = 0
- > Recovery time objective (RTO) = seconds (automatic failover)

Disaster Recovery (Region Failure)

- > RPO != 0
- > RTO = minutes or more (manual failover)
- > No write performance impact

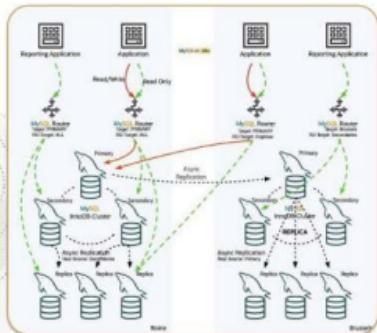
Features

- > Easy to use
- > Add/remove nodes/clusters online
- > No need to reconfigure application if the topology changes

Recovery point objective is the acceptable data loss in a failure. Recovery time objective is the acceptable downtime. Ideally, these are minimal. Different high-availability and disaster recovery plans result in different RPO and RTO, and tradeoffs are frequently required when choosing between potential solutions.

MySQL InnoDB Read Replicas

Improved Scalability and Redundancy



Read Replicas

- > Read-only copies of a cluster member

Scalability

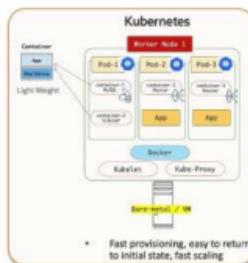
- > Asynchronous replication enables scaling out workload, offloading read requests from your cluster to one, or more, dedicated read-only instances.

Additional Redundancy

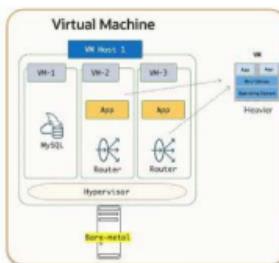
- > Read replicas can be promoted to cluster members.

Fully Managed with MySQL Shell

Kubernetes Vs. Virtual Machine (VM)



VS.

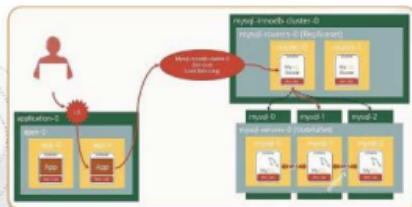


- > A portable, extensible, open-source platform for managing containerized workloads and services, which facilitates both declarative configuration and automation.
- > The desired state and other definitions are defined in a manifest (yaml file).

MySQL and Kubernetes

- > MySQL single instance can be easily deployed with Kubernetes features.
- > A database cluster:
 - Contains multiple nodes
 - Has specific sequences in node's startup/shutdown
 - Has specific sequences in cluster updates
 - Creates multiple yaml files for various resources, such as StatefulSet, ReplicaSets, Services, etc.
 - Scale out/Scale in have requirements.
 - Rolling update has requirements.
- > MySQL Operator for Kubernetes provides an interface to create and manage MySQL clusters where coordination between multiple resources is required.

MySQL Operator for Kubernetes



The MySQL Operator for Kubernetes is designed to operate a MySQL InnoDB Cluster in a Kubernetes environment. It supports:

- > Automated deployment and management of MySQL Server and MySQL Routers
- > Self-healing
- > Backup and restore
- > Rolling upgrades with minimal down time
- > Read scaling
 - Add/remove members as needed
 - Configurable consistency levels
- > The *Helm* package manager

Summary



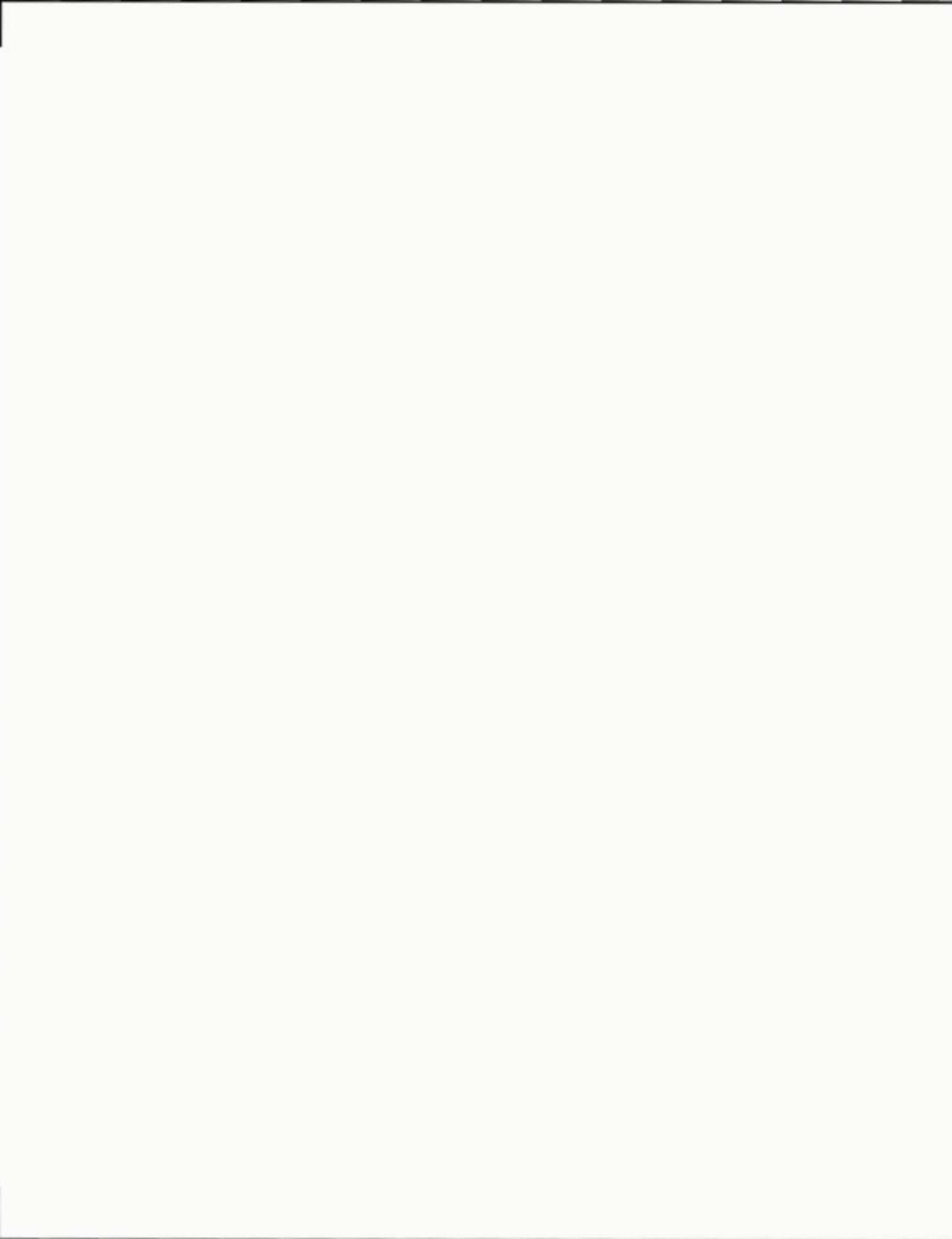
Overview

Components

Split Brain and Network Partitions

Monitoring

Complex Topologies





Monitoring MySQL



Objectives



Monitoring MySQL

MySQL Tools for Monitoring

Monitoring MySQL

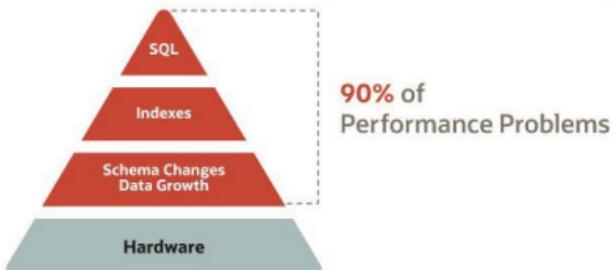




MySQL DBA Challenges

- “The database is slow. What needs tuning?”
- “What are my most expensive queries?”
- “Are indexes optimized?”
- “Is replication lag a problem?”
- “Did my last backup succeed?”
- “When will my disk fill up?”
- “When will I need more hardware to scale out?”
- “Has my database schema changed?”
- “Are there security vulnerabilities that I need to be concerned about?”

Source of Database Performance Problems



MySQL DBA Checklist

| | |
|--|---|
| 1. Ensure your production databases are available | ✓ |
| 2. Monitor MySQL performance throughout the day | ✓ |
| 3. Verify that MySQL replication is working properly | ✓ |
| 4. Confirm that backups have completed successfully | ✓ |
| 5. Monitor disk space to ensure MySQL won't run out of space | ✓ |
| 6. Regularly monitor and identify blocking issues | ✓ |
| 7. Verify there have been no changes to database schema | ✓ |
| 8. Check OS metrics for unusual events | ✓ |
| 9. Check for security vulnerabilities | ✓ |
| 10. Monitor and analyze memory usage | ✓ |

MySQL Tools for Monitoring

Slow Query Log

- > The slow query log consists of SQL statements that take more than long_query_time seconds to execute and require at least min_examined_row_limit rows to be examined.
 - By default, administrative statements are not logged, nor are queries that do not use indexes for lookups.
 - This behavior can be changed using log_slow_admin_statements and log_queries_not_using_indexes.
- > The slow query log can be used to find queries that take a long time to execute and are therefore candidates for optimization.
- > Examining a long slow query log can be a time-consuming task.
 - To make this easier, you can use the mysqldumpslow command to process a slow query log file and summarize its contents.

Performance Schema

- > It provides runtime statistics to monitor MySQL Server execution at a low level.
- > It has a dedicated storage engine PERFORMANCE_SCHEMA.
 - None are persistent (restart of MySQL flush schema)
 - Nearly zero disk interaction
- > Performance schema is not replicated.
- > All recorded statistics are stored in fixed-size ring buffers in memory.



SYS Schema

- > Includes helper objects for DBAs, developers, and operations staff
- > Helps simplify DBA/Ops tasks
 - Monitors server health, user, host statistics
 - Identifies, diagnoses, and tunes performance issues
- > Provides easy-to-understand views with insights into:
 - IO hot spots, locking, costly SQL statements
 - Schema, table and index statistics



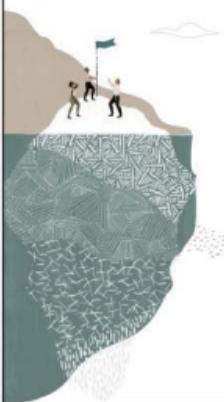
Oracle Enterprise Manager for Monitoring MySQL

Enables monitoring of:

- > Performance
- > Availability
- > Replication
- > InnoDB
- > Firewall
- > Audit Events



Summary



Monitoring MySQL

MySQL Tools for Monitoring



MySQL HeatWave



Objectives



Features and Benefits

Supported Shapes

Multiserver Capabilities

OLTP and OLAP

HeatWave AutoPilot

HeatWave Machine Learning

HeatWave Lakehouse

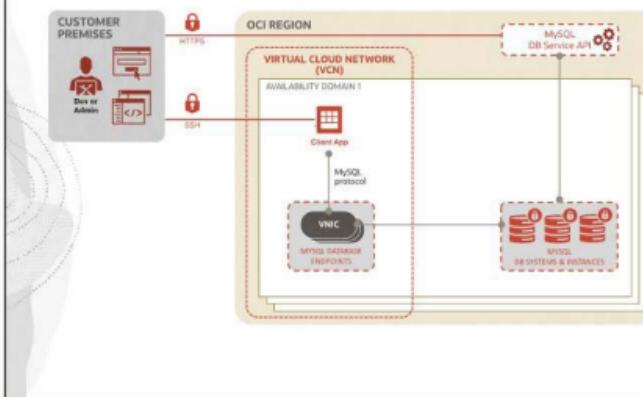
Features and Benefits

MySQL HeatWave Overview

- > Fully managed MySQL Service and very easy to use
 - Deploy, Backup and Restore, High Availability (RPO=0), Resize, Read Replicas, etc.
- > Powerful union of **Oracle Cloud Infrastructure** and **MySQL Enterprise Edition 8**
- > Always up to date with the latest security fixes
- > No need of Extract/Transform/Load processes (ETL)
- > **Use cases:**
 - OLTP only
 - Analytics
 - Query accelerator (OLTP+OLAP)
 - Machine Learning
 - LakeHouse (direct access to Object Store)



Managed MySQL: Focus on Your Business



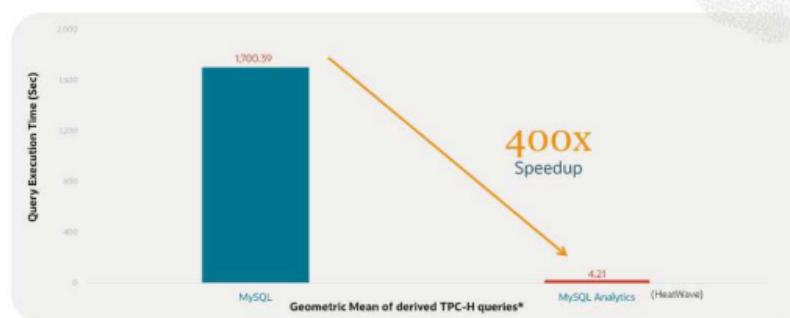
User Responsibility

- Logical schema modeling
- Query design and optimization
- Define data access and retention policies

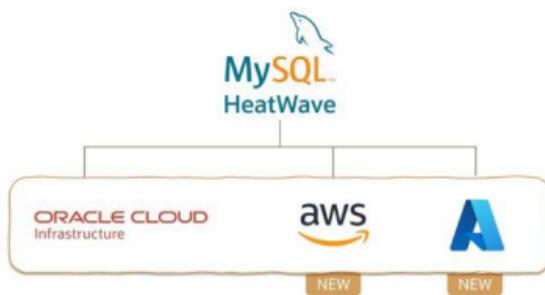
Oracle Responsibility

- Backup and recovery
- Database and OS patching
- Monitoring and log handling
- Security with advanced options available in MySQL Enterprise Edition

HeatWave Speeds Up Analytic Queries



HeatWave Is Available in Public Clouds



Optimized to deliver the best price performance in each cloud

You can deploy MySQL HeatWave in public clouds – it is available on OCI, AWS, and Azure.

HeatWave Performance and Price Summary

TPCH



OLAP Performance

- > 3x faster than Synapse
- > 6.5x faster than Redshift
- > 7x faster than Snowflake
- > 10x faster than Big Query
- > 1100x faster than Aurora
- > 5400x faster than RDS

OLAP Price Performance

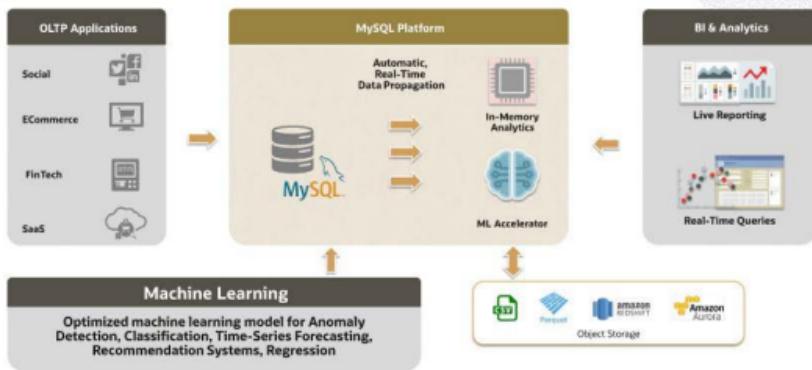
- > 15x better than Synapse
- > 13x better than Redshift
- > 35x better than Snowflake
- > 37x better than Big Query
- > 2200x better than Aurora
- > 7000x better than RDS MySQL

OLAP + OLTP

- > 20x - 500x faster than Aurora
- > 50x - 1200x better price/perf than Aurora

HeatWave

Single System for OLTP, Analytics, and ML



Supported Shapes



MySQL HeatWave Database Service Shapes



| OCPUs | RAM (GB) | AMD E3 (VM.Standard.E3) | AMD E4 (VM.Standard.E4) | Intel X9 (VM.Standard3) | Intel X9 (VM.Optimized3) | Intel X (VM.Standard2) |
|-------|----------|----------------------------|----------------------------|----------------------------|-----------------------------|---------------------------|
| 1 | 8 | ✓ | ✓ | ✓ | ✓ | |
| 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 4 | 64 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 8 | 128 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 16 | 256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| 24 | 384 | ✓ | ✓ | ✓ | | |
| 32 | 512 | ✓ | ✓ | ✓ | | |
| 48 | 768 | ✓ | ✓ | | | |
| 64 | 1024 | ✓ | ✓ | | | |

MySQL HeatWave Shapes – Minimum Configuration

> OLTP

- 1 MySQL Frontend (16 OCPU, 512 GB RAM, 50 GB Storage)

> OLTP + Analytics

- 1 MySQL Frontend (16 OCPU, 512 GB RAM, 50 GB Storage)
- 1 HeatWave cluster node (1 OCPU, 32 GB RAM, 50 GB Storage)
 - Can process up to 50 GB of data

MySQL HeatWave Shapes – Minimum Configuration

> OLTP + Machine Learning + Lakehouse

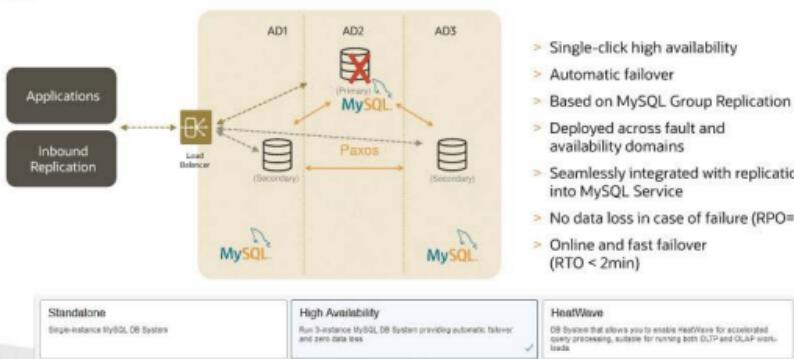
- 1 MySQL Frontend (16 OCPU, 512 GB RAM, 50 GB Storage)
- 1 HeatWave cluster node (16 OCPU, 512 GB RAM, 50 GB Storage)
 - Can process up to 1 TB of data
- Biggest shape for MySQL frontend
- 128 OCPU, 2 TB RAM
- Up to 64 HeatWave cluster nodes in OCI/Azure and 128 in AWS

Multiserver Capabilities

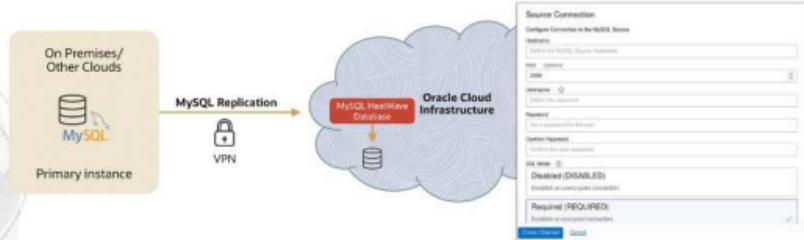
MySQL 8.4 includes support for multiserver connections, which allows you to connect to multiple MySQL servers simultaneously.



High Availability

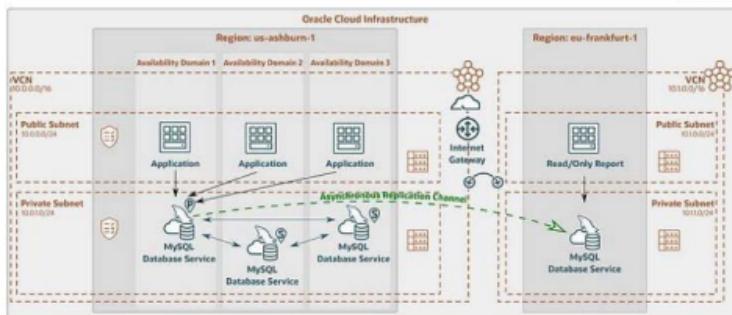


Inbound Replication



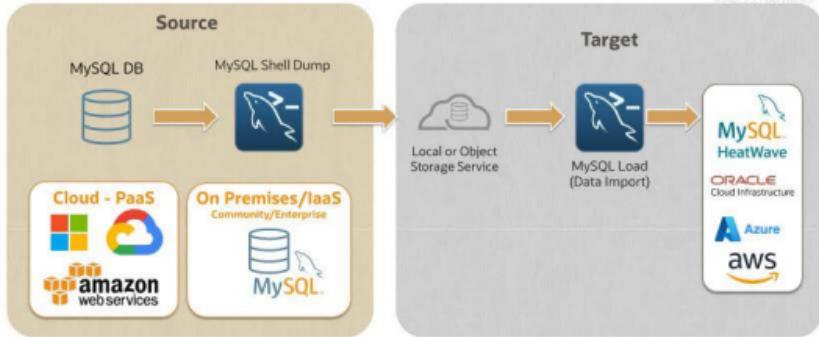
- Multi-source replication is not supported.
- Many filters are pre-configured (also replication from non-GTID sources).

Outbound Replication



MySQL Migration

MySQL Shell Dump and Load



MySQL Database Service provides tools including MySQL Shell Dump & Load to help organizations easily migrate their data out of Amazon RDS, Aurora, and to MySQL Database Service.

Migrating to MySQL HeatWave

The screenshot shows the OCI homepage with a specific section for MySQL HeatWave Migration. The title "MySQL HeatWave Migration Program" is prominently displayed. Below it, a paragraph explains the migration process using MySQL HeatWave, mentioning key resources like whitepapers, best practices, and technical training. A "Get Started" button and a "Request a demo" button are visible at the bottom.

The screenshot shows a blog post titled "Moving MySQL Databases from AWS to OCI Using MySQL Shell". The post includes a brief introduction, a "Read more" link, and a "Comments" section. The main content discusses the migration process using MySQL Shell, mentioning the use of MySQL Shell to move databases between OCI and AWS. It also highlights the use of MySQL HeatWave in OCI.

OLTP and OLAP

OLTP (Online Transaction Processing) and OLAP (Online Analytical Processing) are two distinct types of database systems.

OLTP is designed to handle high-volume, low-complexity transactions in real-time.

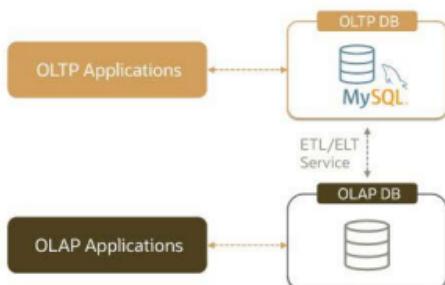
OLAP is designed to support complex analytical queries and reporting over historical data.



A Short Reminder: What's MySQL?

- > A Relational database
- > Transactional, ACID, MVCC – ARIES
- > OLTP: low latency, high throughput
- > InnoDB supports encryption and compression and InnoDB Cluster.
- > A typical MySQL database: Up to some TBs
- > Theoretical limits: The maximum tablespace size is 64TB (256TB).

Without HeatWave: Need Separate Systems for OLTP and OLAP

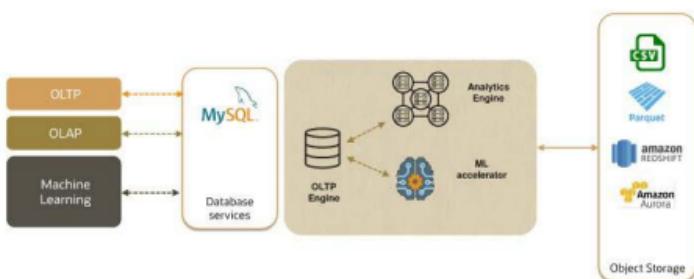


- > Separate analytics database
- > Complex ETL
- > No real-time analytics
- > Security and compliance risks
- > Increased costs

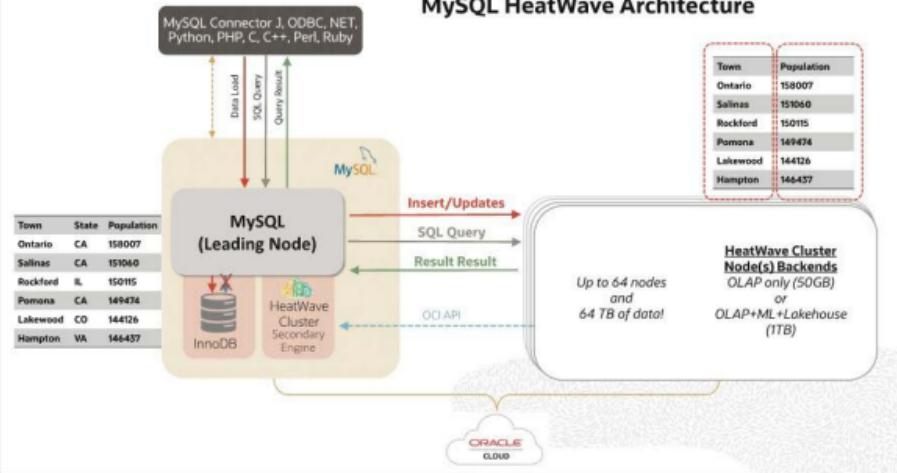
HeatWave for Real-Time Analytics

- > It's a single MySQL database for **OLTP, OLAP/Analytics, and Machine Learning** applications.
- > All **existing applications** will work **without any changes** – incl. Tableau, Qlik, Looker.
- > It provides machine learning-based **automation**.
- > **It provides extreme performance:** Accelerates MySQL by orders of magnitude.
- > It provides direct access to Object Storage for **Lakehouse**.

HeatWave for Real-Time Analytics



MySQL HeatWave Architecture



MySQL HeatWave Cluster

- > Initial load direct from InnoDB to HeatWave cluster nodes
- > Real-time updates for subsequent writes
- > Table/column-based offload
- > OLAP based:
 - Columnar
 - In memory
 - Vectorized
 - Distributed
 - Compressed



HeatWave AutoPilot

Machine Learning-Powered Automation for MySQL HeatWave

- > High query performance at scale, higher OLTP throughput, and the best price performance



And we don't stop there ... **We provide much more automation than what you'd get with another MySQL Cloud service.**

One of the reasons MySQL HeatWave performs so well is MySQL Autopilot. HeatWave's MySQL Autopilot automates many of the most important and often challenging aspects of achieving high query performance at scale—including provisioning, data loading, query execution, and failure handling (the challenges we discussed earlier when running a managed service in the cloud).

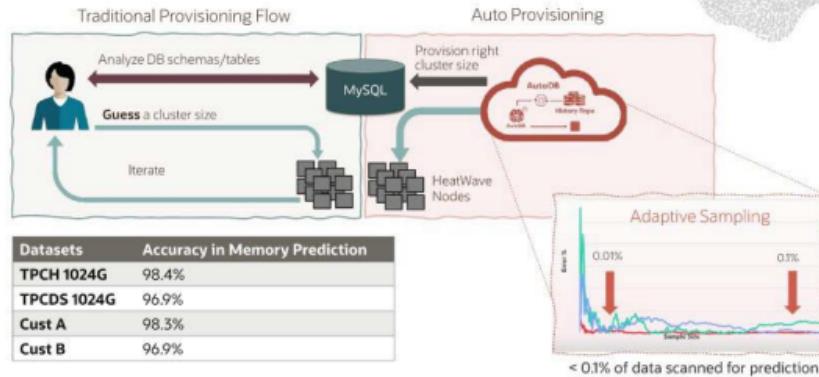
MySQL Autopilot uses advanced machine-learning techniques to automate HeatWave, further improving performance and scalability and making it easier to use—**saving developers and DBAs significant time.** It provides numerous capabilities for HeatWave and OLTP including the following:

- **Auto provisioning** predicts the number of HeatWave nodes required for running a workload, so DBAs and developers don't need to guess or manually estimate the optimal size of the cluster.
- **Auto query plan improvement** learns various statistics from the execution of queries and boosts performance of the system as more queries are run.
- **Auto thread pooling** allows the database service to process more transactions for a given hardware configuration, delivering higher throughput for OLTP workloads and preventing the throughput from dropping at high levels of transactions and concurrency.
- **Auto shape prediction** continuously monitors the workload, including throughput and buffer pool hit rate, to recommend the right compute shape at any given time—allowing customers to always get the best price performance.

No other MySQL database service provides those capabilities.

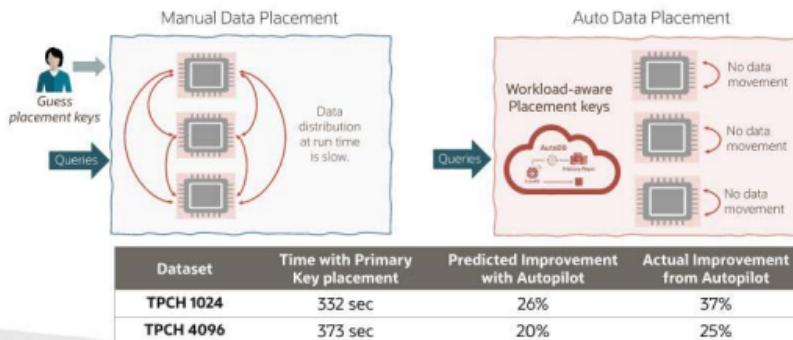
Auto Provisioning

Machine learning prediction of memory usage to estimate cluster size



Auto Data Placement

Machine learning prediction of optimal in-memory partitioning column

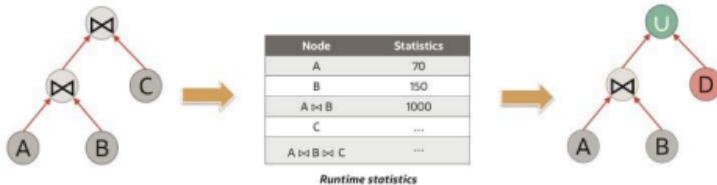


System predicts:

- Optimal columns to partition data in memory based on recent queries
- Improvement in run time

Auto Query Plan Improvement

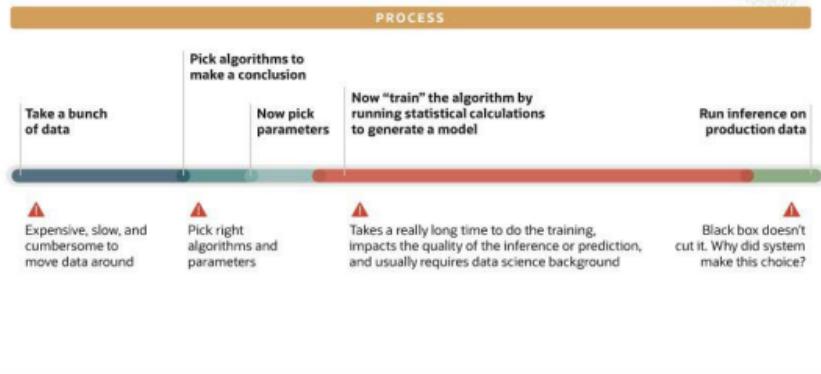
Optimizer learns and improves query plan based on queries executed earlier.



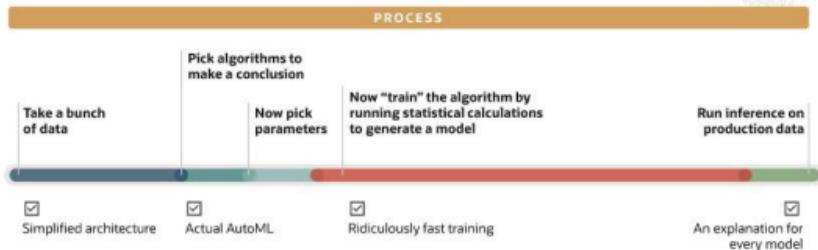
- > Traditional caching techniques are not intelligent.
- > With Autopilot, system gets better as more queries are run.
- > For example, Autopilot improves TPCH, TPCDS 24TB performance by 40%.

HeatWave Machine Learning

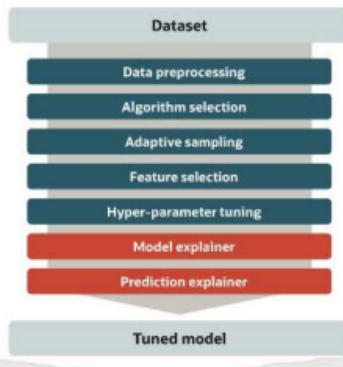
Machine Learning Overview



Machine Learning Overview: HeatWave AutoML



HeatWave Machine Learning



How is training in HeatWave ML fast?

- > Can be done in one pass
- > Highly parallel hyperparameter tuning technique
- > Intelligent sampling that doesn't compromise accuracy
- > Converges automatically
- > Scales with the size of the cluster

HeatWave ML Explainability

Helps in:

- > Regulatory compliance
 - > Fairness
 - > Repeatability
 - > Causality
 - > Trust
- 

Machine Learning Techniques

> Classification

- Predictive modeling problem where a discrete value is predicted for a given input data
 - **Examples:** Classify emails as spam or non-spam, customer churn prediction

> Regression

- Predictive modeling problem where a continuous value is predicted for a given input data
 - **Examples:** Home prices prediction, customer wait time in a queue prediction

> Forecasting

- Predictive modeling problem where a continuous value is predicted for time-ordered observations
 - **Examples:** Demand forecasting based on historical sales data, commodity price forecasting (purchasing aspect)

Machine Learning Techniques

> Anomaly Detection

- Predictive modeling problem where an anomalous value is predicted for a given input data
 - **Examples:** Identifying anomalous packets in network security, identifying anomalous sensor readings in an assembly line

> Recommender System

- Predictive modeling problem where a continuous value is predicted for a pair of user and item
 - **Examples:** Movie recommendation, click rate predictions

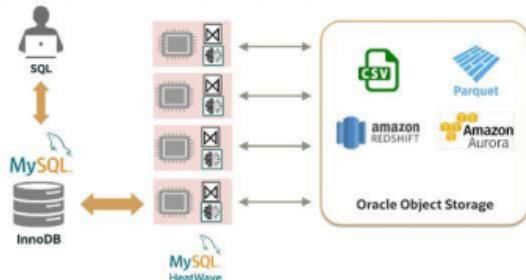
HeatWave Lakehouse



MySQL HeatWave Lakehouse

Supports data on object store, multiple file formats, larger data size

- > Data in InnoDB and HeatWave servers:
 - OLTP
 - Analytics
 - Machine Learning
- > + Data from Object Storage



Transaction Processing

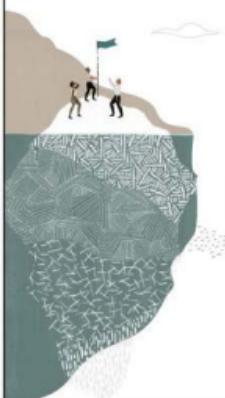
Analytics

Machine Learning

Object Store

Lakehouse

Summary



Features and Benefits

Supported Shapes

Multiserver Capabilities

OLTP and OLAP

HeatWave AutoPilot

HeatWave Machine Learning

HeatWave Lakehouse